WEISANG

FlexPro 2025

Data Analysis & Presentation

The New Dimension in

Data Analysis & Presentation

Quickly and easily
organize, analyze and graph data

Contents

1 Introduction

1.1 Welcome to FlexPro 2025

You will soon discover that by acquiring FlexPro, you have found a software tool that is not only extremely efficient, but it is also easy to use due to its versatility. Thus, you will discover many functions by simply using your intuition as you explore FlexPro. FlexPro offers you a logical structure and user interface that is easy to use, in addition to consistent implementation of standard Windows operations, which you are certain to recognize from other programs such as the Microsoft Office family of products.

You can also find all of the information and instructions in this manual in the extensive FlexPro Online Help, which you can access at any time by pressing the F1 key. This printed manual should provide you with an introduction to working with FlexPro regardless of the computer environment.

If you would like to spend time thoroughly learning about FlexPro, we suggest that you read all of the chapters of this manual in order. You can skip the Working with... sections the first time you read the documentation. You can find additional tutorials on how to use FlexPro in the Online Help. The most important section that we recommend for those new to FlexPro is Getting to Know FlexPro in just 15 Minutes.

We wish you the best of success in working with FlexPro!

The Team at Weisang

1.2 What's New in FlexPro 2025

Here you will find a detailed description of all the new features in FlexPro 2025.

New User Interface and FlexPro Project Database Features

- Display of list elements in the object list

  Formulas or other data objects that provide a list are now displayed like folders in the Folders window. If you select such a list there, its list elements are displayed in the object list, where you can select them individually.

- Simple navigation in the project database

With the new navigation bar in the object list, you can easily navigate from folder to folder in the project database. The functionality corresponds to that of Windows Explorer.

- Search in Project Database dialog box preserves settings

  The Search in Project Database dialog box now retains its settings when you close and reopen it.

- Search and replace text in project databases

  Use the new Find and Replace dialog box to search selected objects, folders or the entire project database for text that you want to replace.

- Undoing global operations

  You can use the Properties window or FlexPro's new Find and Replace function to edit many objects at the same time. You can now undo such an operation as a whole with the Undo command or redo it with the Redo command.

- Calendar time values with time zone

  If you enter calendar time values in the data set editor, you can now optionally specify the time zone, e.g. '02-01-2021 13:14 UTC', '1-2-2021 13:14 UTC-2' or '1-2-2021 13:14 UTC+2:30'.

- Direct selection of list elements in the Object List window

  If you select an analysis object or a formula that provides a list of results, you can now switch to the List Llements tab of the object list to select individual elements for display or further processing.

- Excel workbook and Excel data link deprecated

  The Excel Workbook and Excel Data Link objects, which could be used to open Excel files directly in FlexPro in order to extract data, have been deprecated because newer versions of Excel no longer adequately support the interface required for this. These are no longer available in the standard installation of FlexPro 2025. Instead, use the data import filter for Excel data, which fulfills the same purpose with higher performance and stability.

  Your system administrator can activate support for the Excel Workbook and Excel Data Link objects in the Windows registry. To do this, under HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER a DWORD key

Software\Weisang\FlexPro\14\ExcelImport\SupportExcelFolder must be created with the value 1.

- Management of user permissions via Windows

  The user permissions are now only displayed in FlexPro and can no longer be changed. The permissions are defined by the administrator of the Windows computer on which FlexPro is running. A configuration file (ADMX/ADML) for the Windows Group Policy is available for FlexPro as a separate installation program.

- Correction of broken references when moving objects

  If you move objects or entire folders in the project database, the references broken by the operation are now automatically corrected. Previously, this was only supported for renaming objects.

- Other

  The Unit column is now displayed by default in the object list.

New Presentation Features

- Sorting column and row tables

  You can now simply output the data of a column or row table sorted by any column or row.

- Aligning tables

  Specify how the table should be aligned in the rectangular area in which it is displayed.

- Fixed column width and height for tables

  For the column and row table, you can now specify that the size of the cells should not be adjusted to the content.

- Changing the size of diagrams and tables

  Changing the size with the mouse is now done symmetrically to the center of the object.

- New diagram type Box Plot

The box plot, also known as the box-whisker plot, is a diagram that enables the clear presentation of the most important robust measures of position and dispersion. The minimum, the lower quartile, the median, the upper quartile and the maximum are shown.

New Analysis Features

- Extended analysis object Signal Sampling

  The analysis object has been expanded to include two methods for data reduction based on the FPScript function BlockCompress.

  For more details see: Signal Sampling analysis object

- Extended analysis object Signal Scaling

  You can now also use the analysis object to linearly scale the X-component of a data set.

  For more details see: Signal Scaling Analysis Object

- New analysis object and analysis template Lifetime Analysis (Statistics Option)

  Performs a lifetime analysis based on the two-parameter Weibull distribution. The input for the lifetime analysis is one or more random samples with object service lives.

  For more details see: Lifetime Analysis object and template

- New analysis object Empirical Distribution (Statistics Option)

  Calculates the empirical distribution function (for unclassified data).

  For more details see: Empirical Distribution analysis object

- Estimate parameters of Weibull distribution from sample (Statistics Option)

  The Distribution analysis object can now estimate the parameters of the Weibull distribution from the sample.

  For more details see: Distribution analysis object

- Extension of the analysis objects for acoustics (Acoustics Option)

  - In the analysis objects for acoustics, you can now specify whether the input data is available as voltage signals or pressure signals.

- If no calibration is required, you can now explicitly select No calibration instead of specifying a calibration value of 0 as before.

- External parameterization of analysis objects

  You can now directly insert a data set reference for numerical parameters of analysis objects, e.g. to parameterize several analysis objects centrally or on the basis of a calculation.

  For more details see: External Parameterization of Analysis Objects 293

- Data Query Enhancements
  - The new Update result option updates all objects that are returned as the search result of the data query.
  - The Include subfolders in search option is now also available when searching in parent folders.
  - The performance of chained data queries has been improved.
  - When querying data in indexed files, search results for a file are only returned once it has been fully indexed.

New Data Cursor Features

- Undo function for Edit Value and Edit Range

  The list boxes Cursors[Edit] > Edit Value and Cursors[Edit] > Edit Range now have an Undo command to undo previous edits.

- Undo/redo for setting, removing and designing markers

  The Undo and Redo commands now also support setting, removing and designing markers.

- Resizing symbol markers with the mouse

  You can now change the size of symbol markers directly with the mouse.

New features in the area of data import/export

- Color assignment for data sets from text and Excel files

  In the import wizard for text and Excel files, you can now assign a curve color to the data sets to be imported.

- Data import and export for Apache Parquet data

  Apache Parquet is an open-source, column-oriented data file format designed for efficient data storage and retrieval. It offers efficient data compression and encoding methods with improved performance for processing complex data in large quantities.

- File data link with external or relative file path

  - In the file data link, you can now optionally specify a data set from which the file path is to be taken. This makes it easy to change several links to a different file.

  - You can now also specify the file path relative to the path of the project database.

- Importing additional data sets from an already imported measurement data file

  The new Import and add option leaves existing data sets in the folder unchanged and adds newly imported data sets.

- Importing various data structures from Excel files

  In the Wizard for importing Excel files, you can now specify the data structure to be created from the data, e.g. data matrix or signal series.

- Export of several presentation objects as images

  You can now export several presentation objects together as images.

- New data import for FDSN miniSEED files

  The miniSEED data format was specified by the International Federation of Digital Seismograph Networks (FDSN) for data acquisition, archiving and data exchange.

- New data import for IEEE PQDIF files

  The Power Quality Data Interchange Format (PQDIF) is a binary file format specified in the IEEE 1159.3 standard and is used for the exchange of voltage, current, power and energy measurements between software applications with a focus on power quality.

- Support for National Instruments DASYLab Universal Format 4

  The National Instruments DASYLab import filter now supports the import of DDF files that are stored in Universal Format 4. Since release 16.1, DASYLab writes the UF4 format as soon as a channel that supplies SmartMux data is connected to at least one input of the "Write Data" module.

New Programming Features

- Integration of the Python programming language in FlexPro (Professional and Developer Suite editions)

  Python is a widely used, open-source programming language for which libraries are available for a wide range of technical and scientific applications. FlexPro integrates the Python programming language with the new Python Formula object. Python formulas, functions and lists can be created and used just like their FPScript counterparts. With the "flexpro" object available in Python, you can access the data from data sets or formulas to process them in Python and return the result to FlexPro, with Y, X and Z components if necessary. It is also possible to call the functions built into Python directly from FPScript formulas.

  Python is available in the Professional and Developer Suite editions of FlexPro. With the Basic edition of FlexPro, you can execute Python formulas but not create them.

- Improved debugger

  The debugger for debugging formulas has been completely revised. It supports both FPScript and Python formulas.

  - With the new Call Stack window, you can see exactly which formulas have been called in which order and can switch between the code windows with a simple mouse click.
  - The new Breakpoints window shows you all breakpoints set in formulas in the project database and you can navigate to the code line of a breakpoint with a mouse click.
  - The set breakpoints are now saved in the project database so that they are no longer lost when closing.
  - Breakpoints have a variety of useful properties:
    - You can temporarily deactivate a breakpoint without having to delete it.

 - You can use the optional hit counter to specify after how many hits the debugger should stop.
 - You can specify any FPScript or Python expression as a stopping condition, which must result in true, e.g. "y[i] >= 100".
 - You can display a message if the breakpoint is passed. Embed FPScript or Python expressions in the message text to output current variable contents, e.g. "y[i] = {y[i]}".
- The revised Watch window shows you the current result and the contents of all local variables of a formula in a list. You can add any expressions to watch them. You can display non-scalar data at the touch of a button in the Data Preview window.

- Updated unit management

  The unit management and its documentation have been adapted to the 9th edition of the SI brochure published in 2019, which describes the SI system of units. The prefixes Quecto, Ronto, Ronna and Quetta have been added, the astronomical unit symbol has been changed from "ua" to "au" and the factors of some units have been updated. The unit gal with symbol Gal, defined as 1 cm/s², has been added to the table "Popular non-SI units".

- New FPScript Functions

Function	Description
BlockCompress	Reduces the number of values in a data set using block operations or splits it into a list of segments.
BoxPlot	Determines the statistical parameters for displaying a box plot.
EmpiricalDistribution	Calculates the empirical distribution function (for unclassified data).
Erf	Calculates the Gaussian error function.
Gamma	Calculates the gamma function for real-valued arguments.
LogGamma	Calculates the natural logarithm of the absolute value of the gamma function for real-valued arguments.
ParameterList	Returns the parameters of a data set as a list.
RemoveDuplicates	Removes duplicates from a data series of strings.
SearchListElements	Searches for one or more list elements in a named list.

Function	Description
WeibullFit	Performs a curve fit using the two parametric Weibull distribution (Lifetime Analysis Option).

- Enhancements to existing FPScript functions and operators
  - The FPScript functions J0, J1, Jn, Y0, Y1 and Yn now also support complex numbers.
  - The FPScript functions SIUnits, Unit, UnitSymbol and UnitType now also support lists.
  - With the FPScript function SearchStrings, you can now also search for unique strings.
  - The FPScript function Smooth now supports various modes for smoothing the edges of the data set.
  - You can now use the index operator to extract individual characters or substrings from a string.
  - With indexed assignment, you can now also replace individual characters or substrings in a string.
  - The FPScript function DeltaCompress has an additional argument that controls how void values in the data set are handled.
  - The FPScript function CompensatingSpline has been renamed to SmoothingSpline.
  - For the FPScript function DataQuery, object paths of containers (folders, files, file folders) in which the search is to take place can now also be specified as the first argument.
  - The indirection operator now behaves exactly like a direct object reference. When referring to a data object, it does not provide an object reference but its data. As with the direct object reference, you can force an object reference to a data object with the As Object addition or by using the Set statement.

1.3 Comparison of FlexPro Editions

FlexPro View OEM

The FlexPro View OEM freeware is the perfect add-on to your measurement system. It also covers presentation and interactive analysis with data cursors. The data imported is limited to the data formats of a specific measuring device manufacturer.

FlexPro Basic

FlexPro Basic covers presentation and analysis of individual measurements. This edition includes a basic set of analysis procedures. You can also record and run macros. The VBA development environment, however, is not part of this edition. The same applies to Python formulas. You can execute these but not edit them.

FlexPro Professional

FlexPro Professional covers presentation, analysis of individual and series measurements as well as automation and supports team collaboration. This edition of FlexPro can calculate analyses, preview images and data statistics in the background, uses hardware graphics acceleration and parallelizes vector arithmetical operations and object updating on multicore systems. FlexPro Professional allows you to analyze videos recorded in conjunction with measurement data. This edition of FlexPro features the powerful data query tool for evaluating series measurements and can use the document collection to combine documents from individual analyses into final reports.

FlexPro Professional is the best edition for working in a team, since it supports shared template databases that you can use to access text import schemas, analysis templates and algorithms deployed throughout your enterprise. The Data Explorer option 95 in conjunction with FlexPro Professional in client/server mode in order to provide your team with access to shared data.

FlexPro Professional contains the Microsoft Visual Basic for Applications (VBA) development system, not only allowing you to record and run macros, but also to write your own programs and dialog boxes. FlexPro Professional also provides access to the FlexPro Automation object model, which you can use to control all editions of FlexPro remotely. It also contains a toolkit that you can use to program your own import filters and FPScript functions. Programs written with FlexPro Professional can also be used for other editions of FlexPro.

FlexPro Professional allows you to create control panels and forms without VBA programming. These can then work in all editions of FlexPro.

The Python programming language is integrated in FlexPro Professional. Python formulas, functions and lists can be created and used just like their FPScript counterparts. Your libraries created in Python can be easily integrated and used. The Python formulas created with FlexPro Professional also run in the Basic edition of FlexPro.

You can expand the range of FlexPro Professional mathematical analysis procedures using the following options:

Acoustics Option 308

Digital Filters Option 298

Human Body Vibrations Option

Count Option 300

Order Tracking Option 301

Spectral Analysis Option 303

Statistics Option 297

ASAM ODS Data Import Option 170

Data Explorer Option 95

FlexPro Developer Suite

FlexPro Developer Suite is the cost-effective, complete software package form FlexPro and is equal to FlexPro Professional with all options included.

FlexPro Reader

You can use the free FlexPro Reader to open project databases created with one of the FlexPro editions mentioned above as read only and view the results. You can display diagrams, documents and other objects and use cursors to view the data inside. The automation interface is not available in FlexPro Reader, and a watermark is placed on expressions.

FlexPro Runtime

FlexPro Runtime Edition is required as a runtime environment for programs that you develop for FlexPro if a higher license of FlexPro is not installed on the target system. You can display diagrams, documents and other objects and use cursors. However, only program-controlled editing of objects is possible with this edition. Only data sets can be edited interactively for the ease of data entry.

Comparison chart

■ Included
○ Optional

View OEM
Basic
Professional
Developer Suite

Operation & Data Management	V	B	P	D
Project database, size only limited by hard disk	■	■	■	■
Modern user interface with an office suite look and feel	■	■	■	■
Import Excel files, text files and over 60 binary formats		■	■	■
Customizable menus and toolbars		■	■	■
User profile management		■	■	■
Indexing and data queries for analyzing series measurements			■	■
Multi-core processor support and hardware graphics acceleration			■	■
Share presentation and analysis templates with colleagues on the network			■	■
ASAM ODS data import option			○	■
Data Explorer option			○	■

Presentation	V	B	P	D

2D and 3D diagrams, column, row and cell tables, texts with calculations	■	■	■	■
Document Editor	■	■	■	■
Presentation and document templates		■	■	■
Customizable menus and toolbars		■	■	■
Analyze measurement data and videos synchronously			■	■
Combine individual reports into final reports			■	■

Analysis	V	B	P	D
Analyze and dimension measurement data interactively with data cursors	■	■	■	■
Simple calculations in FPScript	■	■	■	■
Full range of functions and operators for FPScript		■	■	■
Custom FPScript functions		■	■	■
Analyses and presentations at the press of a button with the Analysis Wizard		■	■	■
Design analysis templates yourself		■	■	■
Ready-to-use analyses for curve fitting, signal analysis, statistics, spectral analysis, event isolation, filtering, counting		■	■	■
Parallel looping in FPScript			■	■
Python formulas and Python functions			■	■

	V	B	P	D
Time-frequency spectral analysis			■	■
Circle approximation			■	■
Statistics option			○	■
Counting method option			○	■
Order tracking option			○	■
Spectral analysis option			○	■
Human body vibrations option			○	■
Acoustics option			○	■
Digital filters option			○	■

Automation	**V**	**B**	**P**	**D**
Expand or control FlexPro remotely using Visual Basic, C#, C++, etc.	■	■	■	■
Macro recorder	■	■	■	■
Integrate macros into user interface		■	■	■
Create control panels and forms with controls			■	■
Microsoft Visual Basic for Applications (VBA)			■	■

1.4 License Manager

The FlexPro License Manager shows you what FlexPro licenses are available on your computer and network and allows you to select a license and to specify the preferred FlexPro configuration, consisting of the edition and options.

Single User License

When launching FlexPro after completing the installation, an evaluation license is installed on your computer, which gives you 30 days to try it out. An Internet connection is required in order to download the evaluation license. You will need to activate the license as soon as the evaluation period expires in order to continue using the complete version of FlexPro. Without a valid license, FlexPro runs in reader mode. Activation permanently binds the purchased single-user license to the computer. To activate the license, use the product key supplied with FlexPro. This will allow you to activate the software for personal use up to two times. For instance, you can install and activate FlexPro on your desktop computer and on a laptop. You can activate FlexPro online by communicating directly with the Weisang activation server, or you can activate it offline by swapping files. You can also deactivate a license. The license will then be credited to your product key and you can install FlexPro on a different computer.

Important note Single user licenses cannot be used on a virtual machine or in a terminal server session! If you install a single-user license on a virtual machine, it will be activated there but will not work. You will lose one activation!

Network License

A network license, also known as a concurrent use license or floating license, is usually installed on a server by the network administrator and limits the number of concurrent users. When launching FlexPro, the license is requested by the server, and when you close FlexPro, the license is then available once more to other users. If you have set up FlexPro to use a network license, you must connect to this server as long as FlexPro is running. There are also FlexPro network licenses available that let you check out, i.e. transfer, the license to your local computer for a period of 120 days.

Dongle-based Licenses

Weisang can also provide the license with a USB dongle for an extra charge. This type of license is not tied to the computer, but is instead tied to the USB dongle and can be moved to a different computer by simply inserting the dongle into a USB port.

Note: More information about the License Manager is available in the document Installing and Using the FlexPro License Manager.

1.5 Working with the License Manager

Opening the License Manager

- Click on File > Info > License Manager.

During the evaluation period and after a license has expired, the License Manager will appear automatically when FlexPro is started.

Online Activation

1. Launch FlexPro and click on File > Info > License Manager.
2. Click Local.

If you have a dongle-based license:

3. Insert the USB dongle into an available USB port on your computer.
4. Wait until the USB dongle driver finishes installing and the red LED on the dongle lights up.
5. In the FlexPro License Manager dialog box, click Extended and check the first entry in the Show Host IDs dialog box when it appears. The entry must be HASPHL=xxxxxxxxxx.
6. When online activation is selected, a connection is established with the rlm.igb.weisang.com Weisang activation server. If you want to connect using a proxy server, click Proxy Server to enter the login information.
7. Under Product key enter the product key for FlexPro and for all the options that you want to install.

8. Now click on Activate.
9. The dialog that appears next will show you all the licenses that will be installed once you complete activation. Check the list carefully and click Cancel if you see any errors. Otherwise, click OK to continue with activation.
10. After activation has been confirmed, your available license will be displayed. Select the edition and options that you want to use and close the License Manager by clicking OK.

If online activation fails, FlexPro will display an error message. The cause is usually a block in communication between the Weisang activation server and your computer. If this is the case, try Offline Activation 27.

Offline Activation

To activate a license offline, i.e. without connecting to the Internet, you will need to provide Weisang with what is called the host ID of your computer or your dongle in addition to the product key for the license. The host ID identifies your computer/dongle and the license is tied to the computer/dongle by way of the host ID.

If you have a dongle-based license:

1. Insert the USB dongle into an available USB port on your computer.
2. Wait until the USB dongle driver finishes installing and the red LED on the dongle lights up.
3. Launch FlexPro and click on File > Info > License Manager.
4. In the License Manager, click Extended.
5. In the Show Host IDs dialog box that appears you will see the host IDs. If you have connected the dongle and it is detected correctly, the first entry should be HASPHL=xxxxxxxxxx. Click Copy to the Windows clipboard.
6. Now send an e-mail to licensing@weisang.com. Include in the e-mail the displayed host IDs (paste from the Clipboard using CTRL+V) and the product keys that were provided to you with FlexPro. Please note that your e-mail must include a valid e-mail address for the reply. After a Weisang representative has activated your license, you will receive an e-mail with a license file (.lic) attached.
7. Open the folder `C:\ProgramData\Weisang\FlexPro` and copy the license file into this folder.
   Notes: The `ProgramData` folder is not usually displayed in Windows Explorer. You should therefore enter the entire path name using the keyboard. If a file with

the same name is already in the folder, rename the new file before copying it to the folder.

After you have copied the file, open the FlexPro License Manager again. FlexPro will then import the license and will display the available licenses. Select the edition and options that you want to use and close the License Manager by clicking OK.

Checking Out a Network License

1. Launch FlexPro and click on File > Info > License Manager.
1. Make sure that the Network tab is active and that the specified edition and selected options correspond with the configuration that you want to use offline.
2. If the network license can be checked out, the Check Out button appears. Click on this button.
3. In the dialog box that appears, enter the number of days that you wish to check out the license. This can be any length of time up to 120 days. After this period has expired, the local license will be deactivated automatically and the license will be available on the server again.
4. Click OK to check out the license.
5. If a confirmation message appears, the FlexPro License Manager will automatically switch the license to Local and then you can disconnect the computer from the network. Otherwise, you will have to repeat the process by selecting a shorter checkout period.

Returning a Checked Out License Early

1. Make sure that your computer is connected to the license server network.
2. Click on File > Info > License Manager and make sure that the Local tab is active.
3. If the local license has been checked out, the Cancel button appears. Click on this button.
4. FlexPro then returns the license to the server and the License Manager converts it back to a network license. To close the confirmation dialog box and the License Manager, click OK.

Transferring a License

a) License without a USB key

Before transferring a local software license from one computer to another, deactivate it on the original computer and then reactivate it on the new system. This procedure requires that you have an active maintenance agreement with Weisang for the license.

1. Click on File > Info > License Manager and make sure that the Local tab is active.
2. In the License Manager, click Deactivate.
3. In the dialog box that appears, click OK to confirm that you want to deactivate the license.
4. You can now install the license(s) on another computer by reusing the product key.

b) License with a USB dongle

This type of license is not tied to the computer, but is instead tied to the USB dongle. This means you do not need to deactivate the license.

1. Insert the USB dongle into an available USB port on the computer to which you would like to transfer the license. As soon as you have an Internet connection, Windows will automatically download and install the driver for the USB dongle. Your other option is to install the driver manually. More information about this is available in the document Installing and Using the FlexPro License Manager. If Windows can read the dongle correctly, a red LED lights up on the dongle.
2. Now all you have to do is transfer the license that is tied to the dongle to your computer. You can do this online by entering the product key that came with the dongle in the FlexPro License Manager on the new computer and then clicking Activate. Alternatively, you can copy the license file from `C:\ProgramData\Weisang\FlexPro\FlexPro.lic` to the same folder on the new computer.

Troubleshooting

A failed installation or License Manager setup usually manifests itself when FlexPro cannot find a license when launched. When an error occurs, a small button with a warning triangle appears in the License Manager. If the button appears, click on it to find out more about the error.

Check if the license file is installed correctly on the local computer:

The folder `C:\ProgramData\Weisang\FlexPro` must contain at least one file with the `.lic` extension. The file should contain at least one license entry that looks similar to the following:

```
LICENSE weisang flexprodevelopersuite 14 permanent uncounted
  hostid=rehost=111e02d861211656b958ed82e2b254fc6ebc8608
  _ck=6708fccc64 sig="60P0450WF36CP91AQQNFHB82YFN7ENN6J1XP8WG22H6PTXS
  GM7243KM4X44HP863EKTDNJAJ4R"
```

Checking the Host ID

FlexPro single user licenses use a so-called "re-hostable" host ID. This host ID is created when a license is activated and is deleted when the license is deactivated.

The folder `C:\ProgramData\Reprise\weisang` must contain at least one folder called "License". In the example above, this would be `flexprodevelopersuite`.

To prevent bypassing the activation mechanism, the re-hostable host ID is read only. It is not possible to restore it using a backup program.

If no host ID is available, you will have to carry out activation again. If the host ID is available, but you are unsure of its integrity, please contact Weisang customer support or the retailer where you purchased FlexPro.

Additional Support

Information on troubleshooting FlexPro network licenses is available in the document Installing and Using the FlexPro License Manager.

If you need troubleshooting support, please register the access key**(MXXX-XXXX-XXXX-XXXX**) that came with your FlexPro license at www.weisang.com and request support in MyWeisang. We will then contact you to help you solve the problem.

2 Tutorials and Examples

2.1 Getting Started

Getting to Know FlexPro in just 15 Minutes

This video tutorial provides you with a brief overview of the structure of FlexPro and the basic command methods.

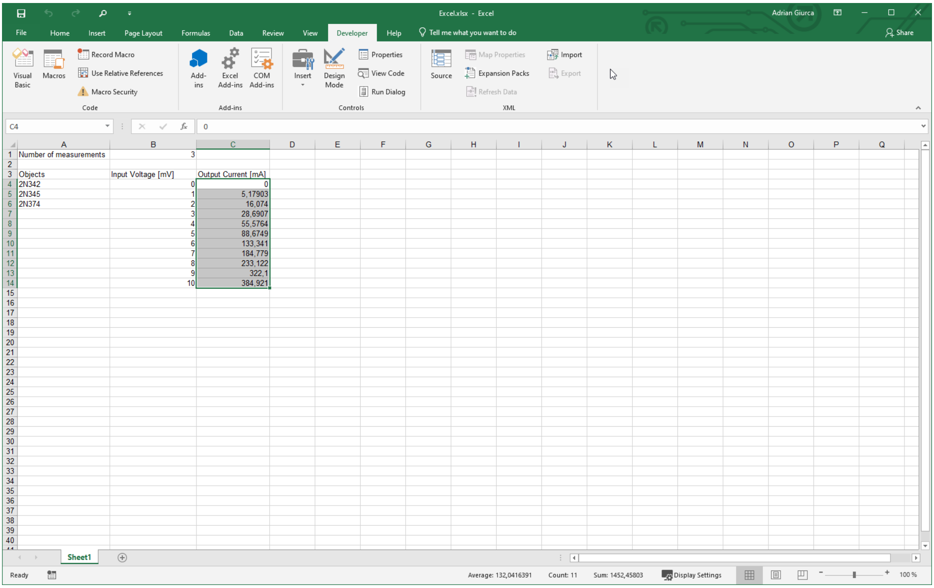
You can find it on the Weisang website under: Getting to Know FlexPro in 15 Minutes

Tips for Excel Users

If you already use Microsoft Excel or some other spreadsheet program, then the information in the following sections will be very useful. Here we show you how you can more easily and efficiently manage operations familiar to you using FlexPro.

Data Management

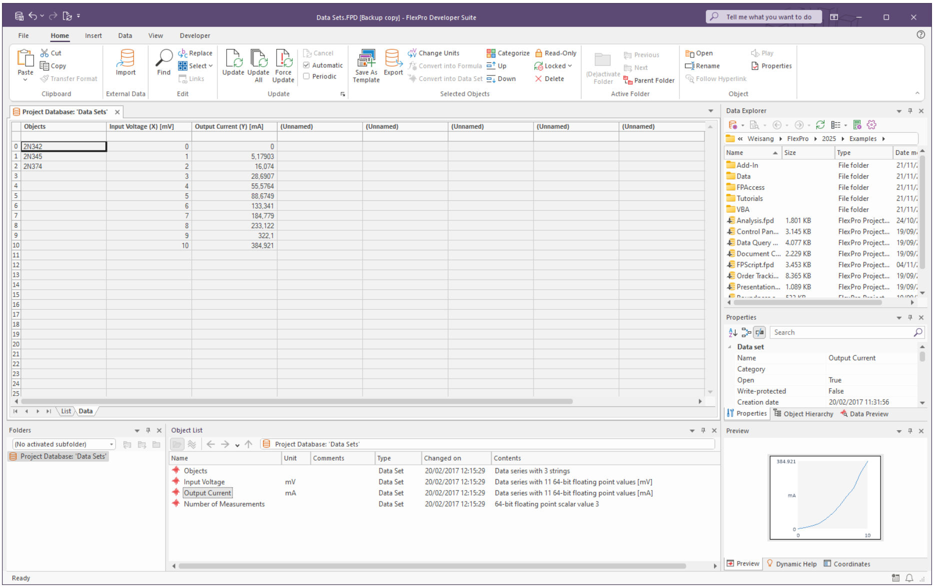
When working with a spreadsheet, you usually arrange all of the data that you want to analyze in a table. Scalar values, such as the serial number or the date of a measurement, occupy one cell in the table. The results of a series of measurements are usually arranged in one or more columns.



As long as the data volumes are small, this is very practical, since you have direct access to all of the data. However, this procedure is limited when you are working with large volumes of data or when the amount of data varies, as in a measurement series, for example.

In developing FlexPro, however, situations like these were taken into account. Data in FlexPro is therefore divided into multiple data sets 132 that are stored together in a FlexPro project database 46. In this case, each data set only accepts data of the same type. If you look at the example table more closely, it is clear that it contains data with different data structures 122 and data types 118. For example, the measurement number is a scalar value, the list with the samples is a data series with strings, and the input voltage and the output current are both data series, each with eleven values. As you can see, the data structure specifies how associated data is organized, and the data type specifies whether the data consists of floating point values, strings or other values.

For each element in the example, FlexPro provides a matching data structure used for the corresponding data set. For example, the serial number of the measurement is saved as a data set Measurement Number with a scalar value. The list contains the test objects as a objects data set with a data series containing several strings and the Input Voltage and the Output Current as data series. In the images above you can see an Excel workbook and the imported data sets in FlexPro. In the example, the data sets are available as Excel data links, i.e. they do not contain the data directly, but refer to the corresponding areas in the Excel file. You can convert these types of links into FlexPro data sets at any time. You then can edit data set values using FlexPro.



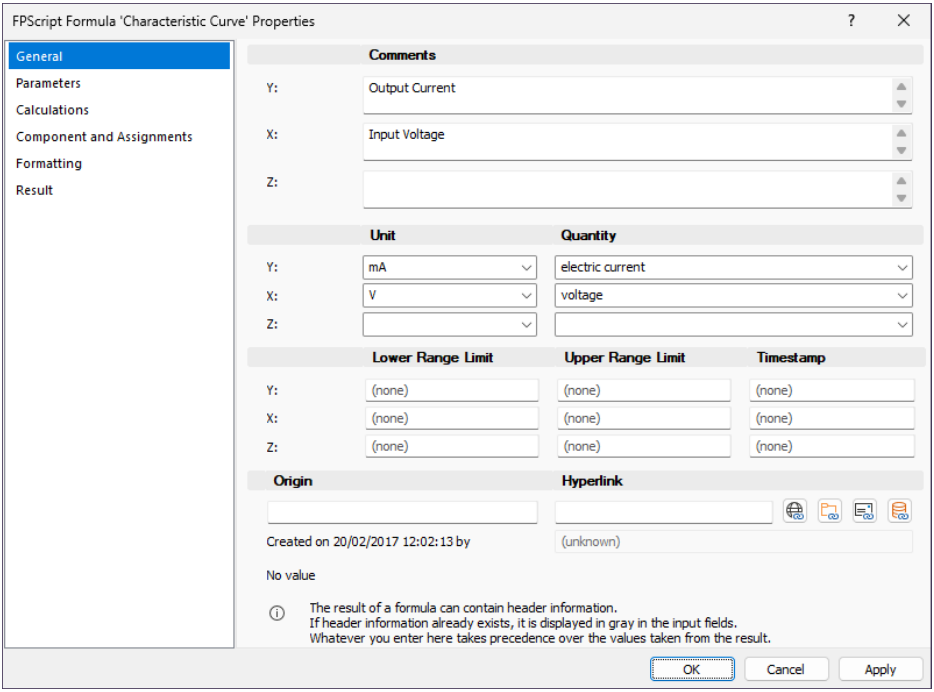
As the above picture illustrates, FlexPro can display data series similarly to Excel in a data grid, and you can easily edit them.

To work seamlessly with FlexPro, it is very important that you recognize which data structures are present. Often one is not aware of this when working with a spreadsheet, since the data structures implicitly result from the way the data is arranged in the table cells. A common mistake, for instance, is to place adjacent columns in a single data set, even though the columns contain different values. The following table will help you determine the correct data structure of the various elements in your spreadsheet.

FlexPro Data Structure	Description	Appearance in Spreadsheet
Scalar value	A scalar value, such as the date of a measurement, a single measurement value or another parameter	A cell containing a value, such as the measurement number in the above example.

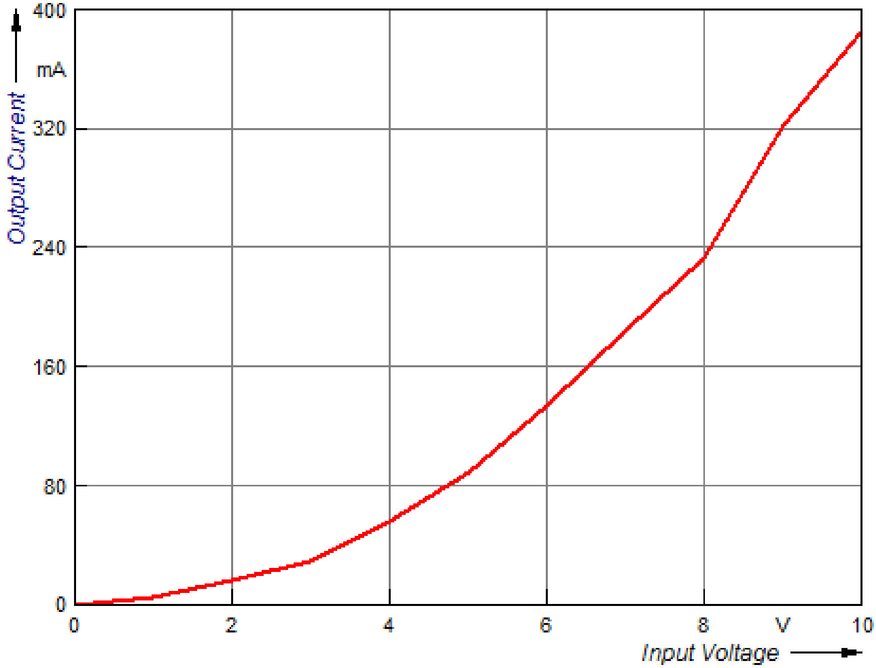
Data series	A column with several associated values. All values have to have the same physical unit.	A column with several cells, such as the input voltages and output currents in the above example
Data matrix	A matrix of associated values. All values have to have the same physical unit.	Several columns adjacent to each other. These matrices are used for surface graphics, for instance.

The data itself often also contains additional information, such as the physical unit or comments. This is saved as header information in the data sets. When the data is later displayed, FlexPro can access it and automatically label the diagram axes, for example.



Signals

If you display the output current as a curve in a diagram in your spreadsheet, then the indices of the measurement values are first displayed on the X axis of the diagram. However, it makes more sense to display the measured characteristic curve, which means that you would have to write the input voltages belonging to the output currents measured onto the X axis. The two data series Input Voltage and Input Current thus together form a signal.



Signals very often occur in measurement applications, for instance, whenever you measure a time series. The times at which the measurement values were sampled form the X component, and the measurement values themselves form the Y component of the signal. Since signals occur so frequently, FlexPro offers you signals in their own data structure. A data set containing a signal consists of two data series of equal length, which are called X component and Y component.

If you display a data series in a diagram as a curve, then the X values are numbered consecutively, as is also the case in the spreadsheet. However, if you display a signal as a curve, the X values belonging to the signal's Y values are automatically displayed on the X axis. Analyzing signals is also easier than analyzing separate measurement series. An integration is then written as `Integral('Characteristic Curve')` instead of `Integral('Output Current', 'Input Voltage')`. You can access one of the signal components any time: `Signal.x` provides the X values and `Signal.y` provides the Y values. You can specify individual comments and the individual physical unit for each component in the data set. FlexPro recognizes more data structures than just the "Signal" data structure presented here. You can find out more under Data Structures 122.

FlexPro usually creates signals as soon as data are imported. When importing text or Excel data, you can specify in the wizard which column contains the X data.

Instead of creating signals, you can also assign 125 data series to each other. Simply enter the data set with the X values as an assigned data set into the data set with the Y values. FlexPro can then automatically adopt the X data set when a diagram is created, for instance.

Editing Data

When you open a FlexPro data set, its contents are displayed as a table in the data set window, allowing you to carry out your editing in the manner with which you are already familiar. FlexPro can display all of a folder's data sets with data series 123 and data matrix 123 data structures next to one another in the folder's Data View 132. It is not necessary for you to open them in separate windows.

The following picture shows a signal opened in the data set editor:

	X	Y
0	0	0
1	1	5.17903
2	2	16.074
3	3	28.6907
4	4	55.5764
5	5	88.6749
6	6	133.341
7	7	184.779
8	8	233.122
9	9	322
10	10	384.921
11		

Data Import

To import data into a spreadsheet, you can read the data from a text file, for instance, which will then be more or less copied "as is" into the spreadsheet.

FlexPro features import filters for a wide variety of binary file formats for measuring devices and data acquisition application as well as a wizard for text and Excel files. You will have to give the wizard some information about the file structure, which you may already be familiar with from working with spreadsheets. The result, however, is always a series of data sets that are displayed in the object list.

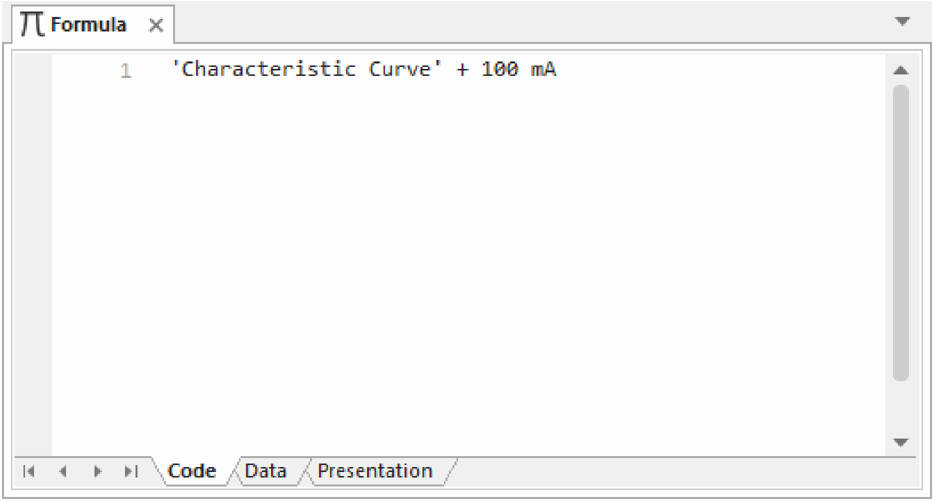
You can use the Excel File Import Wizard to import data that is arranged in columns or rows directly from Microsoft Excel XLS or XLSX files. Excel does not need to be installed on the machine for this type of file import.

In addition to data, FlexPro also transfers information such as units and comments. You can copy data to the FlexPro project database or create links to the original data. If you work with links, changes to the original files immediately affect the analysis results in FlexPro.

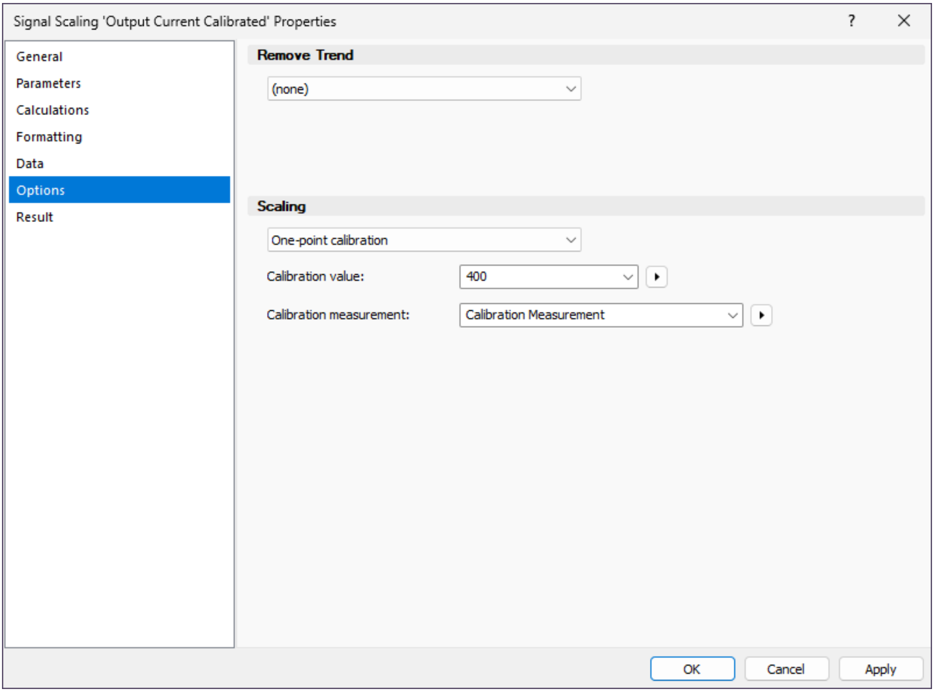
FlexPro can exchange data directly with other programs as well. You can export 186 data in other program formats and use the FlexPro OLE data link 162 to exchange data in both directions dynamically.

Using Data in Calculations

If you use data in spreadsheet calculations, then you enter a formula in the cell in which the result is to appear. This formula refers to the cells to be used in the calculation. The process is similar when working with FlexPro. In the project database, you set up formulas 235 that perform the desired calculations. In addition to providing scalar values as the result for formulas that you set up, FlexPro can also return data series, signals, etc. When the formula is calculated, FlexPro automatically determines the data structure of the result, which depends upon the elements to be used in the calculation. For example, the formula `'Characteristic Curve' + 100 mA` adds, for instance, an offset of 100 mA to the current values contained in the Y component. The result is once again a signal, and you do not have to worry about the X component, which is very practical.



However, you can also write proper programs in formulas. To do this, use the FPScript 246 language. You do not even have to write a formula for common analyses. FlexPro offers special analysis objects 293 for this purpose. Technically, these are also formulas that FlexPro writes for you. You simply select the analysis options in a dialog box. And if you change these later, the formula is corrected automatically.



When using formulas, it is important to know that you can use formulas like data sets. For example, you can calculate the results of other formulas within a formula or display a formula as a curve in a diagram. Formulas even have the same header information as data sets.

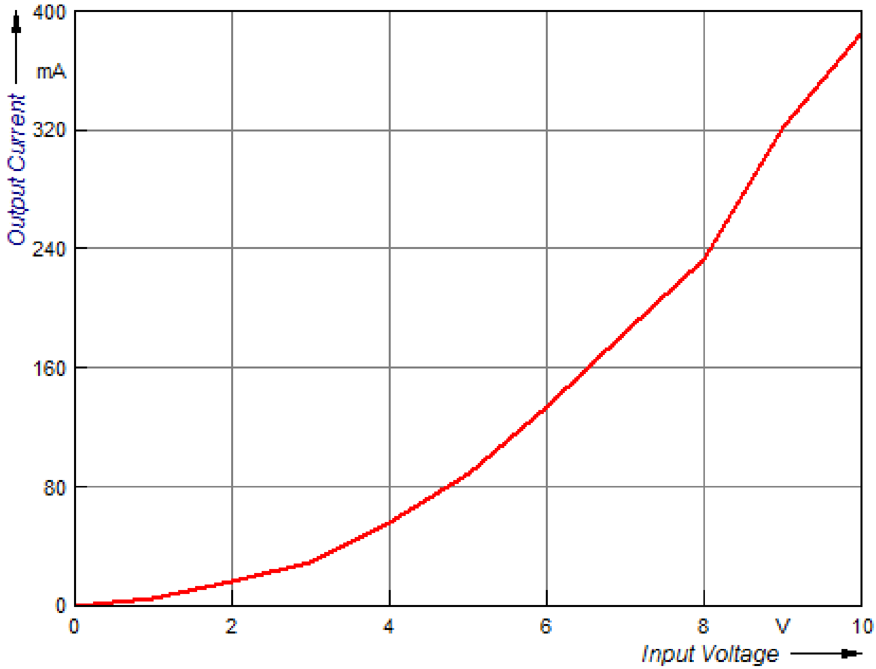
Selecting Data Ranges

You can access sections of data in the spreadsheet by referring to cell ranges. FlexPro offers an index operation 256 for this with which you can extract individual values or ranges of data from a data set or from the result of a formula. The formula `'Characteristic Curve'[0, 9]`, for example, extracts a partial signal with the first 10 points from the characteristic curve.

These index operations are automatically created by FlexPro if, for instance, during the creation of a diagram you do not select entire data sets, but only a cell range in a data set window. The cursors 188 offer a Copy Range command with which you can create interactive ranges from data sets.

Displaying Data Graphically

You can place diagrams onto a sheet to display data in a spreadsheet graphically. You therefore specify the cell ranges for the Y values and possibly also the X values for the curves in the diagram. The process is similar when working with FlexPro. You set up a diagram in the project database using the Diagram Wizard and, in the process, specify the data sets to be displayed as curves in the diagram. Later you can add more curves by simply dragging and dropping data sets or formulas into the diagram. When doing this, FlexPro scales and labels the axes automatically, and comments, units, etc. are taken from the data set header information.



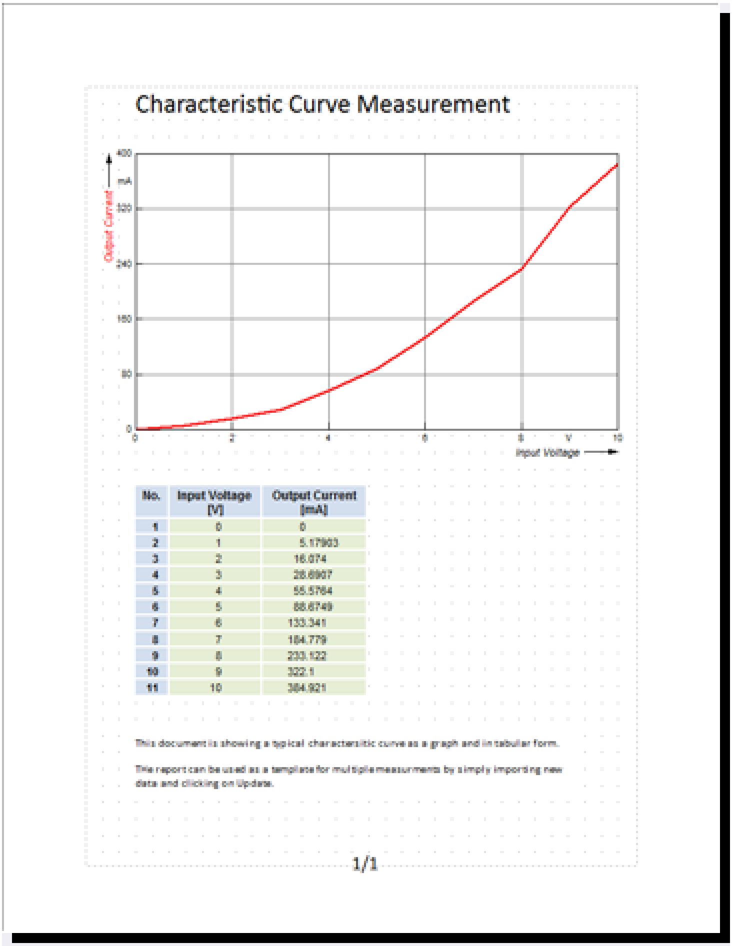
In addition to 2D and 3D diagrams 381, FlexPro offers you additional formatted column tables 437, cell tables 437 and texts 459 into which you can embed scalar values to display your results.

No.	Input Voltage [V]	Output Current [mA]
1	0	0
2	1	5.17903
3	2	16.074
4	3	28.6907
5	4	55.5764
6	5	88.6749
7	6	133.341
8	7	184.779
9	8	233.122
10	9	322.1
11	10	384.921

Documentation

To document the results, you usually set up a table in a spreadsheet where only the relevant data, diagrams, etc. are displayed. To do this, FlexPro has a powerful Document Editor that you can use to create illustrative, multi-page documents 470.

In addition to your own drawings and texts, you can insert links to diagrams, tables and texts that you have set up in the FlexPro project database into these documents. Alternatively, you can also embed these objects directly into the document without having them appear in the project database. Depending on your needs, FlexPro can automatically wrap diagrams and tables across several document pages. You create the document in the project database just like you would other objects.



Characteristic Curve Measurement

No.	Input Voltage [V]	Output Current [mA]
1	0	0
2	1	5.17903
3	2	16.074
4	3	28.6907
5	4	55.5764
6	5	88.6749
7	6	133.341
8	7	184.779
9	8	233.122
10	9	322.1
11	10	384.921

This document is showing a typical characteristic curve as a graph and in tabular form.

This report can be used as a template for multiple measurments by simply importing new data and clicking on Update.

1/1

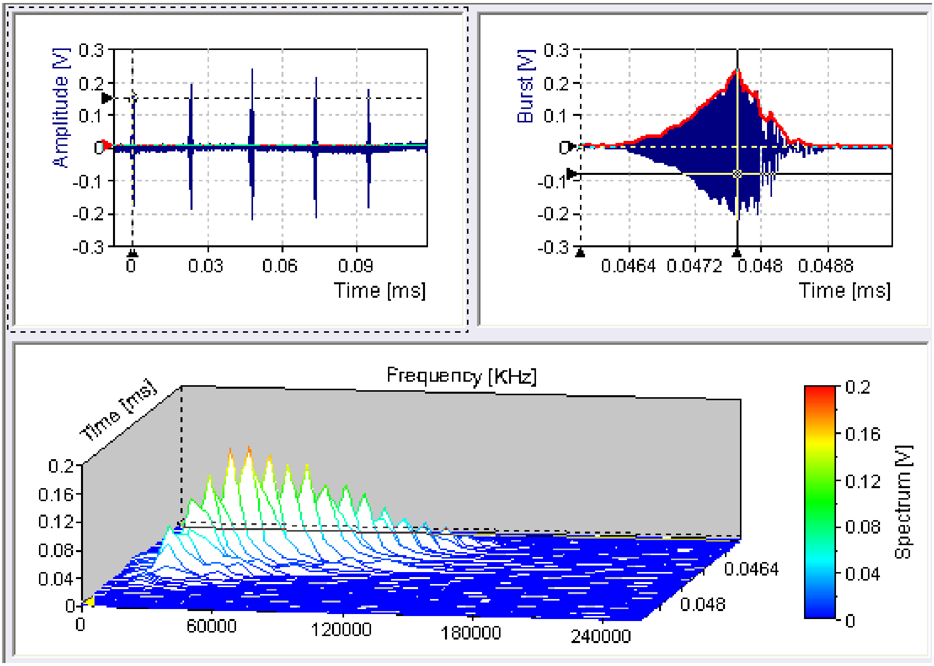
Importing and Exporting Graphics

If you write large reports, you may want to insert a diagram or table into a Word document, for example. This is no problem with FlexPro. You can create the diagrams in the project database and use your mouse to drag a copy into your Word document -- done! You can also insert graphics from other programs into the documents created using FlexPro.

Visually Analyzing Data

If you would like to interactively analyze data in a spreadsheet, you often have no choice but to look at the columns of values or the diagrams created. With FlexPro you can measure the curves in your diagram using cursors 188. Cursors work in 2D and 3D diagrams. You can zoom into sections, edit data, cut out areas and much more. The values under the cursors are displayed in a coordinates window. As an advanced FlexPro user, you can tailor the contents of the coordinates window to fit your needs.

FlexPro also allows you to synchronize cursors across several diagrams. To do this, use the worksheet 493. This is a window that can be split into up to 16 panes. You can insert diagrams into these panes and measure the curves of the diagrams with the cursors.



By the way: In addition, anything you can do in a worksheet can also be done in a document. However, the worksheet is often more practical for working onscreen.

Analysis Automation

To automate an evaluation with a spreadsheet, you usually have to write a program that controls the necessary steps.

FlexPro offers automation without any programming. Due to the fact that all calculation processes are permanently saved in formulas and analysis objects, you can easily apply them to several measurements, for example. Set up folders in the FlexPro project database where you want to save the data from a certain measurement. Before you update a document, for example, just activate230 the folder from which the data is to be taken. FlexPro then accesses the data from the activated folder and you obtain an evaluation of the selected data at the touch of a button.

In addition, FlexPro has a macro recorder that you can use to record operations as Visual Basic macros. With the automation interface, you can integrate FlexPro into applications and manage it remotely.

What's Next?

If you have not worked with FlexPro yet, we suggest that you look at the tutorial Getting to Know FlexPro in just 15 Minutes31.

2.2 About Data Analysis

The data analysis tutorials cover video analysis and test series analysis as well as different analysis procedures supported by FlexPro and various analysis options.

You can find them in FlexPro Online Help.

2.3 About Customizing FlexPro

The tutorials about customizing FlexPro cover the creation of presentation and analysis templates as well as expansion of the FPScript function libraries to include custom functions.

You can find them in FlexPro Online Help.

2.4 More Examples

The FlexPro setup includes additional sample project databases covering a wide range of topics.

The examples can be found in the folder `C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples` or `C:>Users>Public>Public Documents>Weisang>FlexPro><%VERSION_COMMERCIAL%>>Examples`.

Topics covered include:

- Presentation examples
- Analysis examples
- Examples of measurement series analyses
- Data query and document collection
- FPScript examples
- Order tracking examples
- Other examples (SPC, roundness, etc.)

3 Data Management

3.1 Project Database

FlexPro stores all objects that you create or import in a project database. The inner structure of a FlexPro project database resembles the file system on a hard drive. Instead of files on the hard drive, a FlexPro project database contains data sets, formulas, presentation objects, worksheets and documents that you need for your analysis. Just like on a hard drive, you can build up a hierarchy of folders in a FlexPro project database to organize your FlexPro objects.

How Project Databases are Stored

When you store a project database, you are offered two options for saving the project database. Your work with FlexPro does not depend on how you have set up your project databases. The distinction is made for purely technical reasons.

- In a single file

  This is the default storage method for a new project database. All FlexPro objects are combined into one file. The advantage of this format is that it is easier to use because only a single file is set up on the hard drive. If you would like to back up your project database or transfer it to another computer, you only have to copy a single file, e.g. ProjectDatabase.fpd. One disadvantage of this format is the slightly slower access speed, especially when you are working with large amounts of data. A further disadvantage is the higher risk of data loss in the event that the inner structure of the project database file becomes corrupted, such as due to a system failure while saving the file. FlexPro can recognize and delete defective objects within the file.

- In several folders and files

  This is an alternative storage option. In this case, the folder hierarchy which you build up in your project database is associated with folders on your hard drive. When you create the Project Database project database, in addition to the ProjectDatabase.FPD file, two subfolders called ProjectDatabase.DB and ProjectDatabase.TMP are added to the same folder. The project database folder hierarchy is stored under the ProjectDatabase.DB hard drive folder. All FlexPro objects are saved at the corresponding positions within the hierarchy as separate files. The ProjectDatabase.TMP folder is used to store temporary files and only exists while the ProjectDatabase.FPD is open.

  If you want to make a backup of your project database, you will have to make a copy of the file ProjectDatabase.FPD as well as of the ProjectDatabase.DB folder.

The folder-oriented storage system is recommended when you are working with very large amounts of data. You can increase FlexPro's performance by using this type of storage. In addition, there are no limitations in this format with regard to the total size of the project database and the maximum number of folders within it.

Note Changes to FlexPro objects below the DB directory should not be made if possible. All relevant operations are available in FlexPro.

Project Database Storage Options

You can determine whether calculated images of presentation objects and calculated results of formulas should be stored in the project database by setting this in the project database Properties dialog box. Elements that you store in the project database will not have to be recalculated the next time you open the project database. On the other hand, in certain circumstances considerably more space may be used on the hard drive when saving the data, particularly when saving the results of formulas.

Different Modes for Opening Project Databases

You can choose from three different modes to open a project database:

- Backup copy

  When a project database is opened, a backup copy is created. All further work is done in this copy. When the work is saved, the changes to the copy are transferred to the original database. The copy is deleted when the project database is closed. Please note that you can only use this mode if you have sufficient storage space.

  This is the default mode for opening project databases.

- Read-only

  Use this mode to open your project database if you want to be absolutely certain that no data in the project database can be changed inadvertently. Project databases that are write-protected, such as those on a CD, are automatically opened using this mode.

- Direct

  Use this mode if you would like to edit the project database directly or if you do not have enough space available to work with a backup copy. All changes are carried out directly in the project database without a request for confirmation.

Multiple users competing for access to one project database

If a user already has a project database open, additional users can work with the project database at the same time. In this case a copy of the original has to be created. The following table provides an overview of all possible scenarios:

Mode: project data base is already open	Mode: project database is to be opened		
	Backup copy	Read-only	Direct
With Backup	Possible, work carried out on copy of original	Possible, work carried out on copy of original	Possible, work carried out on copy of original
Read-Only	Possible, work carried out on copy of original	Possible	Possible, work carried out on copy of original
Direct	Not possible	Not possible	Not possible

Possible conflicts are automatically recognized, and the system asks whether a copy should be created. Please note that any changes made to the copy will have to be transferred manually to the original database once it is accessible.

Note: Project databases appearing in the list of recently used project databases in the File menu will be opened in the same mode as when they were last opened. A project database will first be closed if it is already open and is then opened again in a different mode. The system will then try to open it in the requested new mode.

Compatibility

FlexPro %VERSION_COMMERCIAL% is backward compatible with previous versions of FlexPro, which means that you can use project databases that you created with older versions of FlexPro in FlexPro %VERSION_COMMERCIAL%. If you open this type of project database in FlexPro %VERSION_COMMERCIAL%. It is then converted to the FlexPro %VERSION_COMMERCIAL% format and can no longer be opened in the previous version of FlexPro.

If you purchased FlexPro %VERSION_COMMERCIAL% as an update to a FlexPro 2021 license, you can continue to use FlexPro 2021. FlexPro 2021 also runs using the FlexPro %VERSION_COMMERCIAL% license. If you want to edit a FlexPro 2021 project database and you want to ensure that the database can continue to be read in FlexPro 2021, then you will need to edit it in FlexPro 2021.

Data Indexing

During indexing, the program runs through all of the objects in the project database and an SQL database is created, into which header data of the objects, such as names, comments, parameters or units, are included. In the case of data sets, additional statistical data such as data set sizes, maxima or mean values are calculated and stored. The index is saved in the project database file. Indexing runs in the background and the index is automatically updated when you modify or add objects. The index created in this way speeds up the search in the project database and allows flexible searching for data with the DataQuery 146 object.

Template Databases

FlexPro template databases are used to store presentation and analysis templates, macros, units and FPScript functions. The following are the project databases that FlexPro keeps open or opens briefly in order to import settings:

- Personal template database
  This is assigned to your Windows user account and is stored on your computer in your Windows user profile.
- Shared template databases (FlexPro Professional, Developer Suite only)
  As a FlexPro Professional and/or Developer Suite user, you can set up a list of shared project databases 74 that are usually located in network folders. These template databases are opened in FlexPro as read-only and only briefly during normal operation so that multiple users can use them at the same time. Exclusive access applies only when you use the Organizer dialog box. You cannot store macros in shared project databases. Use them to share text import schemas, analyses, unit tables and FPScript function libraries within a team.

Working with Project Databases

Creating a New Project Database

To create a new project database with a backup copy:

- Click File > New.

Project databases that you create this way are then stored in a temporary folder on your hard drive.

To create a new project database for direct editing:

1. Select File > New menu to create a new project database.
2. Select File > Save or click on the Save icon on the Quick Access Toolbar.
3. In the dialog box, select the folder, enter a name for the project database and select the storage format, such as In a single file.
4. Select File > Close.
5. Reopen the project database using Open from the File menu and select the Direct mode.

Searching Through a Project Database

You can search the active project database or parts of it for objects that fulfill particular criteria. Do the following:

1. In the Folders window, select the folder that you want to search.
2. Click Home[Edit] > Find. The Search in Project Database window appears.
3. Select the option Include subfolders in search if subfolders of the selected folder should also be searched.
4. Select the option Output in Find Results 2 if the objects found should be output to the Find Results 2 window instead of the Find Results 1 window.
5. Select the option Use search index if you want to speed up the search process. In this case, objects not yet calculated will be ignored during the search. To do this, Data Indexing 74 must be enabled for the project database. The search process may be slow if the Use search index option is not used. However, objects not yet calculated will be updated during the search.
6. The next step is to determine the criteria that the objects searched for should fulfill. You can combine multiple criteria by clicking on More and setting the desired logical operation. The search criterion can be combined with a Boolean AND or OR, where AND takes precedence over OR. The individual search items and search operations are listed below. The input search value is also covered below.
7. Click on New Search. The Find Results window is first cleared and then the objects found are displayed. If you want to start a new search while keeping the objects currently displayed in the Find Results window, then click Search instead of New Search.

8. You can cancel the search at any time by clicking on Stop.

The editing options in the Find Results window where the objects found are displayed mostly match those of the object list.

Search Item

The following table lists all search items supported by FlexPro. In the right-hand column you can see whether a search item is available for searching in the project database, for index-based searching in the project database, or for index-based searching in external files:

Criterion	Description	Available In
Absolute end time	Date and time of the last X value. The data set must either have an X component or it must be selected as the X component itself.	
Absolute start time	Date and time of the first X value. The data set must either have an X component or it must be selected as the X component itself.	
Sampling rate	The sampling rate of an equidistant data set or of its Y component.	
Sampling rate of X component	The sampling rate of a data set's X component.	
Sampling rate of Z component	The sampling rate of a data set's Z component.	
Date modified	Date and time of the last modification of	

Criterion	Description	Available In
	an object.	
Number of dimensions	The number of dimensions for a data set, e.g. 2 for a data matrix.	
Number of columns	The number of columns in a two-dimensional data set.	
Number of rows	The number of rows in a data set. In the case of a one-dimensional data set, this is its number of values.	
Author	The name of the data set author.	
Calculation description	The description of a calculation.	
Calculation unit	The unit of a calculation.	
Calculation formula	The FPScript code of a calculation.	
Calculation name	The name of a calculation entered on the Calculations tab of an object.	
Calculation name and value (numerical)	The name and value of a string calculation. Both attributes must match.	Index-based search only
Calculation name and value (string)	The name and value of a numerical calculation. Both attributes must match.	Index-based search only

Criterion	Description	Available In
Calculation value (numerical)	The value of a numerical calculation.	
Calculation value (string)	The value of a calculation that produces a string.	
Upper range limit	The upper range limit of a data set or of its Y component.	
Upper range limit of X component	The data upper range limit of a data set's X component.	
Upper range limit of Z component	The data upper range limit of a data set's Z component.	
Lower range limit	The data lower range limit of a data set or of its Y component.	
Lower range limit of X component	The data lower range limit of a data set's X component.	
Lower range unit of Z component	The data lower range limit of a data set's Z component.	
File extension	The file name extension of the file from which the data set originates, such as "fpf" for FlexPro text data.	File index only
File name	The name of the file from which the data set originates.	File index only
File type	The type of file from which the data set originates, as it is	File index only

Criterion	Description	Available In
	displayed in the FlexPro Import dialog box, e.g. "FlexPro Text Data (*.fpf)".	
Data structure	The data structure of a data set.	
Data type	The data type of a data set or of its Y component.	
Data type of X component	The data type of a data set's X component.	
Data type of Z component	The data type of a data set's Z component.	
Digital	The "Digital" attribute of a data set or of its Y component. (True, if the data set only contains zeros and ones.)	
Digital X component	The "Digital" attribute of a data set's X component.	
Digital Z component	The "Digital" attribute of a data set's Z component.	
Unit	The unit of a data set or of its Y component.	
Unit of the X component	The unit of a data set's X component.	
Unit of the Z component	The unit of a data set's Z component.	

Criterion	Description	Available In
Scalar value (numerical)	The numerical scalar value of a data set.	
Scalar value (string)	The scalar string value of a data set.	
Creation date	Date and time of the creation of an object.	File index only
Formula	The FPScript code of a formula or of an object derived from a formula.	
Quantity	The name of the physical quantity of a data set or of its Y component.	
Quantity of X component	The name of the physical quantity of a data set's X component.	
Quantity of Z component	The name of the physical quantity of a data set's Z component.	
Grandparent folder name	For an object in the project database, this is the name of the folder containing the folder in which the object is located. For an external data set, this is the name of the folder in which the file that contains the data set is located.	
Origin	The origin of a data set. Usually this is	

Criterion	Description	Available In
	the path name of the file from which it originates.	
Hyperlink	The hyperlink assigned to an object.	
Increment	The increment of an equidistant data set or of its Y component.	
X component increment	The increment of a data set's equidistant X component.	
Z component increment	The increment of a data set's equidistant Z component.	
Internal path	The path of an object inside a project database.	
Category	The object category assigned with the Categorize Object List command.	
Comments	The comments of an object or of a data set's Y component.	
Comments of X component	The comments of a data set's X component.	
Comments of Z component	The comments of a data set's Z component.	
Machine	The host name of the computer on which the file is	File index only

Criterion	Description	Available In
	stored from which a data set originates.	
Maximum	The maximum of a data set or of its Y component.	
Maximum of X component	The maximum of a data set's X component.	
Maximum of Z component	The maximum of a data set's Z component.	
Minimum	The minimum of a data set or of its Y component.	
Minimum of X component	The minimum of a data set's X component.	
Minimum of Z component	The minimum of a data set's Z component.	
Mean	The mean value of a data set or of its Y component.	
Mean value of X component	The mean value of a data set's X component.	
Mean value of Z component	The mean value of a data set's Z component.	
Name	The object name.	
Object path	The path of the file on the hard drive or of the folder in the project database where the object is stored.	

Criterion	Description	Available In
Object path and name	The object name and its path in the project database or on the hard disk.	
Object text	Any text appearing in an object.	Search in database only; no index
Object type	The object type.	
Folder path	The path of the folder on the hard drive on which the file is stored from which a data set originates.	File index only
Parameter name	The name of a parameter entered on the Parameters tab of an object.	
Parameter name and value (numerical)	The name and value of a string parameter. Both attributes must match.	Index-based search only
Parameter name and value (string)	The name and value of a numerical parameter. Both attributes must match.	Index-based search only
Parameter value (numerical)	The value of a numerical parameter.	
Parameter value (string)	The text of a string type parameter.	
Parameter unit*	The physical unit of a numerical parameter.	
Relative end time	The last X value of a data set. The data	

Criterion	Description	Available In
	set must either have an X component or it must be selected as the X component itself.	
Relative start time	The first X value of a data set. The data set must either have an X component or it must be selected as the X component itself.	
Standard deviation	The standard deviation of a data set or of its Y component.	
Standard deviation of X component	The standard deviation of a data set's X component.	
Standard deviation of Z component	The standard deviation of a data set's Z component.	
Void values	The "Contains void values" attribute of a data set or of its Y component.	
Void values in the X component	The "Contains void values" attribute of a data set's X component.	
Void values in the Z component	The "Contains void values" attribute of a data set's Z component.	
Parent folder name	For an object in the project database, this is the name of the folder that	

Criterion	Description	Available In
	contains the object. For an external data set, this is the name of the file that contains the data set (without the file extension).	
Folder path	For a data set on the hard drive, this is the path name of the folder in which the file that contains the data set is located.	Index-based search only
Timestamp	The timestamp of a data set or of its Y component.	
Timestamp of X component	The timestamp of a data set's X component.	
Timestamp of Z component	The timestamp of a data set's Z component.	

***** When searching for the unit, the search is case sensitive. When searching in other text, however, case does not matter.

Search Operations

Depending on the search criterion data type, you can choose one of the following search operations:

Data type	Operation	Search for object if...	Available In
Text	contains	The specified text appears in the searched text.	
	does not contain	The specified text does not appear in the searched text.	

Data type	Operation	Search for object if...	Available In
	is	The specified text is identical to the text in the searched text.	
	begins with	The text that is searched starts with the specified search text.	
	ends with	The text that is searched ends with the specified search text.	
	does not start with	The text that is searched does not start with the specified search text.	
	does not end with	The text that is searched does not end with the specified search text.	
	matches	The text to search matches the specified search pattern. See below for more details.	Search in database only; no index
Numeric	equal to	The value to find is identical to the reference value.	
	less than	The reference value is smaller than the value to find.	
	less than or equal to	The reference value is smaller than or equal to the value to find.	
	greater than	The reference value is greater than the value to find.	

Data type	Operation	Search for object if...	Available In
	greater than or equal to	The reference value is greater than or equal to the value to find.	
Date & Time	equal to	The reference date matches that of the date to find. The time is ignored in this comparison.	
	newer	The date and time occur after the date & time specified for the search.	
	older	The date and time occur before the date & time specified for the search.	
Object	is	The searched object matches the selected type.	
	is not	The searched object does not match the selected type.	
Attribute	true	If the attribute to check is present.	
	not true	If the attribute to check is not present.	

Entering Search Values

You can enter numerical values as floating point, time span or calendar time. For example:

```
123
-1.25
1.3E-10
12:13:14.56
4:12:13:14
```

```
23.1.2010 12:13:14.56
23.2.2010
2/23/2010 12:13:14.56
```

You can also enter complex values:

```
1+3i
(1 ; 3)
```

FlexPro forms its absolute value and uses it for the search.

You can also calculate the value using an FPScript expression. Start the formula with '=':

```
= 0.5 * PI
```

Please note that in FPScript code the period is always used as the decimal separator.

Pattern matching when searching for text

The matches search criterion helps you search for text patterns. The text pattern is defined here as what is called a regular expression. All text is found that matches the given regular expression. In regular expressions, special characters and sequences are used to represent a text pattern. The following table describes these characters and sequences and provides some examples.

Character	Description
^	Finds the beginning of the string.
$	Finds the end of the string.
*	Never finds the preceding character or finds it several times. For example, "zo*" finds "z" or "zoo".
+	Finds the preceding character once or several times. For example, "zo+" finds "zo" but not "z".
?	Finds the preceding character zero times or once. For example, "b?ig?" finds "ig" in "signal".
.	Finds every single character.
x|y	Finds either x or y. For example, "z|wood" finds "z" or "wood". "(z|w)oo" finds "zoo" or "wood".
{n}	n is a non-negative integer. Finds exactly n times. For example, "o{2}" does not find the "o" in "Bob" but it does find the first two o’s in "foooood".

Character	Description
{n,}	n is a non-negative integer. Finds at least n times. For example, "o{2,}" does not find the "o" in "Bob" but finds all o's in "foooood." "o{1,}" is equivalent to "o+". "o{0,}" is equivalent to "o*".
{n,m}	m and n are non-negative integers. Finds at least n and at most m times. For example, "o{1,3}" finds the first three o's in "fooooood." "o{0,1}" is equivalent to "o?".
[xyz]	A group of characters. Finds every one of the characters included. For example, "[abc]" finds the "a" in "signal".
[^xyz]	A group of excluded characters. Finds every character not included. For example, "[^abc]" finds the "s" in "signal".
[a-z]	A range of characters. Finds every character in the range. For example, "[a-z]" finds every alphabetical character from "a" to "z".
[^m-z]	A range of excluded characters. Finds every character not in this range. For example, "[m-z]" finds every character that does not lie between "m" and "z".
\b	Finds a word limit, i.e. the position between a word and a space. For example, "al\b" finds the "al" in "signal", but not the "al" in "signals".
\B	Finds a non-word limit. "ea*r\B" finds "ear" in "never early".
\d	Finds a number. Equivalent to [0-9].
\D	Finds a non-numeric character. Equivalent to [^0-9].
\s	Finds a space character. Equivalent to "[ ]".
\S	Finds everything but space characters. Equivalent to "[^]".
\w	Finds every word character including underline. Equivalent to "[A-Za-z0-9_]".
\W	Finds every character that does not belong to a word. Equivalent to "[^A-Za-z0-9_]".

Note: The search only takes into account text that is found for which the regular expression is a pattern for the complete text. "Signal+", for instance, finds the object with the name "Signal1", but not the object "Signal20". The regular expression "Signal+" only looks for the string "Signal2" in "Signal20".

Replacing Text in a Project Database

You can search selected objects, the active project database or parts of it for text to replace it:

a) Search and replace in specific objects

1. Select the objects in the object list or use the Home[Edit] > Find command to search for the objects in which you want to replace text and then select the objects found.
2. Click on Home[Edit] > Replace to open the Find and Replace dialog box.
3. If necessary, obtain further help in the dialog box.

b) Search and replace in a specific folder

1. Open the folder in the object list but do not select any objects in it.
2. Click on Home[Edit] > Replace to open the Find and Replace dialog box.
3. If necessary, obtain further help in the dialog box.

c) Search and replace in all objects in the project database

1. Click on Home[Edit] > Replace to open the Find and Replace dialog box.
2. In the Find and Replace dialog box, select Find and replace in complete project database.
3. If necessary, obtain further help in the dialog box.

Pattern matching when searching for text

The Use search pattern option is used to search for text patterns. The text pattern is defined here as what is called a regular expression. All text is found that matches the given regular expression. In regular expressions, special characters and sequences are used to represent a text pattern. The following table describes these characters and sequences and provides some examples.

Character	Description
^	Finds the beginning of the string.
$	Finds the end of the string.
*	Never finds the preceding character or finds it several times. For example, "zo*" finds "z" or "zoo".

Character	Description
+	Finds the preceding character once or several times. For example, "zo+" finds "zo" but not "z".
?	Finds the preceding character zero times or once. For example, "b?ig?" finds "ig" in "signal".
.	Finds every single character.
x|y	Finds either x or y. For example, "z|wood" finds "z" or "wood". "(z|w)oo" finds "zoo" or "wood".
{n}	n is a non-negative integer. Finds exactly n times. For example, "o{2}" does not find the "o" in "Bob" but it does find the first two o’s in "foooood".
{n,}	n is a non-negative integer. Finds at least n times. For example, "o{2,}" does not find the "o" in "Bob" but finds all o’s in "foooood." "o{1,}" is equivalent to "o+". "o{0,}" is equivalent to "o*".
{n,m}	m and n are non-negative integers. Finds at least n and at most m times. For example, "o{1,3}" finds the first three o’s in "fooooood." "o{0,1}" is equivalent to "o?".
[xyz]	A group of characters. Finds every one of the characters included. For example, "[abc]" finds the "a" in "signal".
[^xyz]	A group of excluded characters. Finds every character not included. For example, "[^abc]" finds the "s" in "signal".
[a-z]	A range of characters. Finds every character in the range. For example, "[a-z]" finds every alphabetical character from "a" to "z".
[^m-z]	A range of excluded characters. Finds every character not in this range. For example, "[m-z]" finds every character that does not lie between "m" and "z".
\b	Finds a word limit, i.e. the position between a word and a space. For example, "al\b" finds the "al" in "signal", but not the "al" in "signals".
\B	Finds a non-word limit. "ea*r\B" finds "ear" in "never early".
\d	Finds a number. Equivalent to [0-9].
\D	Finds a non-numeric character. Equivalent to [^0-9].
\s	Finds a space character. Equivalent to "[ ]".
\S	Finds everything but space characters. Equivalent to "[^]".
\w	Finds every word character including underline. Equivalent to "[A-Za-z0-9_]".

Character	Description
\W	Finds every character that does not belong to a word. Equivalent to "[^A-Za-z0-9_]".

See Also

Searching through a Project Database 50

Project Database 46

FlexPro Object 105

Object List 80

Defragmenting a Project Database

- File > Info > Defragment Project Database allows you to defragment a project database as long as it is not currently opened in FlexPro or marked as read-only.

Note: When defragmenting, free space is removed from the file and fragmented data is consolidated. A defragmented project database is usually more compact and allows for faster data access.

Saving the Workspace

To save the layout of the database window currently open in the workspace:

- Click on View[Workspace] > Save Workspace.

To save the layout of the window currently open in the workspace automatically every time the database is saved:

- Check the option View[Workspace] > Save Automatically.

Note: When opening a database, the workspace last saved will be restored automatically.

Optimizing FlexPro for Large Volumes of Data

FlexPro combines a high level of ease of use with high performance in a unique way and is ideal for a wide range of applications. In some areas, however, a higher level of ease of use and robustness can only be achieved when there is a loss of data throughput. FlexPro solves this problem by offering you different settings that make it possible for you to optimize the application for ease of use and data throughput. The default FlexPro settings are optimized for the highest level of ease of use and robustness. Please note the following if you want to optimize FlexPro for large volumes of data or for high data throughput.

Project Database Format and Operating Mode

The standard way that FlexPro operates is to work with a temporary copy of the FlexPro project database and to save it first where you specify by using the Save Project Database command. When working with large volumes of data, you should create the project database and save it in the In several folders and files storage format at the preferred location on your hard drive before actually starting to work with it, then open it in For direct editing mode:

1. Start FlexPro to obtain an empty project database.
2. Click on File > Save As.
3. In the Save As dialog box, select the hard drive folder in which the project database is to be saved.

Note You should save the project database to a local hard drive and not in a folder on your network.

---

4. Now enter the desired name in the File Name field.
5. Select In several folders and files as the storage format and then click Save.
6. Now click File > Open.
7. In the Open dialog box click on Browse and select the project database that you just saved.

Note The project database file extension is .fpd. This same folder contains a subfolder named the same, but with the file extension .db. This folder will later contain all objects stored as files.

---

8. For the Mode select Direct.

The project database has now been created and can be used.

Deactivating Global Link Correction When Renaming Objects

If you rename an object in a FlexPro project database, FlexPro searches the entire project database for objects that have references to this object in order to correct them. This can take a lot of time if the project database is large.

1. Click on File > Options.
2. Click on the Operation tab.
3. Uncheck the option Search the complete project database when looking for references. FlexPro then searches only folders in which the renamed object is located.

Deactivating the Preview

The FlexPro Preview option may be deactivated to increase the data throughput. To do this, click Close in the title bar of the Preview window.

Note You can use View[Task Window] > Show > Preview to reactivate the preview any time.

Preventing Recalculation of Images and Calculated Results

FlexPro can save images of diagrams and tables as well as calculated results of formulas in the project database. These are immediately available and do not have to be recalculated when you re-open the project database.

1. Click on File > Options.
2. In the Options dialog box, switch to the Project Database tab.
3. Select the options Save calculated images of presentation objects in the project database and Save calculated results of formulas in the project database.

Data Import

If you want to import data into the FlexPro project database, you should only select the necessary data sets and import them in Copy mode. If you are working with equidistantly sampled signals, you should import them as signals.

1. Click Data[Import from Files and Databases] > Binary Data.

2. In the Import dialog box in the Channel selection field select the option Manually and in the Data field select the option Copy. Under Data structure select the option Signals.
3. Click on Open to import the data.

If you are not sure which channels of the file you need when importing, we recommend using the Link mode. The individual channels are then only imported when you use them for the first time, e.g. in the Preview.

Using the Data Explorer for Large Data Volumes

For large volumes, we recommend using the FlexPro's Data Explorer Option. FlexPro then indexes the data sets in the background and you can quickly see a preview of the data without having to import it. You can use the Data Query to import the specific data you need for your analysis. See also Data Explorer Option 95.

Using Cursors for Large Data Sets

If you would like to use cursors for several channels of a large measurement, for instance, you can either use a single diagram with stacked Y axes or a worksheet object with one plane per diagram. The worksheet provides more options for zooming and scrolling the individual channels independently.

1. Select all of the data sets for which you want to use cursors.
2. Click on Insert[Containers] > Worksheet and click on the icon with the desired window layout.

The cursors are placed when positioning the mouse on the point closest to the position clicked. To do this, the distances for all XY value pairs for the curve are calculated, which causes a delay when working with large volumes of data. Therefore, you should switch to a mode that accounts for only the selected X position.

1. Click on the diagram for which you want to accelerate positioning using the mouse.
2. Right-click with your mouse on the worksheet and select Properties.
3. In the Properties dialog box click on the Cursor Settings (Active Diagram) tab.
4. In the Cursor positioning field select the option Use X position.
5. Repeat these steps for any additional diagrams.

When zooming to sections, you should select Spread X Range Between Cursors with rubber banding as the preferred method.

1. Click on the right side of the desired section to place the leading cursor at this position.
2. Press the TAB key to activate the origin cursor.
3. Click on the left side of the desired section.
4. Select Cursors[Zoom Curve] > Spread.

Note You should select the smallest section as possible. Subsequent scroll operations will then be faster. To scroll, you should use the arrow on the scrollbar for shorter moves. You should only use the slider on the scrollbar for long jumps on the curve.

Optimizing the Analysis

You should set up and test your analysis smaller data sets and then apply the prepared analysis to a large volume of data. For instance, you can import a relatively small data set, use FlexPro's Reduce function to decimate the data, or create a section of the data using the Index operator.

You can optimize the amount of run-time environment memory on your computer required for the FPScript analysis language.

1. Click on File > Options.
2. Click on the System Settings tab.
3. Under Maximum size of data sets in memory enter a value in megabytes that is higher than the size of the data sets that you are using. If you have, for instance, data sets with 10,000,000 64-bit floating point values, you should enter a value that is at least than 80 megabytes (1 megabyte = 1,048,576 bytes), since 8 bytes are needed for each 64-bit floating point value.
4. Under Maximum memory allocation for data sets enter a value that equals no more than 70 % of the computer's available memory.
5. Close FlexPro.
6. Now open Windows Task Manager by right-clicking on the taskbar and selecting Task Manager.
7. Now restart FlexPro and carry out a typical data analysis.
8. Follow the page file usage on the Performance tab of the Task Manager. The value should be displayed as the total value in the Physical Memory field and should not exceed more than 20 % of the memory available on your computer.

9. If the memory requirement limit is exceeded significantly, you should set the value in the Maximum memory allocation for data sets field back to a lower value.

Additional Optimization Options

If you have considered all of the points mentioned above, you can increase FlexPro's data throughput considerably. However, before you start analyzing and managing large volumes of data, you should check to see if the amount of data cannot be reduced when you acquire it. You should set the sampling rate for the measurement in accordance to the bandwidth of the signals to be measured and not to the capabilities of the measurement system. If your system has a good anti-aliasing filter, it generally does not make much sense to oversample the signal considerably. FlexPro also has no trouble jointly processing and displaying data with different sampling rates. You can therefore also capture slower signals, e.g., temperatures, with a lower sampling rate than faster signals.

Additional options for optimization are available when you select the appropriate graphic format. For instance, for a 3D data set, a waterfall chart is much faster than a surface chart. You should also always take into account whether the data must be presented in its full resolution or if the data can be decimated in advance.

Troubleshooting Project Databases

I deleted a lot of data from my project database, but the file size did not shrink.

The project database has to be defragmented to delete free space from the file.

- Click File > Info > Defragment Project Database after closing the project database.

Renaming objects in the project database takes a long time.

If you rename an object in a FlexPro project database, FlexPro searches the entire project database for objects that have references to this object in order to correct them. This can take a lot of time if the project database is large.

1. Click on File > Options.
2. Click on the Operation tab.
3. Uncheck the option Search the complete project database when looking for references. FlexPro then searches only folders in which the renamed object is located.

FlexPro uses too much memory.

You can optimize the memory requirements for FlexPro and the runtime environment for the FPScript analysis language for your computer.

1. Click on File > Options.
2. Click on the System Settings tab.
3. Under Maximum memory allocation for data sets enter a value that equals no more than 70 % of the computer's available memory.
4. Under Maximum size of data sets in memory enter a smaller value. Usually, there are several small and some large data sets in a project database. The value that you set here should be somewhere in the middle. Note that the size of a data set in bytes equals in the number of values multiplied by the number of bytes per value. A 64-bit floating point value takes up 8 bytes, for instance.
5. Under Maximum number of folders in memory enter a smaller value. Try out different values to find the setting that provides optimum performance. Lowering the value has effect only if there is no connection in the form of object references between the folders in the project database.
6. Under Maximum count of loaded preview images enter a smaller value. A preview image can take up to 120 KB of memory.

Upon reopening a database, calculations already performed are repeated.

You can ensure that FlexPro stores all calculated results in the project database. These then do not have to be recalculated in the next session.

1. Click File > Info > Project Database Properties.
2. Click on the General tab.
3. Select the options Save calculated images of presentation objects in the project database and Save calculated results of formulas in the project database.

FPScript formulas that provide a vague result, such as functions containing a user prompt or use the Noise function, are recalculated at each update and are not stored in the database.

The project database requires a considerable amount of hard drive space.

You can reduce the amount of hard drive space required by project databases by not saving optional elements on the hard drive.

1. Click File > Info > Project Database Properties.
2. Click on the General tab.
3. Uncheck the options Save calculated images of presentation objects in the project database and Save calculated results of formulas in the project database.

Setting Up a Shared Template Database

Only FlexPro Professional and FlexPro Developer Suite support shared template databases.

1. Click on File > Info > Organizer.
2. In the Organizer dialog box, click on the Templates tab.
3. Next, click on the Shared Template Databases button.
4. In the Shared Template Databases click on the New button to create a new entry.
5. Now enter the full path names to the template database in the field or click on the button after the field to select a database.

Setting Up Indexing of Project Databases

Changing Indexing for the Active Database

1. Click File > Info > Project Database Properties.
2. On the Indexing tab of the Properties dialog box and select the desired options.

Setting Up Indexing for Newly Created Databases

3. Select File > Options.
4. In the Options dialog box, switch to the Project Database tab.
5. Select the option Activate data query indexing under Default settings for newly created databases.

Setting the Save Options

To change the save options for the active database:

1. Click File > Info > Project Database Properties.
2. On the General tab of the Properties dialog box, select the desired options.

To set the save options for newly created databases:

1. Select File > Options.
2. In the Options dialog box, switch to the Project Database tab.
3. Select the relevant options under Default settings for newly created databases.

Exporting Project Databases as a Web in HTML Format

Using HTML Export, you can publish your evaluations as an HTML web within minutes. The HTML web created can then be viewed with any web browser such as Microsoft Internet Explorer or Firefox. You can export individual objects, folders and subfolders or entire project databases. Several exports can be brought together in a single web.

For each object to be exported, FlexPro generates one HTML page, or in the case of multi-page objects and large databases, several HTML pages. FlexPro can convert tables, text and data sets completely into HTML code. Other objects, such as diagrams and documents, are incorporated into the HTML pages as PNG images.

FlexPro can also include a table of contents in the web. A hierarchy view like in Windows Explorer is also possible.

HTML

HTML (HyperText Markup Language) is the standard "markup language" used for documents on the World Wide Web. HTML is developed by the World Wide Web Consortium. The HTML programming language uses tags to specify how page elements such as text and graphics are to be displayed in a web browser and how a web browser is to react to user actions such as activating hyperlinks via the keyboard or mouse. Most web browsers, in particular Microsoft Internet Explorer and Firefox, also recognize HTML tags, which are not yet part of the current standard. During export into HTML format, FlexPro automatically generates HTML files. Therefore, knowledge of the HTML programming language is not required.

HTML Templates

FlexPro generates the HTML pages for the individual objects based on an HTML template file. For each object type a separate template exists that you can customize. Fields in the template determine which object elements are to be integrated where.

The HTML templates are located in a subfolder within the FlexPro program folder. The correct name for a normal installation of FlexPro is C:\Program Files\Weisang\FlexPro <%VERSION_COMMERCIAL%>\ENU\HTML Templates. To customize the templates to fit your requirements, you should copy the complete folder and change the copied files only. In the second step of the HTML Export Wizard, you can specify the folder from which the templates are to be taken. You can use the following fields in the HTML templates:

Field	Meaning
<!-- %(Activated Folder Comments) -->	Comments on the activated subfolder of the folder containing the object to be exported.
<!-- %(Activated Folder Name) -->	Name of the activated subfolder of the folder containing the object to be exported.
<!-- %(Author) -->	Name of the data object author.
<!-- %(Begin Hyperlink) --> <!-- %(End Hyperlink) -->	The hyperlink specified on the General tab of the Properties dialog box of the object to be exported. The text enclosed by the two fields is highlighted as the hyperlink.
<!-- %(Comments) -->	Comments of the object to be exported.
<!-- %(Comments Or Name) -->	Comments or, if there are no comments, the name of the object to be exported.
<!-- %(Database Name) -->	Name of the project database to be exported.
<!-- %(Database Path) -->	Path of the project database to be exported.
<!-- %(Date) -->	Date of the export.
<!-- %(Folder Comments) -->	Comments of the folder containing the object to be exported.
<!-- %(Folder Name) -->	Name of the folder containing the object to be exported.
<!-- %(Name) -->	Name of the object to be exported.
<!-- %(Next Page) -->	For multi-page objects, this is a button to switch to the next page.

Field	Meaning
<!-- %(Number Of Pages) -->	The number of pages in a document object.
<!-- %(Object) -->	The HTML code of the object to be exported.
<!-- %(Origin) -->	The origin of the data of a data object.
<!-- %(Path) -->	Path of the object to be exported.
<!-- %(Page Number) -->	The current page number of a document object.
<!-- %(Previous Page) -->	For multi-page objects, this is a button to switch to the previous page.
<!-- %(Quantity) -->	The name of the physical quantity of a data object or its Y component.
<!-- %(QuantityX) -->	The name of the physical quantity of a data object's X component.
<!-- %(QuantityZ) -->	The name of the physical quantity of a data object's Z component.
<!-- %(Style) -->	Stylesheet definition that is required for the object.
<!-- %(Table Of Contents) -->	The HTML code of the table of contents.
<!-- %(Time) -->	Time of the export.
<!-- %(Unit) -->	The physical unit of a data object or its Y component.
<!-- %([Unit]) -->	The physical unit of a data object or its Y component, placed in brackets.
<!-- %((Unit)) -->	The physical unit of a data object or its Y component, placed in parentheses.
<!-- %(UnitX) -->	The physical unit of a data object's X component.
<!-- %([UnitX]) -->	The physical unit of a data object's X component, placed in brackets.
<!-- %((UnitX)) -->	The physical unit of a data object's X component, placed in parentheses.
<!-- %(UnitZ) -->	The physical unit of a data object's Z component.

Field	Meaning
`<!-- %([UnitZ]) -->`	The physical unit of a data object's Z component, placed in brackets.
`<!-- %((UnitZ)) -->`	The physical unit of a data object's Z component, placed in parentheses.
`<!-- %(User) -->`	Name of the FlexPro user.

Exporting a Complete Project Database

1. Select the root folder of the project database displayed in the first line of the Folders window.
2. Click Data[Export] > Export.
3. Navigate to an empty folder where the web page is to be created.
4. Under File Type select HTML Document for the format.
5. Click on Save to start the HTML Export Wizard, which will guide you through the remaining steps.

Exporting Individual Project Database Objects

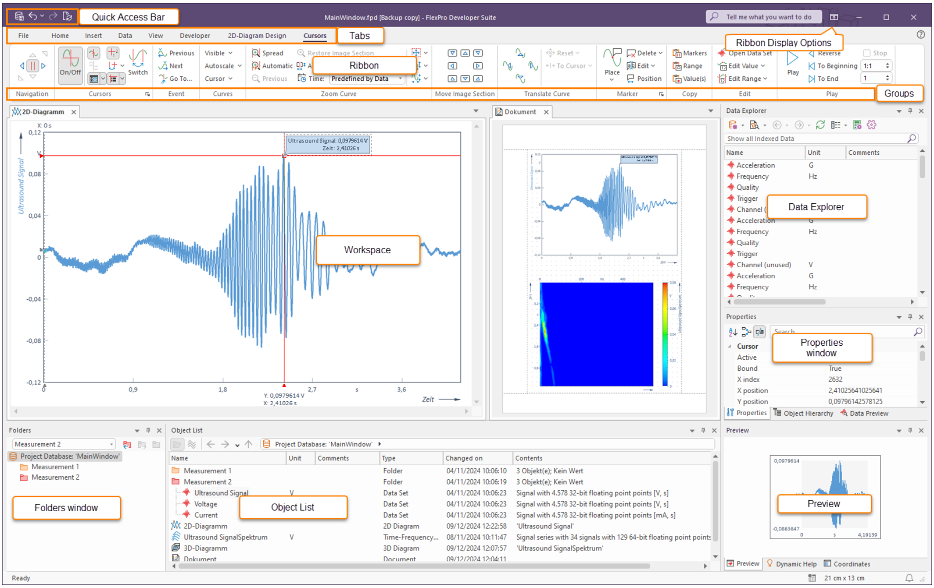
1. Select all of the objects in the object list that you want to export to the file. Alternatively, you can also select an object for export that is in a worksheet or document.
2. Click Data[Export] > Export.
3. Navigate to an empty folder where the web page is to be created.
4. Under File Type select HTML Document for the format.
5. Click on Save to start the HTML Export Wizard, which will guide you through the remaining steps.

Exporting Multiple Project Databases to a Single Web Document

1. Select the root folder of the first project database to be exported. The root folder is displayed in the first line of the Folders window.
2. Click Data[Export] > Export.
3. Navigate to an empty folder where the web page is to be created.
4. Under File Type select HTML Document for the format.

5. Click on Save to start the HTML Export Wizard, which will guide you through the remaining steps.
6. Select the root folder of the next project database to be exported.
7. Click Data[Export] > Export.
8. Click on Save without changing the folder beforehand.
9. You can now change settings in the HTML Export Wizard, if necessary. However, you should not change the settings for the table of contents.
10. Repeat steps 6 through 9 for each additional project database.

3.2 FlexPro Main Window



The main FlexPro window shows the data of an open project database and offers you options for navigating and searching within the project database as well as editing objects in the project database. The main menu consists of the following components:

Ribbon

The ribbon offers you convenient access to the FlexPro commands. It is divided into several tabs. Only one tab is active at a time. The example above shows the Cursors tab. You can switch tabs by clicking on the tab title. The commands are divided into groups whose names appear below the icons. The FlexPro documentation uses the following syntax to refer to a command: Tab[Group] > Command, e.g. Cursors[Event] > Next. Some icons open menus from which you can choose an entry. This can be identified by a small arrow next to or under the icon; for instance: Cursors[Marker] > Place > Point Marker.

The ribbon can also be minimized so that only the tabs are displayed in order to have more space for the work area.

Quick Access Toolbar

The customizable Quick Access Bar lets you access important commands without switching tabs.

In the field Tell me what you want to do? you can search for commands by their name and execute them directly.

Folders

This window, which is usually displayed on the left-hand side of the FlexPro main window, shows all folders and all data objects that provide a list.

Object List

The object list has two views: Objects and List Elements.

If you select a folder in the Folders window, the object list switches to the Objects view and shows all objects and subfolders in this folder.

If you select a list in the Folders window, the object list switches to the List Elements view and shows the elements in the list. In the List Elements view, you can select list elements individually, e.g. to create a diagram for them. If you want to use the complete list selected in the Folders window for an operation, simply switch to the Objects view. The list is now already displayed there as selected.

As with Windows Explorer, you can move, copy, rename and delete objects in the Objects view. What is different from Windows Explorer, however, is the fact that in the object list you can also re-sort data sets; for instance, you can stipulate a certain order when creating a diagram.

You can divide the objects in the object list into categories to later be able to display only the objects from one or more categories.

In the object list, only the content of the folder selected in the Folders window is displayed. You can display the content of additional folders by opening these in the workspace.

When you use your mouse to point to the name of an object in the object list, a popup window appears for this object. If the object is a data object, all numerical properties of the object's data, such as extreme values or data structure, are displayed.

Object Hierarchy

This window displays the objects of the folder selected in the Folders window hierarchically. You can therefore simply reconstruct which objects are linked directly or indirectly with a particular object. The display is clearer than the object list, since only the objects not used in any other object, such as your documents or worksheets, appear at the very top level. When you select an object in the hierarchy, the selection represents not only the object itself, but also all of the subordinate objects that it uses. This makes it easier for you to copy complete object networks, for instance.

Event Log

All errors and warnings that occur when updating analyses and documents are displayed in the Event Log with the time of their occurrence and the name of the source. Typical events include: data sets not found, syntax errors in formulas or calculation errors, such as a division by zero. The Event Log supports two modes: Event History and Troubleshooting. Event History logs all events chronologically, while Troubleshooting deletes old messages before each update so that only errors that still remain are displayed.

Breakpoints

The breakpoints that you have set in formulas for debugging are displayed in this window.

Call Stack

When debugging formulas with the debugger, all formulas called up to the current execution position are displayed in this window.

Find Results 1 & 2

These windows show the objects found when searching in the project database.

Watch

This window displays the result of the current statement as well as the contents of selected variables.

Preview

The Preview window allows you to quickly view the content of a selected object.

Data Preview

Data Preview is an advanced preview window specifically designed for data sets. The data set just selected is displayed there as a curve. You can use data cursors to measure the data or enlarge a section.

Properties

The powerful Properties window allows you to quickly edit the properties of selected objects.

ASAM ODS Data Source

This window is only available when the ASAM ODS Data Import Option 170 is installed. It displays the data from a linked ASAM ODS data source and lets you navigate and search within the hierarchical structure as well as import measured data from the ASAM ODS data source to the FlexPro project database.

Data Explorer

The Data Explorer indexes the data on your hard drive. You can use it to quickly search through your data, view a preview with statistics and import data. This window is only available when the Data Explorer option 95 is installed. Alternatively, Data Explorer displays the files on your hard disk in a similar way as Windows Explorer.

Dynamic Help

This window displays selected sections and statements related to the object that you are currently processing or to the window in which you are currently working. Simply click on a help topic to display it.

Coordinates

This window shows the coordinates while you are using the data cursors in diagrams.

Workspace

This is the core area of the main window where FlexPro displays the open objects in the project database on tabs. You can divide the workspace horizontally and vertically into tab groups to display windows side by side.

Context Menu

A context menu allows you to choose the most important commands for the current selection or the current working environment. You obtain a context menu for a certain object by clicking on the object with the right mouse button.

Working with Windows

Showing a Window

To display a window in the workspace:

- Click on the window tab or click View[Window] > Switch Windows and select the window from the list.

To display a task window:

- Click View[Task Windows] > Show and select the window from the list.

Activating a Window

1. Press and hold the CTRL key while pressing the TAB key to view a selection window.
2. Now press the TAB key multiple times until you see the window you want to view. Now release the CTRL key to open the selected window.

Closing a Window

- Click on Close to the right of the window title bar.

Saving the Workspace

If you want the current window arrangement in the workspace to be automatically saved each time the project database is saved:

- Select the option View[Workspace] > Save Automatically.

If you want to save a specific window arrangement in the project database:

1. Uncheck the option View[Workspace] > Save Automatically.
2. Click View[Workspace] > Save Workspace.

Hiding an Anchored Task Window

- Select the option Auto Hide to the right of the window title bar.

Notes

- If the task window is hidden automatically, only its title appears on the edge of the main window. If you use your mouse to point to this title, the window appears temporarily.
- To display the window permanently, uncheck Auto Hide.

Moving a Task Window

Freely Positioning an Anchored Task Window or Tab Group

1. Point to the title bar of the anchored window or tab group and press your left mouse button.
2. Drag the window to a location where no positioning arrow appears and release the mouse button.

Freely Positioning a Task Window in a Tab Group

1. Point to the window's tab group and press the left mouse button.

2. Drag the window to a location where no positioning arrow appears and release the mouse button.

Anchoring a Floating Task Window to the Location Where It Was Last Anchored

- Double-click on the window's title bar.

Anchoring a Task Window or Tab Group

1. Point to the title bar of the anchored window or tab group and press your left mouse button.
2. Drag the window to a location where the positioning arrow appears. The resulting anchor position is now highlighted.
3. Release the mouse button.

Saving a Custom Window Arrangement

1. Click View[Task Windows] > Save Arrangement.
2. In the dialog box that appears, enter a name for the arrangement and close it with OK.

Deleting a Custom Window Arrangement

1. Open the menu View[Task Windows] > Restore Arrangement.
2. Right-click the custom window arrangement that you want to delete.
3. Select Delete from the context menu that appears.

Restoring a Window Arrangement

- Open the View[Task Windows] > Restore Arrangement menu and select one of the predefined or saved arrangements.

Working with the Object List

Selecting the Folder/List to be Displayed

To display a folder in the object list, click on the name of the desired folder in the Folders window. If the name of the folder to be displayed is not visible because it is located in a tree of the folder hierarchy that is not displayed, expand the corresponding tree by clicking on the '+' icons in front of the parent folders until the folder to be displayed becomes visible.

If the folder immediately above the folder to be displayed is currently displayed in the object list, double-click on the desired folder name in the object list to display the folder.

To display a folder in the object list, click on the name of the desired folder in the Folders window. The object list then switches from the Objects view to the List Elements view and you can select individual elements. You can switch between the two views using the Objects and List Elements tabs in the object list's toolbar.

Note If a formula that returns a list as a result is not displayed in the Folders window, this is because the formula has not yet been calculated. You can fix this by selecting the formula in the Objects view of the object list and then clicking on Home > Update > Update. A list can in turn contain lists. These sub-lists are then also displayed in the Folders window and you can descend in the list hierarchy by clicking on the '+' symbols.

Selecting Objects/List Elements

You can use the known procedures from Windows Explorer in FlexPro to select objects or list elements. The following table offers a summary and presents two FlexPro-specific selection options:

What would you like to select?	Procedure
An object/list element	Click on the desired object or move the selection onto the object using the arrow keys.
Multiple objects/list elements	Click on the desired objects while holding down the CTRL key.
Neighboring objects/list Elements	Click on the first object to be selected. Press the SHIFT key and then click on the last object to be selected.
All	Open the list box under Home[Edit] > Select and choose Select all or use the CTRL+A shortcut.

What would you like to select?	Procedure
Objects/list elements not selected	Open the list box under Home[Edit] > Select and select Invert Selection.
Linked objects	Open the list box under Home[Edit] > Select and select Linked Objects. All objects in the folder displayed that the selected object requires for evaluation or display are also selected. This command makes it easier to move or copy complex analyses and presentations.

Note You can also select data sets in the Names line of the Data view of an open folder. The procedure is the same as described above.

Selecting the Display Type

For the object list, select View[Object List] > Change View or use the context menu to select one of four display options:

Command	Display
Large Icons	The objects in the folder to be displayed are arranged in rows, displaying the name of the object under a large icon that clarifies the object type.
Small Icons	Here, the objects are also arranged in rows, but a smaller icon is used, which is placed in front of the object name.
List	With this display option, the objects are arranged in columns, with a small icon and then the object names.
Details	The detail view uses a line for each object in the folder displayed. After the icon and the object name, the line contains the object comments, its file type in plain text, the creation date and a description of the contents of the object.

Sorting the Object List

The list of objects displayed can be sorted in the Details view.

By clicking on one of the column headers at the top of the object list, the rows of the list view are sorted in ascending order according to the criterion selected. The sorting column is represented by a triangle in the column header. If you then click again on the button of the current sorting column, this reverses the sorting order, which is then represented by an inverted triangle on the button.

In the name column there is a third mode in which no triangle is displayed. The objects are then displayed in the order defined by the user. In this mode you can sort the objects manually.

Note Manual sort is not available if you display the objects in groups.

Manually Sorting the Object List

To re-sort the objects displayed in the Details view of the object list, do the following:

1. Click on the Name column title in the object list until no triangle appears. This chooses the user-defined order.
2. Select a number of objects that you want to move to a different place.
3. Use the mouse to drag the selected objects to the position where you want to insert them and drop them there. You can insert the objects between two objects, before the top object or under the bottom object.

or

1. Click on the Name column title in the Object List until no triangle appears.
2. Select the objects you want to move.
3. Press the UP button or the DOWN button while holding down the CTRL and ALT buttons to move the objects up or down by one position.

Notes

- When creating diagrams or tables, the data sets selected are assigned in order from the top to the bottom to the curves or columns respectively. If you place the data sets in the correct order from the beginning, you will avoid having to make corrections in the diagram or table later.
- Manual sort is not available if you display the objects in groups.

Setting the Column Width

You can change the column width of the individual columns in the object list's Details view by moving the mouse cursor in the column header area onto the dividing lines between the buttons. The mouse cursor then changes shape. The following actions are then possible:

- You can individually set the width of the column by using your left mouse button to click and then drag the line to the right or left.
- If you double-click, the width of the selected column is automatically adjusted so that all of the entries in this column can be read completely.

Displaying Objects from Particular Categories

To display only the objects of a particular category in the object list:

- Select the desired category in the View[Object List] > Show Object Categories selection box.

To display objects from multiple categories in the object list:

1. Select Multiple categories in the selection box View[Object List] > Show Object Categories.
2. In the dialog box that appears, select the categories to be displayed.

To display objects from all categories in the object list:

- Select All in the selection box View[Object List] > Show Object Categories.

Displaying Several Folders

In the object list, only the content of the folder selected in the Folders window is displayed. You can display the content of additional folders by opening these in the workspace. To do this, double-click on a folder in the Folders window that you want to open, or select a folder from the object list and choose Open from the context menu.

Navigating in the Project Database

With the new navigation bar in the object list, you can easily navigate from folder to folder in the project database.

To display the parent folder of the folder shown in the object list:

- Click on Upwards in the navigation bar of the object list or press the ALT + UP key.

To display the previously displayed folder:

- Click on Back in the navigation bar of the object list or press the ALT + LEFT key.

To display the folder shown afterwards:

- Click on Forward in the navigation bar of the object list or press the ALT + RIGHT key.

To display any previously displayed folder:

- Click on the Recent button in the navigation bar of the object list and select a folder from the list.

To display a folder that is located in the path of the currently displayed folder:

- In the navigation bar of the object list, click on the name of the parent folder that you want to display.

To display a neighboring folder of a parent folder:

- In the navigation bar of the object list, click on the arrow behind the parent folder to open a list box in which you can select the desired neighbor of the folder.

Displaying the Folder of an Open Object

To display the parent folder of an object shown in a window:

1. Right-click with your mouse on the object window tab.
2. In the context menu that appears, select Display Folder with this Object.

Activating and Deactivating a Subfolder

To activate a subfolder:

- Select the subfolder from the Activate or deactivate folder list box at the top of the folder window.

To deactivate a subfolder:

- Select (No activated subfolder) from the Activate or deactivate subfolder list box.

Displaying the Activated Subfolder

You can display the content of the active subfolder 230 in the object list. To do this, select the option View[Object List] > Show Activated Subfolder. In the Details view, the contents of the activated folder will be displayed indented below the folder. You can work normally with the displayed objects. For instance, you can select data sets to create a diagram.

When accessing objects, objects from the activated subfolder override objects of the same name in the main folder. You can hide these overlapped objects by selecting the option View[Object List] > Hide Overlapped Objects. If you do not hide the objects, their icons will be displayed with less contrast in the object list.

Selecting the Columns to be Displayed

In the object list's Details view, you can add or remove columns.

1. Right-click with your mouse on a column header.
2. In the context menu that appears, select a column to be displayed, or deselect a column that is to be removed.

Note A newly added column appears to the right of the list. You can drag it with the mouse to another location.

Changing the Column Order

In the object list's Details view, you can change the column order.

- To do this, click on a column header and drag it to the desired location while holding down your mouse button.

Displaying Objects in Groups

In the Details view, the list of objects displayed can be grouped according to the currently selected sorting column.

1. Right-click with your mouse in the object list.
2. In the context menu select the option Show in Groups.

Working with the Object Hierarchy

Displaying Linked Objects

- In the object hierarchy, click on the plus symbol of the object with the linked objects that you want to view.
- You can view the objects linked indirectly with the object by opening additional levels the same way.

Displaying the Purpose of Linked Objects

1. In the Object Hierarchy window click on the > sign to the left of an object symbol to view its linked objects.
2. Now point with your mouse on a linked object to view the popup window showing the purpose of the object.

Selecting an Object and Linked Objects

- Click on the name of the object in the Object Hierarchy window to select it.

Note: You can select only one object at a time in the object hierarchy. However, by selecting the object, all objects dependent on the selected object are selected as well. This makes it possible to copy or move a selected object while simultaneously copying or moving all linked objects with it.

Working with the Properties Window

Editing Properties

1. Select all objects for which you want to edit a particular property, for instance, the labels for the X and Y axes of a diagram or several data sets and formulas in the object list. You can select several objects in presentation objects by holding down the CTRL key and clicking on them. With the ALT key pressed, you can select all elements of the same type, such as the symbols on all curves.
2. Click on the edit box in the Properties window for the property you want to edit. If you are working with a numerical value or text, you can enter the value or text directly. If you are working with a drop down list box, then you can open a list and make a selection. Otherwise, a button is displayed to the right of the field so that you can open a dialog box for editing.
3. You can finish your input using the ENTER key. The change will affect all selected objects containing this property.

Notes

- Placing your mouse pointer on the name of a property displays the property description.
- If the edit box for the selected property appears empty, then you are either working with an empty text field or the property occurs in several objects with different content. In any case, your input is entered into all selected objects containing the edited property.

Showing or Hiding Properties Not Available in All Objects

If you have selected several different objects, e.g. data sets and formulas in the object list or curves or axes in a diagram, you can show or hide properties that are not common to all objects:

- Select the option Show Only Common Properties from the Properties window toolbar to hide these properties, or uncheck the option to show them.

Note: Properties that occur in several objects and that have different content are displayed with an empty edit box. If you edit this type of property, then this change will affect all selected objects.

Showing or Hiding a Properties Group

- The properties are displayed in groups. To show or hide a properties group, click on the triangle icon in the Properties window to the left of the grouping icon.

Showing or Hiding Properties with Read-Only Access

Properties with read-only access appear in gray. You can show or hide these types of properties:

- Select the option Show Only Changeable Properties from the Properties window toolbar to hide these properties, or uncheck the option to show them.

Sorting or Displaying in Groups of Properties

You can display the properties shown in the Properties window either as a sorted list or in groups.

- Select the option Sort Properties from the Properties window toolbar to display the properties as a sorted list, or uncheck the option to show them in one or more named groups.

Note: The groups correspond to the elements of the selected object, e.g. the connection line or symbols of a selected curve.

Working with the Event Log

Fixing the Cause of Events

- Double-click on the event for which you want to fix the cause. The object that triggered the event opens, and you can fix the error.

Clearing the Event Log

- Click on Clear Event Log on the Event Log toolbar.

Customizing Event Handling

You can customize event handling in FlexPro to meet your specific requirements. For example, you can specify a maximum number of events that should be collected in the log and whether the log should be saved in the project database.

- Click on Properties on the Event Log toolbar.

3.3 Data Explorer Option

The Data Explorer option gives you the ability to display the contents of files without having to import the data itself. You can also preview single channels as well as statistical quantities, such as maximum, minimum and mean.

You can easily search for data set properties, preselecting what you need for file import. The results can then easily be imported 167 into FlexPro.

You can use the DataQuery 146 object to find and browse through data on your hard drive and then further process this data directly without having to import it.

File Indexing

File indexing in FlexPro sifts through your hard drive data and prepares an index database, which is used by Data Explorer and the DataQuery object to find data. FlexPro not only stores all header information in the database, such as units, comments and parameters, but also information about the data itself, such as the value count, sampling rate, mean or a thumbnail for the data preview.

Before you can use Data Explorer, you need to set up File Indexing 99. To do this, specify the folders to be searched on your hard drive and the data formats to be included in the search. You can also specify when FlexPro should perform indexing and at what time intervals the index database should be updated.

If you enable file indexing, FlexPro will close, but then FlexPro will remain open in the background so that it can index the files. This can be identified by the FlexPro icon that appears in the Windows Taskbar system tray. If you right-click on the icon, a menu appears which you can use to quit FlexPro, for instance.

Note If you copy or delete files in a hard disk folder that is set up for indexing, FlexPro automatically adds these files to or removes them from the index. However, FlexPro must be running during this procedure.

File Indexing in Client/Server Mode

In addition to using file indexing for your personal database, you can also use it for larger data archives that are on a server. In this case, a FlexPro Professional instance is installed on the server and will handle the indexing. The data collected during indexing is stored in a Microsoft SQL Server database, which must be provided on the server (this license is not included with FlexPro). FlexPro Professional users can access the index database via an ODBC database connection and by sharing the database folders.

Note While FlexPro stores the indexed data for personal databases in an SQLite database, a multi-user database is required for client/server mode. Currently only Microsoft SQL Server version 2008 or higher is supported.

Data Explorer Views

Data Explorer features two different types of views:

The All Files views are similar to Windows Explorer. All files and folders on your hard drive and on the network are displayed. You can import files using drag-and-drop or copy them to the object list using the Data Explorer clipboard.

The Indexed Files views show you only the files that are detected and indexed by FlexPro. In this view you can see individual data channels in the measurement data files. If you click on a data channel, detailed information will be displayed in the Properties window and in the preview.

You can view both views as a list or as a tree in a hierarchical structure.

Preview and Properties

If you use your mouse to point to a channel displayed in Data Explorer, a popup window will with detailed information will appear. If you select the channel in Data Explorer, a preview and properties will be displayed.

Search Filters

Search filters allow you to specify which data to display in Data Explorer. In addition to the default search filter, you can also define additional search filters which FlexPro can store in the project database or in a template database. You can combine multiple search criteria in a single search filter. For instance, you can search for all data sets whose names start with "Axial", have a unit of m/s² and a maximum exceeding 10.

You can carry out simple filtering by object name without having to define a search filter. Just enter part of the name in the input box under the Data Explorer toolbar. Only objects whose names contain the input text are displayed.

Calculations

Calculations for Data Sets

These are small FPScript formulas that you can use to calculate quantities for the indexed data sets. For instance, a "span" calculation would look like the following:

```
Range(data)
```

Set up the calculations before you start indexing the data. You can assign a search filer to each calculation. The particular calculation is evaluated only for data sets that pass the search filter.

Calculations can return scalar values or any other data structures as the result. You set up a calculation, enter whether it will be considered an additional data set or whether it will be assigned as an attribute to a data set for which you have made a calculation. The latter is allowed only for calculations that result in a scalar value.

Calculations that are to be considered as an additional data set appear in Data Explorer as additional entries, where the name is formed from the name of the data set and the name of the calculation. For example: "DatasetSpan".

If you later import an indexed data set to which you have assigned calculations as an attribute, then these calculations will be entered in the Header Information 126 of the data set. Using the FPScript code `Dataset.Calculations("Span")` provides the ability to access to a "Span" calculation stored in the data set. Since the calculations are stored as formulas, they always produce the correct result even if you change the data set data.

You can use calculations in data queries 146 to search for data. For instance, you can define a search filter that provides all data sets with the unit Pa and a span greater than 10.

Calculations for Files

These are not for a single data set, but for a complete file. Therefore, you cannot assign them as an attribute to a data set, but instead can only store them as additional data sets. During the calculation you have access to all channels in the currently indexed file and can therefore carry out cross-channel calculations. For instance, an "output" calculation would look like the following:

```
Current * Voltage
```

Using Data Explorer

Changing the Data Explorer View

You can view files on your hard disk or indexed data in the Data Explorer task window.

Note: Indexed data can only be displayed if data indexing has been enabled and the files have already been indexed.

Changing Views

From the Change View dropdown list box in the toolbar of the Data Explorer window, you can choose one of the following views:

View	Shows
List view (indexed data sets)	Shows indexed data sets as a list.
List view (indexed files)	Displays the indexed files as a list.
Tree View of Indexed Files	Displays the indexed files and the data sets that they contain in a tree view.
List View of Hard Disk Folder	Displays the contents of a folder on the hard disk as a list.
Tree View of Hard Disk Folder	Displays the contents of a hard disk folder and its subfolders hierarchically.

In the case of indexed data views, use a search filter that you create or select using the Search command on the Data Explorer toolbar to determine whether all indexed data sets or only selected data sets will be displayed.

The list and tree views of a hard disk folder are similar to Windows Explorer. Use these views to import non-indexed data into FlexPro using drag-and-drop.

Customizing the List View

You can display different properties for the indexed data sets in the list view. Right-click with your mouse on the column title to open a menu where you can choose the attributes to be displayed.

Setting Up File Indexing

1. Click File > Options or click on the Options button on the toolbar of the Data Explorer window.
2. In the Options dialog box, click on the File Indexing tab.
3. Select the desired options.

Database & Mode of Operation

Use Database Setup to set up the database connection for the index database. To index your personal data, choose SQLite as the default setting. See also: Setting Up Indexing in Client/Server Mode 100.

Calculations Setup is used to set up calculations that are executed for particular data sets and which are to be assigned to them in the index. See also: Setting Up Calculations 104.

From the Mode dropdown list, select the indexing mode for data files on the hard disk.

- Select No indexing to disable indexing.
- Select Index in stand-by mode to enable indexing as long as FlexPro is not used interactively. Use this option if indexing interferes with your work in FlexPro.
- Select Always index to enable indexing to run in the background even while you are working with FlexPro. Use this option if you want to complete the indexing as quickly as possible.

Several indexing options are available:

- Restart indexing: Restarts indexing, i.e. the program searches through all the data again from the beginning, but does not delete the index database.
- Reset Indexing: Discards the index database and restarts indexing from the beginning.

Use the Indexing Details... button to open a dialog box where you can monitor the current indexing process. Here you can also specify whether errors that occur during indexing are to be written to the file.

In the Indexing repeat rate field, specify how often the program should search for new files to index on the hard drive. This setting is for security purposes only. Usually changes to the file system (added, renamed, deleted files and folders) are detected automatically and indexing is restarted.

Use the option Activate file indexing when Windows starts up to specify whether indexing should start automatically when Windows starts up.

File Types To Index

Here you choose which file types known to FlexPro should be included when indexing, or select the Index all supported file types option.

Assign File Types

Particularly in the case of text data, the file name extension may apply to more than one text import schema, making it difficult to assign the right one to the text file. FlexPro can index only data that can have a unique import filter or text import schema specifically assigned to the data. Use the assignment table called Assign File Types to specify which import filter or which text import schema is to be used for specific folders and files on the hard disk. You can learn more about this feature by opening the Help dialog box.

Note: You will know whether you have to make an assignment for a file when you manually import it. Whenever the Choose Import Format dialog box appears and more than one file format or text import schema is listed, this means that the data format is ambiguous and you need to solve the problem.

Folders to Index

Here you specify the hard drive folders in which the program will search for data files. Use the Subfolders column to specify whether to search through specific subfolders as well.

In the Excluded Files and Folders field, specify the file types (e.g. *.exe; *.sys) or paths of folders and individual files that should be ignored when indexing.

Setting Up File Indexing in Client/Server Mode

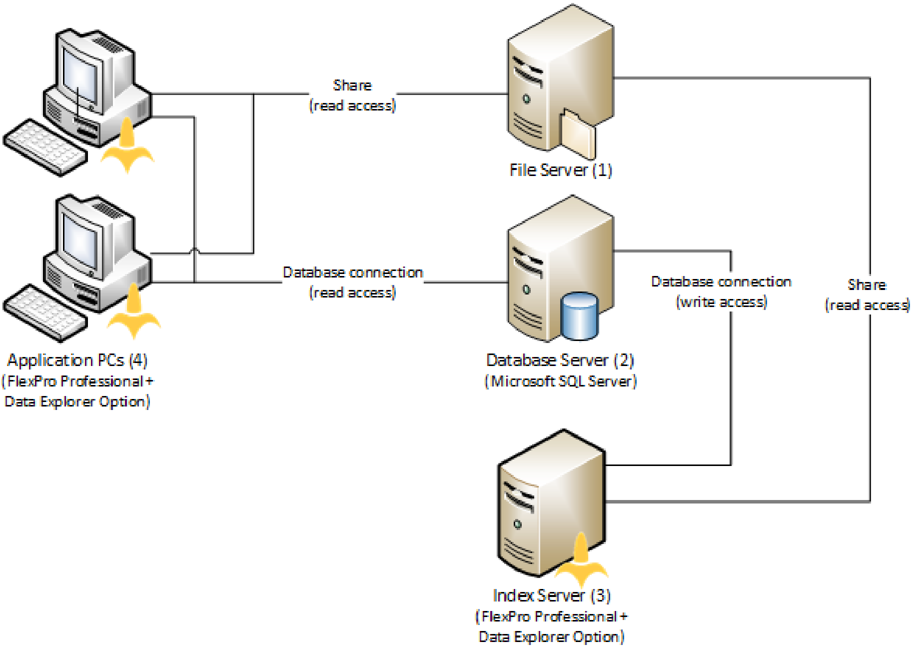
File indexing in client/server mode offers the advantage of allowing more than one FlexPro user to access shared data without each user having to create and update their own index database.

Components

The following components are used in this operating mode:

1. One or more file servers on which the data to be indexed is stored.
2. A database server with Microsoft SQL Server on which the index database is stored.
3. An index server on which an instance of FlexPro Professional is installed with the Data Explorer option, which indexes the data.
4. Several clients on which FlexPro Professional is installed with the Data Explorer option.

The file, database and index server functions can be set up on a single Windows computer or on separate computers.



Security

FlexPro does not use its own layer of safety for file indexing, but instead uses Microsoft Windows and Microsoft SQL Server user account management data.

- For the data to be indexed, Folder Sharing must be set up, which gives FlexPro users and the index server the ability to access and read the data.

- The user account under which the FlexPro index server instance runs requires write access to the SQL server and read access to the data's folder sharing.
- To set up an index database, additional temporary administrator rights are required on the SQL server. For this purpose, you can enter relevant login information during setup.
- The FlexPro user accounts need read access to the SQL server and read access to the data's folder sharing.

Setting Up the File Server

1. Log onto the server using the account that allows you to set up file sharing.
2. Open Windows Explorer.
3. Right-click with your mouse on the folder that contains the data and select Share with specific people.
4. Add the accounts of all the users who are allowed to access the data and give them read access.
5. Add the account under which the FlexPro instance runs on the index server and give it read access as well.

Note: Microsoft Windows manages access rights separately for sharing and for folders and files in the file system. If configuration differs, the more restrictive configuration applies. If you are still unable to access the data despite having sharing enabled, this is usually because the access rights at the file system level are not sufficient.

Setting Up the Database Server

1. Install Microsoft SQL Server (not included with FlexPro).
2. Set up an administrator account.
3. Give all users read access who are allowed to access the data (see "Setting Up the Index Server").
4. Give all users write access who are allowed to access the data (see "Setting Up the Index Server").

Setting Up the Index Server

1. Log in using the account that you want to use to run the FlexPro Indexer.

2. Install FlexPro Professional and the Data Explorer option. To do this, you may need the computer’s administrator account login information.
3. Launch FlexPro.
4. Click on File > Options.

Note You can also open the Options dialog box using the Options button on the Data Explorer window toolbar.

1. In the Options dialog box, click on the File Indexing tab.
2. Click on Set Up Database...
3. In the first step of the wizard, choose Microsoft SQL Server for the database connection.
4. Enter the required connection settings in the second step of the wizard. Click on Test Connection to test the selected settings. This only checks whether the index server can be accessed using the connection settings and whether the ODBC driver version is compatible with the Microsoft SQL Server version.
5. When you go to the third step of the wizard, it checks whether an index database with the name entered in the second wizard step already exists on the SQL server. If not, you can add the database. Otherwise, you can delete and add a new database. You need to have SQL server administrator rights to do this (member of the sysadmin role). When the index database is created, the user role db_index_reader is set up, which must be assigned to the user for reading.
6. In the fourth step, the wizard checks whether write access is available to the index database using the selected connection settings. The user name and password can be changed if necessary.

Summary of the roles that have to be assigned to the relevant user accounts in Microsoft SQL Management Studio:

Task	Server/user role membership
Create index database	sysadmin
Reset index database	db_owner
Write access (server)	db_owner
Read access (client)	db_index_reader

Setting Up Calculations

1. Click Calculations Setup on the Data Explorer toolbar.
2. In the Calculations list, click on Add Calculation.
3. Call up Help in the Edit Calculation dialog box to obtain more information.

Note You cannot use the FPScript Debugger to check the code at this point, since no data is present yet. To make sure that the calculation works properly for indexing, it is recommended that you import a typical data set to which the calculation can be applied and enter the code for the calculation on the Calculations tab in the Properties dialog box of this data set. In the Edit Calculation that now appears, you can use the FPScript Debugger to test the code.

Updating the View

You can update the view in Data Explorer by clicking on the Update button in the toolbar of the Data Explorer window.

Note: An update may be required if the indexed data has changed.

Searching in Data Explorer

In Data Explorer you can set up search filters for indexed data and perform a search using a number of search criteria.

Note: To do this, file indexing must be enabled.

Setting Up Search Filters

1. Select the option Find from the Search dropdown list box in the toolbar of the Data Explorer window, or click on the Search button in the toolbar of the Data Explorer window.
2. Specify where to save the search filter. For a one-time search that will not be saved, you can select Data Explorer and define a standard filter. Otherwise, you have the option of saving the search filter in a personal template database or the open project database.
3. Add a new search filter and give it a name.

4. Under Search Criteria specify the properties that you want to find.

Selecting Search Filters

Pre-defined search filters appear in the Search dropdown list box and can be selected there.

Removing Files or Folders from the Index

Select one or more data sets

1. From the Data Explorer list view.
2. Right-click with your mouse in the selection to open the context menu.
3. Select Remove All File(s) of Selected Channels from Index.

Select a folder or file

1. In the Data Explorer tree view.
2. Right-click with your mouse in the selection to open the context menu.
3. Select Remove File(s) / Folder from Index.

Note The files or folders removed from the index are copied to the list of excluded files and folders. This ensures that they are not indexed again.

3.4 FlexPro Objects

All data, analyses and presentations are stored by FlexPro as objects in the project database. These objects are linked to one another and thus form a dynamic network. For instance, a document is linked to the diagram within it, the diagram is linked to an analysis object and the analysis object is linked to the data set containing the raw data.

Updating

After editing one or more objects, they will have to be updated. In certain instances, updating can require a substantial number of calculations, which is why there are various options available to you to manage the update process. You can manually update individually selected objects or all open objects. FlexPro then analyzes the object network and updates only those elements that are not current. You can, however, force a complete update. If you enable automatic update, then FlexPro updates the objects automatically when they are opened or when changes are made to them. FlexPro can update objects in the background. This allows you to continue working during the update. Objects that are currently being updated, however, are blocked from use.

If you work with data links that refer to changing data, you can enable periodic updating. FlexPro will then refresh the display automatically at the specified rate.

We recommend using automatic updating if you are working with a smaller amount of data. FlexPro will then update objects automatically every time they are edited.

Read-Only and Locked Attributes

You can set every object in the FlexPro project database to read-only. You will no longer be able to edit the attribute of a read-only object, but you can continue updating the object, e.g. after you import new raw data. If you use cursors in a read-only diagram, all functions that would normally change the diagram, such as zooming, setting and removing markers, or translating curves, are blocked. Even folders can be set to read-only. In a read-only folder you cannot add or remove objects. You can, however, continue to edit the objects themselves as long as they are not also read-only.

You can lock the object to provide further protection for an object. A locked object can no longer be updated. For instance, a locked diagram behaves like a static image, and a locked formula, analysis object or data link is like a data set. The data of locked formulas, analysis objects and data links are stored in the project database and remain unchanged after the database is reopened. You cannot use the cursors in a locked diagram or document. Worksheets cannot be locked.

Indexing Objects

If indexing is switched on, all objects in the project database are indexed. However, you can exclude individual objects or entire folders from indexing. These will then no longer be found when searching via the project database index or via the data query. This makes sense, for example, for the data query objects themselves.

Hidden Objects

You can hide individual objects or entire folders in the project database. These are then no longer displayed in the object list, but can otherwise be used normally.

Properties Dialog Box and Window

A Properties dialog box is assigned to each object. Here, you can adjust all of the object-specific settings. For example, you can change the comments for a data set or the options of an analysis object.

There are often more settings options available in the Properties dialog box than in the editor that you normally use to edit the object. Thus, for example, you can also set the order in which the curves are to be drawn in the Properties dialog box of a diagram. Often there is no access in the Editor to the item you would like to edit, e.g. if you would like to delete a curve from a diagram containing a data set that does not provide any data. The curve is then no longer visible in the Editor and can therefore not be clicked. You also have to use the Properties dialog box in such cases.

FlexPro also has a Properties window in which all properties of the selected objects are displayed as a list. The advantage of the Properties window over the Properties dialog box is that you can edit the properties of several objects at once. For instance, to enter the physical unit "V" in ten data sets simultaneously, select these in the object list and enter "V" in the Y Unit field.

Object Parameters

In every FlexPro object, including folders, FlexPro manages a list of parameters that you can edit freely. Each parameter can take a scalar value in one of the data types supported by FlexPro. You can also assign a physical unit to the value. You can later query the parameters using a formula and calculate or output them. Some import filters store additional information as parameters.

Naming Conventions for Objects

The following points must be taken into consideration when assigning names to FlexPro objects:

- The maximum length of an object name is 200 characters.
- The following characters must not be used: \ / : . * ? ! " ' < > |
- No differentiation is made between upper and lower case when objects are saved, i.e. the names "OBJECT" and "Object" are considered identical.
- Objects of different types can have identical names, where data sets and formulas are considered one object type in this case.

Note A name should start with an alpha character and only alpha characters, numerical digits and the '_' character should be used for the rest of the name. This corresponds to the syntax rule for identifiers in FPScript.

File Name Extension

In a project database folder, all objects of the same type have to have a unique name. Thus, it is not possible, for instance, to place two documents with the name "Document" in the same folder. However, you can use the same name for objects of different types, e.g. a diagram and a column table both with the name "Evaluation". This means that the name of an object is not enough to clearly identify an object.

Therefore, a file name extension is assigned to each object class. This extension is appended to the name to clearly identify it. The following table shows the file name extensions used:

Object Class	File Name Extension
Folder, Data Set, Formula, Analysis Object, Data Link Object, etc.	(no name extension)
Document	.doc
Worksheet	.wks
2D Diagram	.2d
3D Diagram	.3d
Media	.med
Column table / Row table	.tab
Cell Table	.ttb
Text	.txt
Control panel	.cpn
Folder	.fld (optional)

As you can see from the table, you do not have to specify a file name extension for data objects, i.e. for data sets, formulas and objects derived from them. The same applies to folders that when used as data objects render their content as a list. This is practical because these are the most frequently accessed objects in formulas.

To access the "Data" data set comments within a formula, write:

```
Data.Comments
```

If you would like to access the comments from a document called "Evaluation", you have to use the following expression:

```
Evaluation.doc.Comments
```

Working with Objects

Creating Objects

1. In the Folders window, select the folder where you want to insert the object to display this folder in the object list.
2. Now select all of the data to be displayed in the object or to be calculated by the object. If you want to select data, you can highlight entire data sets within the object list or within the name row of the open folder's data view, or you can select a cell range in a data set window or in the data view.
3. Select the type of the newly created object from the Insert ribbon tab. Alternatively, you can right-click with your mouse to open the object list context menu and select the object from the Insert submenu.
4. An object of the selected type is added with a unique name to the displayed folder. If necessary, a wizard will appear first to assist you in setting the object properties.

Note: You can also embed diagrams, tables, media and text objects to documents 478 or worksheets 495. These will not appear in the object list.

Updating Objects

To update one or more objects:

1. Select the objects in the object list or an object in the Object Hierarchy window.
2. Click Home[Update] > Update.

To update all objects in a folder:

1. Select the folder in the Folders window.
2. Click Home[Update] > Update.

Note: Objects contained in subfolders are only updated if they are used by objects in the selected folder.

To update an object that is open:

1. If necessary, click on the tab of the window where the object is displayed to activate it.
2. Click Home[Update] > Update.

To update all objects that are open:

- Click Home[Update] > Update All.

If you use the Update menu instead, select the Force Update command, the selected objects will be fully updated even if FlexPro was unable to detect any relevant changes.

To update objects automatically every time they are edited:

- Select the option Home[Update] > Automatic.

To update the active object or all opened objects periodically:

1. Click Home[Update] > Change Update Interval.
2. Specify the desired update interval in the format HH:MM:SS.
3. If the command Update All is to be performed periodically, select the option Update all. Otherwise, FlexPro uses the Update command.
4. Close the dialog box with OK.
5. Select the option Home[Update] > Periodic. The periodic update will take place until you uncheck the option again.

Updating Objects in the Background

1. Click on File > Options.
2. Click on the System Settings tab.
3. Select the option Update objects in the background.

Note: The option also affects updating of Preview.

Changing the Update Interval for the Periodic Update

1. Click Home[Update] > Change Update Interval.
2. Specify the desired update interval in the format HH:MM:SS.
3. Choose whether the Update or Update All command is to be used.

Changing Lock and Read-Only Object Attributes

To change the lock or read-only attribute of one or more objects:

1. Select the objects in the object list.
2. Right-click with your mouse on the selection to check or uncheck the Locked or Read-only option in the context menu.

To change the lock or read-only attribute of an object and all linked objects:

- Right-click with your mouse on the object in the Object Hierarchy window and check or uncheck the Locked or Read-only option in the context menu.

To change the lock or read-only attribute of all objects in a folder:

1. Right-click with your mouse on the folder in the Folders window and check or uncheck the Locked or Read-only option in the context menu.
2. In the dialog box that now appears, specify whether the attribute should be changed for the folder itself or for all the objects and subfolders within it.

Excluding or Re-Including Objects for Indexing

To exclude or include one or more objects or folders from indexing:

- Right-click with your mouse on the objects or folders in the object list and in the context menu select or deselect the option Do not index.

To exclude or include an object and all linked objects from indexing:

- Right-click with your mouse on the object in the Object Hierarchy window and in the context menu select or deselect the option Do not index.

Hiding Objects and Showing Hidden Objects

To display hidden objects in the object list:

- Select the option View[Object List] > Show Hidden Objects.

To hide or show objects again:

1. Select the objects in the object list.
2. Right-click with your mouse on the selection and in the context menu select or deselect the option Hidden.

To hide or show an object and all linked objects:

- Right-click with your mouse on the object in the Object Hierarchy window and in the context menu select or deselect the option Hidden.

To hide or show all objects in a folder:

1. In the Folders window, right-click with your mouse on the folder and in the context menu select or deselect the option Hidden.
2. In the dialog box that now appears, specify whether the attribute should be changed for the folder itself or for all the objects and subfolders within it.

Categorizing Objects

1. In the object list, select the objects you would like to assign to a new or existing category.
2. Choose the Categorize command from the context menu or on the Home ribbon tab in the Object group.
3. In the dialog box displayed, enter the desired category or select one of the existing categories from the list box.

Opening an Object

To open a folder in a separate window:

- Right-click with your mouse on the name of the object in the object list or in the object hierarchy and select Open from the context menu.

To open the formula of an analysis object or data link object:

- Right-click with your mouse on the name of the object in the object list or in the object hierarchy and select Open from the context menu.

To open any other object:

- Double-click on the name of the object in the object list or object hierarchy.

Undoing and Restoring Edits Made to an Object

a) In the object list:

1. Right-click with your mouse on the object.
2. In the object context menu that appears, select Undo or Restore.

b) In the open window of the object:

1. Select Undo or Restore from the FlexPro Quick Access Toolbar.

Undoing and Restoring Edits Made to Multiple Objects

- If you have edited several objects using the Properties window or the Find & Replace function, you can undo or redo this operation as a whole by clicking in the object list and then selecting Undo or Redo in the FlexPro Quick Launch bar.

Displaying an Object's Properties Dialog Box

- Right-click with your mouse on the object in the object list, object hierarchy or Folders window and choose Properties from the context menu.

Renaming an Object

1. Select the object to be renamed from the object list, object hierarchy or data view of an open folder.
2. Press the F2 key or click on the object that is already highlighted. An input field appears, in which the object name is selected.
3. Enter a new name or change the existing name.

Moving Objects

To move objects via the Clipboard:

1. Select the objects in the object list that you want to move.
2. Click Home[Clipboard] > Cut or use the keyboard shortcut CTRL+X.
3. Open the folder where you would like to move the object. You can also open the target folder in another project database.
4. Click Home[Clipboard] > Paste or use the keyboard shortcut CTRL+V.

To move objects with the mouse:

1. Select the objects in the object list that you want to move.
2. Use your mouse to click on one of the objects selected. While holding down the left mouse button, move the mouse cursor onto the target folder and once there, release the mouse button.

Note: If you move an object with a name that has already been used in the target folder, then FlexPro's behavior depends on which settings you chose on the Operation tab of the Options dialog box. If you enabled Automatic renaming of objects with copying or moving actions, the object names affected are made unique by appending a number to them in the target folder. Otherwise, a dialog box appears, allowing you to decide whether you would like to cancel the copying process, rename the object or replace the object in the target folder.

Copying Objects

To copy objects via the Clipboard:

1. In the object list, select the objects you would like to copy.
2. Click Home[Clipboard] > Copy or use the keyboard shortcut CTRL+C.
3. Open the folder where you would like to save the copy. You can also open the target folder in another project database.
4. Click Home[Clipboard] > Paste or use the keyboard shortcut CTRL+V.

To copy objects using the mouse:

1. In the object list, select the objects you would like to copy.
2. Hold down the CTRL key and use your mouse to click on one of the selected objects. While holding down the left mouse button, move the mouse cursor onto the target folder and once there, release the mouse button.

Note: If you copy an object with a name that is already in use in the target folder, the behavior of FlexPro will depend on which setting you selected on the Operation tab in the Options dialog box. If you enabled Automatic renaming of objects with copying or moving actions, the object names affected are made unique by appending a number to them in the target folder. Otherwise, a dialog box appears, allowing you to decide whether you would like to cancel the copying process, rename the object or replace the object in the target folder.

Copying Objects and Linked Objects

To copy FlexPro objects with linked objects, you first have to select them in the object list or object hierarchy. To do this, proceed as follows:

In the object list:

1. In the object list, select the objects you would like to copy.
2. Click with your right mouse button on one of the selected objects and choose Select Linked Objects from the context menu.

In the object hierarchy:

- In the object hierarchy, select the object you would like to copy.

Note: You can select only one object at a time in the object hierarchy. This selection, however, does not represent only the object itself, but also represents all objects under it in the hierarchy.

To copy the selected objects via the Clipboard:

1. Click Home[Clipboard] > Copy or press the keyboard shortcut CTRL+C.
2. Open the folder where you want to place the copy by selecting it in the Folders window. You can also open the target folder in another project database.
3. Click Home[Clipboard] > Paste or use the keyboard shortcut CTRL+V.

To copy the selected objects with the mouse:

4. Hold down the CTRL key and use your mouse to click on one of the selected objects. While holding down the left mouse button, move the mouse cursor onto the target folder and once there, release the mouse button.

If the objects to be copied need to be renamed in the target location because that location already contains objects with the same name, then FlexPro corrects all links so that the copied object network continues to work properly in the target location and no links are broken.

Deleting Objects

1. From the object list, select the objects you want to delete.
2. Click Home[Selected Objects] > Delete or press the DEL key.

Note: You can also delete a single object or folder in the object hierarchy or in the Folders window.

Editing Object Parameters

In the Properties window:

In the Properties window you can edit the parameters of one or more objects at the same time.

1. Select one or more objects with parameters that you would like to change.
2. Scroll down the Properties window until you see the Parameters group.
3. Enter the new values for the parameters that you want to change.

In the Properties dialog box:

In the Properties dialog box, you can edit parameters as well as add, remove or resort them.

1. Right-click with your mouse on the object in the object list and select Properties.
2. Switch to the tab called Parameters.

To add a parameter:

3. Click Add Parameter or press ENTER.
4. In the Name field, enter a unique name for the parameter.
5. Choose the desired data type in the Data Type field.
6. In the Value field, enter the value of the parameter and append the unit symbol, if applicable.

To change a parameter:

3. Select the parameter from the Parameters list.
4. Change the name, data type, value or unit and exit the input field with the TAB key for the change to take effect.

To delete a parameter:

3. Select the parameter from the Parameters list.
4. Click Remove Parameter or press the DEL key.

To sort the parameters in the list:

3. Select a parameter from the Parameters list.
4. Click Move Parameter Up to move it up one position.
5. Click Move Parameter Down to move it down one position.

Note: You can use FPScript formulas to enter data. To do this, the text to be entered must start with '=' for instance, the value 5 is entered for the parameter when you input '=2 + 3'.

Printing Objects

To print several FlexPro objects:

1. From the object list, select the objects you want to print.
2. Right-click with your mouse in the context menu and select Print.
3. In the Print dialog box that appears, select the printer and the page layout.
4. Click on OK to start printing.

Note: To cancel printing, press the ESC key.

To print an object displayed in a window:

1. Click on the object in the window that you want to print.
2. Select File > Print.
3. In the Print dialog box that appears, select the printer, the print range and the number of copies.
4. Click on OK to start printing.

3.5 Data Object

Data object is the umbrella term for data sets 132, formulas 235, analysis objects 293 and data link objects 160, or for any objects that contain data or return data as a calculated result. Each folder in the project database is also a data object, as it represents the data sets stored in it as a list 124. All data objects except data sets are based on FPScript 246 code. These data objects, however, can be used like data sets and can be displayed, for instance, as a diagram or table. The result can have one of the data structures and data types supported by FPScript. You can convert all FPScript-based data objects into data sets at any time. The formula containing the FPScript is calculated one last time and then is replaced by its result.

Data Type

FlexPro is capable of managing and processing the data so that the memory required is optimally adapted to the resolution of the data. This is particularly useful for large data sets, in view of the fact that the memory required can vary by up to a factor of four, depending on the data type.

FlexPro achieves this by using different data types, i. e. storage formats for numbers and text. The system has over four numerical basic data types that can be used for real and complex numbers as well as numeric data types for calendar time and time span. In addition to this, there is a data type for strings, for Boolean values and for the **Empty** data type, which represents an empty result.

The following table provides an overview of the data types.

Data Type	Explanation	Notation in FPScript
Empty	Uninitialized variable, empty result or placeholder for a function argument.	Empty <Identifier>
Boolean value	Can have the value TRUE or FALSE.	TRUE, FALSE
16-bit integer	Can contain integral numbers in the range from -32768 to +32767.	1234S, -1245S
32-bit integer	Can contain integral numbers in the range from -2147483648 to +2147483647.	1234N, -1234N, 1000000N
64-bit integer	Can contain integral numbers in the range from -9223372036854775808 to +9223372036854775807.	1234L, -1234L, 2147483648N
32-bit floating point value	Can contain floating point values in the range from -3.4e+38 to +3.4e+38.	1.23S, 1.45e-3S, ?S
64-bit floating point value	Can contain floating point values in the range from -1.7e+308 to +1.7e+308.	1.23, 1.45e-3, 1.23L, ?
Complex 16-bit integer	Can contain complex integers in the range from -32768 to +32767.	(1234S, -1245S)
Complex 32-bit integer	Can contain complex integers in the range from -2147483648 to +2147483647.	(1234N, -1234N)
Complex 64-bit integer	Can contain complex integers in the range from -9223372036854775808 to +9223372036854775807.	(1234L, -1234L)
Complex 32-bit floating point value	Can contain complex floating point values in the range from -3.4e+38 to +3.4e+38.	(1.23S, 1.45e-3S)
Complex 64-bit floating point value	Can contain complex floating point values in the range from -1.7e+308 to +1.7e+308.	(1.23, 1.45e-3)

Data Type	Explanation	Notation in FPScript
Calendar Time	Can contain data and time values. Calendar time values are stored as 64-bit floating point values representing the number of seconds that have passed since 1/1/1970 Coordinated Universal Time (UTC). Calendar time values before this date are not supported, i.e. the floating point value based on this calendar time value cannot be negative. FlexPro interprets calendar time values as quantities using seconds as the implicit unit. The unit symbol, however, is not output. Calendar time values are converted to the time zone configured under File > Options > Language and Region. An entered calendar time value is also interpreted in this time zone unless you explicitly specify the time zone when entering it. This is done by specifying the time offset to UTC in hours and minutes.	Examples in local time: ‘12/23/2024 12:23:30.1234' ‘12/23/2024 12:23:30.1234' '?/?/?' (Elements of time can be omitted starting from the right and are then taken as 0) Examples in local time: '12/23/2024 12:23:30 UTC+2' '12/23/2024 12:23:30 UTC+3:30' '12/23/2024 12:23:30 UTC+3.5'
Time Span	Can contain time spans written in the formats DD:HH:MM:SS or HH:MM:SS. The values are stored as 64-bit floating point values with the unit as seconds. FlexPro interprets time spans as quantities using seconds as the implicit unit. The unit symbol, however, is not output.	'12:23:30.1234' '11:12:23:30.1234' -'12:23:30' (negative time span) '?:?:?'
String	Can contain strings of up to 2 GB in length.	"This is a string"
Object Reference	A reference to any automation object or FlexPro object.	Set Object = Diagram.2d

As can be seen from the examples above, FlexPro uses the decimal point or the appended 'N' or 'n' to distinguish between integers and floating point values. The resolution can be specified using the appended 'S' or 'L', or 's' or 'l'. 'S' stands for "Short", i.e. a 16-bit integer or 32-bit floating point value, and 'L' stands for "Long", i.e. a 64-bit integer or 64-bit floating point value. If you omit the suffix, the higher resolution is automatically selected for floating point values, and for integers, a 32-bit or 64-bit resolution is used, depending on which format can store the specified constant. The 'L' can therefore usually be omitted.

Note How decimal numbers to which you have not added a suffix or decimal point are interpreted depends on the setting Interpret decimal numbers without a decimal point as floating point values on the General tab of the project database Properties dialog box. This option is enabled by default, which means that the numbers are interpreted as 64-bit floating point values. This setting prevents accidental rounding errors that may occur when using integer arithmetic. The integral division 1 / 2 results in 0 and not 0.5, for instance.

If you are calculating data with a different format, FlexPro adapts the data types automatically. Automatic adaptation always happens in such a way that no information is lost. If, for instance, an integer and a floating point value are to be added together, the integer is converted to a floating point value and not the other way around, because otherwise decimal places might be lost. For constant data 271, which you use in formulas, FlexPro recognizes the data type from the notation. You have the option of converting the data types explicitly by using an operator for type conversion 276:

`FloatingPoint64 DataSeries` converts all values in a data series into 64-bit floating point values.

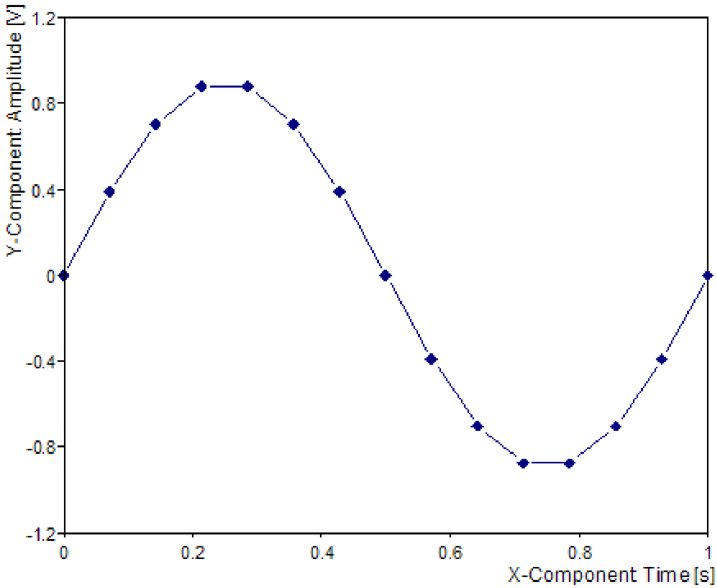
Object references can only be used on a limited basis. You can assign an object reference to a local variable but neither pass it as an argument to an FPScript function nor return it as the result of a formula. If an object reference is used as a function argument, as a return value of a formula, or in a mathematical calculation, then it is converted into its value. If the object, to which the reference refers, is a data object, e.g. a formula, then its return value is used; for all other objects, the value is Empty.

Component

Data sets can have a composite data structure, i.e. a data structure that consists of up to three components. These components are called the X, Y and Z components. The description corresponds to the names of the axes of 2D and 3D diagrams.

A simple time signal consists, for example, of a Y component, which contains a data series with the voltages, and an X component, which contains a data series of equal length with the associated times. One value each from the Y and X component together form one point of a curve in the two-dimensional plane of a 2D diagram.

The following illustration clarifies these relationships:



Time [s]	Amplitude [V]
0	0.0000
0.0714	0.3905
0.143	0.7036
0.214	0.8774
0.286	0.8774
0.357	0.7036
0.429	0.3905
0.5	0.0000
0.571	-0.3905
0.643	-0.7036
0.714	-0.8774
0.786	-0.8774
0.857	-0.7036
0.929	-0.3905
1	-0.0000

Within formulas, you can separately access the individual components of the data set by specifying the name of the component as a name extension. `Signal.Y` extracts, for instance, the Y component from the data set `Signal`.

Data Structure

Data sets can have different data structures. The data structure specifies how the data is organized in the data set. A differentiation is made between simple data structures and composite data structures. Composite data structures are built up from the simple data structures. The following table provides an overview of simple data structures:

Data Structure	Contents
Scalar Value	A scalar value.

Data Structure	Contents
Data Series	A one-dimensional array of values.
Data Matrix	A two-dimensional array of values. The columns can be regarded as data series with the same number of values.

All data types are permitted for the simple data structures.

The following composite data structures are based on the structures above:

Data Structure	Contents	Application Example
Signal	A Y component and an X component, each with a data set of the same length.	A time signal. The Y component contains the voltage and the X component the associated time.
Signal Series	A Y component with a data matrix and an X component with a data series. The length of the data series must match the number of rows in the data matrix.	A spectral series. The Y component contains several frequency spectra and the X component of the frequency values applicable for all spectra.
Signal series with a two-dimensional X component	A Y component and an X component with a data matrix of the same size.	A series of time signals. The Y component contains the voltages of the individual signals and the X component, contains the associated calendar times.
Signal series with a Z component	As with a signal series, however, with an additional Z component with a number of values that must correspond to the number of columns in the Y data matrix.	A spectral series, where the time points for which the spectra have been determined are entered into the Z component.
Signal series with two-dimensional X component and Z component	As with a signal series with a two-dimensional X component, however, with a Z component with a number of values that must correspond to the number of columns in the Y data matrix.	A series of time signals. The Y component contains the voltages of the individual signals and the X component contains the associated times in seconds. The Z component contains the calendar times at which the signals were measured.

Data Structure	Contents	Application Example
Space Curve	A Y component, an X component and a Z component, each with a data series of the same length.	A trajectory in three-dimensional space. A measurement that was taken at arbitrary locations on a surface.
List	A list in which each element can have any data structure and can especially be a list itself. The elements in lists can also contain object references.	A list containing several data sets, such as the result of a data query. A complex data structure, such as the result of a regression, which, in addition to a data series with found parameters, also contains their confidence intervals and the goodness-of-fit measure of the regression.

For the X and Z components of the composite data structures, all real data types as well as the calendar time and time span types are allowed. For the Y component, additional complex data types and the Boolean value type are permitted. The composite data structures from FlexPro especially simplify the analysis in FlexPro, since the X, Y and, if applicable, Z components in a single formula can be calculated. A formula that calculates a Fourier transform, for instance, can re-calculate the amplitudes of the time signal in spectral amplitudes and the time values into frequency values at the same time.

Use the List data structure to create complex data structures, for instance, if the data structures supported by FlexPro are not suitable for the result. You can have the results returned as a list even when a calculation procedure returns several results that cannot be calculated independently of each other as separate formulas. Lists are the basis for multi-channel analyses using FlexPro. Many analysis objects as well as functions and operators in FPScript support lists.

Lists are special because they are only allowed as a return value for formulas, but not as a data structure for a data set. Instead, a folder in the project database represents the data sets contained in it as a list, whereby component data sets assigned to each other in the list are combined into composite data structures. When converting a formula that returns a list into a data set, a folder is therefore created in which the elements of the list are inserted as data sets.

Data Unit

Each component of a data object can have a physical unit and is thus converted to a quantity. The Unit Manager is an integral part of FPScript. You can make calculations with quantities directly in FPScript. FPScript processes the unit as an integral part of the data, is able to convert units, recognizes incorrect calculations and determines the result unit of a calculation from the units of the arguments. The following FPScript code generates, for instance, a signal with a sine wave with an amplitude of 2 V, a frequency of 0.1 kHz and a sampling interval of 1 ms:

```
// Increasing time values in the interval [0 s, 10 s]
// and sampling interval 1 ms
Dim t = Series(0 s, 10 s, 1 ms)
Signal(2 V * Sin(2 Pi * 0.1 kHz * t), t)
```

Alternatively, the unit can also be assigned subsequently via the Header Information 126 of the data object. Please see below for further details.

Assigning Data Objects

FlexPro normally saves all components of a composite data structure together in a single data set; for instance, the amplitudes and time values of a time signal. This also applies if there are several data sets with the same X component, such as when you import data from a multi-channel measurement. The time values for the individual channels are usually linearly increasing, so FlexPro only has to store the initial value, increment and number of values as the parameters instead of columns of numbers. There are, however, cases where the X values must be stored explicitly, e.g., because the sampling rate varied during the measuring process. In this case, storing the same X values in each data set would be inefficient. Therefore, FlexPro supports another option to manage composite data structures. To do this, the Y values and the X values are stored in separate data sets as data series, and these data sets are then assigned to each other.

This assignment is stored in the header information of the data set and can be changed on the Component & Assignments tab of the Properties dialog box.

The data set that contains the data series with the X values is marked as the X component there, and the data sets containing the Y values for individual channels of a measurement are marked as Y components respectively. In addition, reference to the X data set is made in the header information of the Y data sets.

By making assignments, FlexPro can always find the appropriate X data set, e.g., when creating a diagram. In the diagram's curve, FlexPro then activates the option

Separate data sets for X and Y components and additionally enters the X data set assigned in the Y data set into the curve.

Not only can this assignment be carried out for data sets, but also for all data objects, such as formulas. For formulas, however, the assignment is usually not necessary, since the X component can simply be included in the result with the Signal function.

Data Order

The data order is a property of a data series, which specifies how the values in the data series are sorted. A distinction can be made between the following data orders:

Description	Purpose
None	The values in the data set are not in any particular order.
Constant	The data set is a scalar value, or all values in the data set are identical.
Increasing	Each value in the data set is greater than or equal to its predecessor.
Linearly increasing	The distance between neighboring values is constant and greater than zero.
Decreasing	Each value in the data set is less than or equal to its predecessor.
Linearly decreasing	The distance between neighboring values is constant and less than zero.

A signal for which the X component is linearly increasing is described as sampled equidistantly. Some algorithms for data analysis, e.g., FFT, require this.

FlexPro displays the data order of the individual data object components in a popup window when you use the mouse to point to the object in the object list.

Header Information

Header information, also known as metadata or data header, are attributes that you can assign to a data object.

For this purpose, you can enter the following in the Properties dialog box of a data object (the first four attributes apply to each object, not only for data objects):

- Comments

These are comments concerning the content of the object. The comments appear together with the name in the object list. You can enter up to three comments for data objects with a composite data structure. One for each component.

- Hyperlink

  A reference to an external file, website, e-mail address or another object in the project database. The Follow Hyperlink command opens the object to which it referred.

- Date

  The creation date and time of the object are specified here.

- Parameters

  A freely editable list of parameters 107 in which additional information can be stored.

The following attributes are specific to data objects:

- Calculations

  A list of calculations 97 that provide quantities related to the data set. Usually you set up the calculations for data indexing. When importing a data set, Data Explorer then copies these into the data set header information.

- Unit

  The physical unit for the individual components of the data object. The interpretation of the unit in the header information depends on whether the data object is a data set or a formula, or an analysis object:

  - The unit is assigned to the data of a data set without transformation, and in the future, these will be returned as present in this unit. In data sets, FlexPro generally always keeps the data unit in sync with the unit in the header information.
  - The data of a formula or analysis object is, however, transformed to this unit before each access. If FPScript code provides a result without a unit, the specified unit will be assigned without transformation. However, if the code provides a result with a unit, this will be transformed into the specified unit. If the unit in the header information is incompatible with the unit that the result of the FPScript formula provides, then you will see an error message from FlexPro.

Note: To transform the data of a data set into a different unit, use the command Change Units or enter the new unit symbol in the Data Unit field of the Properties window.

- Quantity

  The name of the physical quantity for the individual components of the data object.

- Upper and lower data range limits

  With measured data, these fields specify the data range used in the physical unit of the data, such as -5 V to +5 V. FlexPro can use them for autoscaling axes. The data range limits are optional. If you do not specify them, the minimum and maximum in the data set will be used instead.

- Timestamp

  FlexPro uses these fields, if specified, to convert time data from relative time to absolute time and vice versa. The timestamp is a calendar time value, which is assigned the time t = 0 in the time data. When converting to absolute time, this is added to all data, and when converting to relative time, it is subtracted.

- Author

  The author of the object. This is particularly useful for data sets entered manually and for formulas.

- Origin

  A description of the origin of the data object, e.g. the name of the location of a measurement.

- Assigned data sets

  Refers to assigned data objects that form a composite data structure together with the data of the object (see above).

- Format, X format, Z format

  Specifies how the data of the data object should be formatted when output.

- Color

  A color can be assigned to each data set while the data is imported. This color then has priority over the color scheme of the diagram when displayed as a curve. The colors are read from the file to be imported as long as they are stored in the file. The result is that the curves in FlexPro are displayed in the same colors that are displayed on the measuring device.

Data Export

FlexPro offers you the option of exporting data sets, formulas and related objects. You can save the data in the FlexPro (.fpf), Text (.txt), CSV (.csv) and (.wav) file formats as well as several proprietary file formats. CSV stands for Comma Separated Values. The values are separated in this format with the list separator specified in the Windows Control Panel under Region.

When saving in CSV or text format, you have the option of exporting several objects below one other or side-by-side to a single file. For these formats, you can additionally select different file formats and specify whether the result or formula code is to be output for the formulas.

The FlexPro format is similar to that variant of the text format, where the data of several data sets are stored side-by-side in a table and with all of the header information. Here, however, an additional row with a data format identifier is then output. Regardless of the system settings, floating point values are always output with a point as the decimal symbol. Calendar times are also always output in the same format, regardless of the system settings.

Primarily signals and signal series with two signals are suitable for export as an audio file. Signals are treated as a mono signal, and signal series with two signals are treated as a stereo signal. Signal series with more than two channels are exported as multi-channel wave files accordingly. The data's X component must correspond to one of the following sampling rates supported by the audio file format: 11025 Hz, 22050 Hz or 44100 Hz.

You can export the data from formulas, data sets and related objects as OLE objects and thus insert links to this data into OLE client applications. FlexPro provides the data in CSV and text formats.

Working with Data Objects

Playing Data Objects as an Audio File

1. In the object list, click on a data set, a formula, an analysis object or a data link object.
2. Click Home[Object] > Play.

Note: Use the media object to play audio signals synchronized with measurement data.

Editing a Data Object's Header Information

1. Use your right mouse button to click in the object list on the formula or in the Formula Editor or data set window to open the context menu.
2. Select Properties to open the Properties Dialog Box 107.
3. You can make changes to all settings on the General tab.

Note Analysis objects 293 can often automatically generate the units and comments for your result. You can activate this mechanism by leaving the corresponding text boxes empty.

Changing Data Object Units

Converting Units of Data Sets and Formulas

Only components that already have a unit are transformed to the new unit. Otherwise, the unit is only assigned without changing the data.

Via the Properties window

1. Select the data sets with a unit that you want to convert. Make sure to select only data objects with the same unit.
2. In the Properties window, enter the new unit in the fields Y, X or Z Data Unit.

Via the Replace or Convert Units dialog box

1. In the object list, select the formulas and data sets with a unit that you want to convert. You can also select data objects that have different units.
2. Select Home[Selected Objects] > Change Units.
3. In the Replace or Convert Units dialog box enter the new units for the units that you want to convert.
4. For the Action, select Convert.
5. End the process by clicking OK.

Replacing Units of Data Sets

In this process, the units are replaced without changing the data.

Via the Properties window

1. Select the data sets that have a unit you want to replace.
2. In the Properties window, enter the new unit in the fields Y, X or Z Unit.

Via the Replace or Convert Units dialog box

1. Select the data sets that have a unit you want to replace. You can also select data sets that have different units.
2. Select Home[Selected Objects] > Change Units.
3. In the Replace or Convert Units dialog box enter the new units for the units that you want to replace.
4. For the Action, select Replace.
5. End the process by clicking OK.

Converting Formulas, Analysis Objects or Data Link Objects into Data Sets

1. Select all of the objects in the object list that you want to convert into data sets.
2. Click on Home [Selected Objects] > Convert into Data Set.

Note When you convert a formula into a data set, you lose the program code. To avoid this, you should lock the formula.

Converting Analysis Objects or Data Link Objects into Formulas

1. Select all of the objects in the object list that you want to convert into formulas.
2. Click Home[Selected Objects] > Convert into Formula.

Note When converting an analysis object into a formula, you lose the ability to set parameters via the Properties dialog box. You can, however, freely edit the resulting formula.

Merging Multiple Data Objects into a List

1. In the Folders window, select the folder in which you want to add the new list.
2. Select one or more data sets or formulas in the object list.
3. Open Insert[Data] > Formula and select List. A new formula that consolidates the selected data objects into a list is created, which you can edit.

3.6 Data Set

The data set is the data object 118 in which FlexPro stores data that cannot be calculated as a formula. When importing multi-channel measurement data, FlexPro creates several data sets, one per measurement channel. Even if you enter data manually, you should create a separate data set for each measured quantity. It is especially important that all values in a data set have the same physical unit.

Data Editing Options

FlexPro offers you several options for directly editing data in data sets.

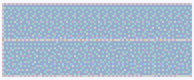
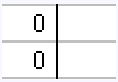
- If the data set to be edited is a data series or a data matrix, you can edit this in an open folder's data view. The data view allows editing across several data sets. This means that you can edit several data sets at the same time in a single data grid.
- In the data set window, you can edit each data set independently of its data structure. However, the data set window always only displays the data of an individual data set.
- If you are measuring the curves in diagrams using the cursors, you can edit the data under the cursor and between cursors directly.

Data Set Window and Data View

The data set window and the data view function much like a spreadsheet in Excel or another spreadsheet program. You can edit them like in a spreadsheet, add, copy and move data and much more.

Data view allows you to work with the data of more than one data set as opposed to when using the data set window. All data sets with the data series and data matrix data structures 122 in the open folder are displayed in one data grid.

The following table clarifies the user interface elements of the data set window:

Element	Purpose
	The upper left corner functions like a button; if you click on this button, the complete data set is selected.
	There are buttons on the left side that number the rows of the data set, starting with zero. By clicking on this button, you can select the complete row in the data set. The buttons on the left side are described as row headers.
	At the top you will find buttons identifying the component of the data set. If the component represents a data matrix, the components are numbered. By clicking on this button, you can select the complete column in the data set. The buttons at the top are described as column headers.
	These cells, which, depending on the structure of the data set, can be found at the bottom and/or on the right, act as placeholders for new values that can be inserted there.
	These cells identify areas where no input is possible. They result from the structure of the data set.
	Dividing lines provide visual separation of data set components, since all components of a data set are always displayed together in the data set window.

The following table describes the user interface elements of the data view:

Element	Purpose
	This button is located directly under the "Close" button. It selects all data.
	On the left-hand side are the buttons that number the rows starting with zero. By clicking on this type of button, you select the complete row in the data view. The buttons on the left side are described as row headers.
	At the top you can see the name row, which shows the names and component identifiers of the data sets displayed in the data view. Selections in the name row correspond to the selections in the object list. By dragging the mouse across the name row, you can rearrange the data sets in the data view.
	Underneath the name row you can see buttons identifying the columns of the data sets. If a data set represents a data matrix, the columns are numbered starting with zero. By clicking on this type of button, you can select the complete

Element	Purpose
	column. The buttons at the top are described as column headers.
	Empty cells at the bottom and/or on the right serve as fields for new values that can be inserted there. If necessary, the data set concerned will be enlarged, or a new data set is created.

In the data view and the object list, you can merge several data series and/or data matrices into a single data matrix 122. To create a new data matrix, for instance, enter the data first as separate columns (data series) and then merge these into a matrix.

FPScript formulas can be used when entering values in the data view or in the data set window. To do this, the text that you input must start with '=', for example '=2 + 3' places the value 5 into the cell. You can also initialize one- and two-dimensional arrays; for instance, '=0#3#4' initializes a matrix with 3x4 elements with zeros.

You can also enter the values with a unit, such as'. '3 mA'. If the particular data set component does not have a unit, then the specified unit is assigned to all data. Otherwise, the specified value is transformed to the unit of the data set. If this is not possible, an error message appears and the input value is discarded.

Assigning Data Sets

The data view offers a particularly convenient way of assigning 125 data sets. Simply specify for the individual data sets whether these are to represent the X, Y or Z component 121 of a composite data structure 122. You can then determine from the order of the data sets which X or Z data sets are to be entered into the Y data sets headers as assigned data sets. This also applies if you have several of these XY or XYZ arrangements, e.g. a X data set, to which three Y data sets are assigned and another X data set to which four Y data sets are assigned. The data view always arranges the Y data sets belonging to a certain X data set to the right of that X data set. You can always assign a data set identified as Y to another X data set by simply using your mouse to move this to position to the right of the other X data set.

Automatic assignment follows this pattern:

1. All Y data sets that follow an X data set without interruption are assigned this X data set.
2. A Y data set with a Z data set immediately to its right is assigned this Z data set.

Some examples of common data structures and their arrangement in the data view:

- Time (X) Signal1(Y) Signal2(Y) Frequency (X) Spectrum1(Y) Spectrum2 (Y)

  Signal1 and Signal2 are time signals with a Time that is assigned as an X data set. Spectrum1 and Spectrum2 are frequency signals with a Frequency that is assigned as an X data set.

- Sampling time(X) Measurements(Y) Trigger time(Z)

The Sampling time and Trigger time data series are entered into the Measurements data matrix as an X or Z component. Together they form a signal series with a Z component.

- XLocus (X) Measurements (Y) ZLocus (Z)

  The XLocus and ZLocus data series are locus coordinates for the data found in the Measured values data series. Together they form a space curve.

Adding Components

You can also add an X or Z component with linearly increasing values to one or more data sets at a later time. This makes sense, for instance, when you know the sampling rate of the measured data, but the time information was not stored in the data file. You can access the relevant commands from the object list and data view.

Working with Data Sets

Creating a Data Set

In the object list:

1. In the Folders window, select the folder where the new data set is to be placed.
2. Open the list box under Insert[Data] > Data Set and select the desired data structure or click on Data Set Wizard to open the wizard where you can also specify the size and data type.

In the data view of the open folder:

- Enter data in an empty column. A data set with the data series data structure is automatically added that contains the data.

Opening a Data Set

To open a data set in its own window:

- Double-click on its name in the object list. If the data set is already displayed in the data view of the folder where it is located, FlexPro does not open a new window, but instead moves the data view to the foreground. Otherwise, FlexPro opens the data set in a new data set window.

To open a data set in the data view:

- In the Folders window, right-click on the folder containing the data set and select Open in the context menu.

Note The data set is only displayed if it has the data structure data series or data matrix.

Arranging Data Sets in the Data View

1. Select the data set you would like to move to a different position by clicking on its name.
2. Drag the data set to the required position and drop it there.

Note: If you change the order of the data sets in the data view, this will automatically change the assignments between the data sets. You will find out more under Assigning Data Sets 134.

Selecting Data Sets in the Data View as Components

1. In the name row of the data view you select all those data sets which you would like to mark as a component.
2. Click Data Design[Data Set] > Component and choose X, Y, Z or None.

Changing the Data Type

1. In the Data View select a data set by clicking on its name in the header. You can skip this step in the data set editor.
2. Click Data Design[Data Set] > Data Type.

3. On the Data Types tab in the data set Properties dialog box you can now select a different data type for the individual components of the data set.

Merging Data Sets to a Data Matrix

In the Data view of an open folder

1. In the name row of the Data view or in the object list, select all data sets you would like to merge into one data matrix.
2. Click Data Design[Columns] > Merge.

In the object list

1. In the object list, select all the records you want to merge into a data matrix.
2. Open the context menu by right-clicking and select Merge to Data Matrix.

Note The resulting data set receives its name and further header information from the data set displayed furthest to the left. If the data sets have differing numbers of rows they might be filled with void values or empty strings.

Concatenating Data Sets of Time Segments of a Long-Term Measurement

Proceed as follows to concatenate the data sets of time segments of a long-term measurement that you have stored in folders:

1. In the object list, select the folders that contain the time segments to be concatenated.
2. Select Insert[Data] > Formula > Concatenation.

Note The formula created returns a list with the concatenated data sets. If the time segments appear one above the other instead of one after the other, this is because the X component of the data sets contains relative time, i.e. always starts with the same value. In this case you should set the argument ShiftXValue of the ConcatenateList function to TRUE.

You can also use a data query 146 to concatenate time segments.

Adding an X or Z Component

1. In the name row of the data view or in the object list, select the data sets to which you want to add a component.
2. Click Data Design[Data Set Layout] > Insert Data Set Element and select the desired component.
3. In the dialog box that now appears, choose whether the additional data component is to be added As separate data series or if it is to be included in the selected data sets.
4. Now add comments and the unit for the component and choose its data type.
5. Use the fields Starting value and Increment to set the linearly increasing or decreasing data for the new data series.

Note If you use the command in data view and the option As separate data series is not enabled, then the data sets to which you have added a component will no longer appear in data view. Data sets with a composite data structure cannot be displayed in data view.

Changing the Column Width

Using the keyboard:

1. In the data view, select one or more data sets by clicking on the name in the header while holding down the CTRL key. You can skip this step in the data set window.
2. Enter the desired column width in the input field Data Design[View] > Width.

Using the mouse:

1. In the data view, select one or more data sets by clicking on the name in the header while holding down the CTRL key. You can skip this step in the data set window.
2. Move your mouse to the dividing line between two column headers.
3. Drag the dividing line to the desired position.

Note This setting can be made for every data set individually and is saved together with the data set. Setting the width for individual components of a data set is not possible.

Working with Data

Navigating and Selecting Data

Using the mouse to navigate and select data:

Selection	Action
Single cell	Click on the cell or press an arrow key to navigate to the cell
Cell Range	Click on the first cell in the range and drag your mouse to the last row or hold down the SHIFT key and then use the arrow keys or Page Up or Page Down to move within the data grid. Alternatively, you can click on a cell and then use your mouse to navigate to the section of the data grid that contains the cell, which represents the opposite corner of the desired selection. Hold the SHIFT key down and click on this cell.
Entire Row	Click on the row header.
Entire Column	Click on the column header.
Adjacent Rows or Columns	Drag the mouse across the row or column headers.
All Cells	Click on the button in the top left corner of the data set window or press CTRL+A.

Using the keyboard to navigate and select data:

Key	Action
CURSOR KEYS	Move the cell selection in the corresponding direction, if possible. If you hold down the SHIFT key, the cells that you cross over are selected.
TAB, SHIFT + TAB	Moves the cell selection to the right or left, if possible.
ENTER KEY	Moves the cell selection down, if possible.
CTRL + HOME	Moves the cell selection to the top left corner of the data grid.

Key	Action
CTRL + END	Shifts the view onto the data grid so that the last row of the data grid is visible. The last cell of the last row is selected.
CTRL + LEFT	Moves the cell selection into the first column of the current row.
CTRL + RIGHT	Moves the cell selection into the last column of the current row.
CTRL + UP	Moves the selection to the first row of the current column.
CTRL + DOWN	Moves the selected cells to the last row of the current column.
SHIFT + LEFT	Extends the current selection by one column to the left.
SHIFT + RIGHT	Extends the current selection by one column to the right.
SHIFT + UP	Extends the current selection by one row up.
SHIFT + DOWN	Extends the current selection by one row down.
SHIFT + CTRL + LEFT	Extends the current selection to the first column.
SHIFT + CTRL + RIGHT	Extends the current selection to the last column.
SHIFT + CTRL + UP	Extends the current selection to the first row.
SHIFT + CTRL + DOWN	Extends the current selection to the last filled row.
SHIFT + CTRL + HOME	Extends the current selection to the upper left corner.
SHIFT + CTRL + END	Extends the current selection to the bottom right corner.
F2	Starts editing a cell value. The cell value is shown as being selected.
Any key	Replaces the text of the currently selected cell with the character entered. If the selected area was previously larger than one cell, it is reduced to the edited cell.

Navigating to a particular cell:

1. In data view, click on any cell in the data set to which you want to navigate. You can skip this step in the data set editor.
2. Click Data Design[Value] > Go To.
3. In the dialog box that now appears, enter the coordinates of the cell to go to via the X index and, if applicable, Z index or via the X value and, if applicable, Z value.

Changing and Adding Values

1. Before you can edit a data set, you have to open it. To do this, double-click on the data set in the object list. If the data set is already displayed in an open folder's data view, the data view is activated and the data set is displayed. Otherwise, a data set window appears, allowing you to edit the data set.
2. Using the mouse or keyboard, move the cell selection to the cell you want to edit. To insert data, select an empty cell.

Key	Function
CURSOR KEYS	Move the cell selection in the corresponding direction, if possible. If you hold down the SHIFT key, the cells that you cross over are selected.
TAB, SHIFT TAB	Moves the cell selection to the right or left, if possible.
ENTER KEY	Moves the cell selection down, if possible.
CTRL + HOME	Moves the cell selection to the top left corner of the data grid.
CTRL + END	Moves the cell selection to the lower right corner of the data grid.
CTRL + LEFT	Moves the cell selection into the first column of the current row.
CTRL + RIGHT	Moves the cell selection into the last column of the current row.

3. Press the F2 key or start entering the desired value immediately.
4. Finish entering the value by pressing the ENTER key.

Your input is converted into the valid data type for the component and inserted into the data set.

Depending on the data type of the data set to which the selected cell in data view belongs, the following rules apply when you input values:

<u>Real Floating Point Values</u>

For a void value, enter '?'.

<u>Complex Values</u>

Both the real and imaginary parts of the complex value are entered in a cell. The data view displays complex numbers in the format '(*real* ; *imag*)' or *real*+*imag*i and also accepts both input formats. When entering data in the format '(*real* , *imag*)', you can omit the parentheses. The two values must be separated by the list separator (usually ',') specified in the Options dialog box.

Boolean Values

If you would like to enter Boolean values, you can either enter the texts 'TRUE' or 'FALSE', or specify 0 for FALSE and 1 for TRUE. The data view itself always displays Boolean values as words.

Time Span Values

Enter these in the format 'D:H:M:S.ssss' or 'H:M:S.ssss'. Enter the fractional digits for seconds up to the desired resolution.

Calendar Time Values

Here, FlexPro accepts the formats 'M/D/Y H:M:S.ssss', 'Y-M-D H:M:S.ssss' and 'D.M.Y H:M:S.ssss'. When entering data, you must always use these formats, even if the data is displayed in a different format.

Note You can use FPScript formulas to enter data. To do this, the text that you input must start with '=', for example '=2 + 3' places the value 5 into the cell. You can also initialize one- and two-dimensional arrays; for instance, '=0#3#4' initializes a matrix with 3x4 elements with zeros. If you want to enter a string that starts with '=' in a text data set, you need to add an apostrophe first in order to prevent your input from being interpreted as FPScript code.

Adding Columns and Rows

To add one or more rows to the data set, which is displayed in the data set window, or to all data sets displayed in the data view:

1. Select as many rows as you would like to insert by clicking and dragging the row headers.
2. Next, select Data Design[Rows] > Insert.

The selected number of new rows is inserted before the row selected.

To add one or more rows to individual data sets in the data view:

1. Select as many rows as you would like to insert as a cell range.
2. Next, select Data Design[Rows] > Insert.

The selected number of new rows will be inserted before the rows highlighted in the relevant data sets.

To add one or more columns to the data set, which is displayed in the data set window, or in the data view:

1. Select as many columns as you would like to insert by clicking and dragging the column headers.
2. Then click on Data Design[Columns] > Insert.

The selected number of new columns will be inserted in front of the highlighted columns.

While in data view, if you had marked the columns in a data matrix, the number of columns will increase. Otherwise, new data sets will be created which start out empty.

Notes

- No rows or columns can be added for scalar values.
- No columns can be added for scalar values, data series, signals and space curves.
- With signal series, columns can only be inserted in the range of the Y component.
- For signal series with a two dimensional X component, at least two columns (for X and Y) always have to be inserted.

Deleting Columns and Rows

To delete one or more rows from the data set, which is displayed in the data set window, or from all data sets displayed in the data view:

1. Select the rows to be deleted by clicking and dragging on the row headers.
2. Then click on Data Design[Rows] > Remove.

The rows selected are then deleted from the data set(s).

To delete one or more columns from the data set that is displayed in the data set window or in data view:

1. Select the columns that you want to delete by clicking and dragging on the column headers.

2. Then click on Data Design[Columns] > Remove.

The columns selected are then deleted from the data set.

Notes

- You cannot delete the only column or row if there are only scalar values.
- No columns can be deleted for signals and space curves.
- The X component and, if present, any Z component cannot be deleted for signal series.
- For signal series with a two-dimensional X component, at least two columns (X and Y) must always be removed.

Copying Data via Drag-and-Drop from the Object List

You can use the mouse to add entire data sets to a data set in the data set window or in data view:

1. Select the data set to be added from the object list.
2. Drag the selected data set to the point in the data set window or data view where you want to insert the data. A gray frame appears in locations where you are allowed to insert data. The frame size depends on the size of the source data set. By moving your mouse, the frame moves in the data grid. If you move the mouse to the edge of the data set window, the visible section is moved as far as possible in the direction of the movement.
3. Move the gray frame to the desired location and then release the left mouse button. The data from the data set is then copied to the selected location. Any previously existing data will be overwritten.

Copying Data via the Clipboard

1. Select the desired data set in the object list or the desired range in a data set window or in the Data view and execute the Copy command using the CTRL+C key combination.
2. In the data set window or in data view, move the cell marker using the mouse or keyboard into a gray cell. Depending on the data structure 122 of the edited data set, these type of cells are located under and/or next to the values that are already present. For scalar values, no new data can be entered.

3. Use the Paste command by pressing the keyboard shortcut CTRL+V. The values from the clipboard are inserted from the insertion position to the right and/or downwards, overwriting any existing data. If the structure of the data in the clipboard does not match the structure of the data set in the Data Set window or the insertion point, the Paste command will not be available.

Copying Data via Drag-and-Drop from a Data Set Window or the Data View

You can insert ranges from data sets into the data set in the data set window using the mouse:

1. In the data set window of the source data set or in data view, use your mouse or keyboard to select the range that you would like to transfer into the target data set.
2. Move the mouse to the edge of the selection. The shape of the mouse cursor changes from the cell cursor to an arrow cursor.
3. Hold down the left mouse button and drag the range to the position in the data view where you would like to copy the data. A gray frame appears at the positions where insertion is possible. The frame size corresponds to the size of the selected range. By moving your mouse, the frame moves in the data grid. If you move the mouse to the edge of the data set window or data view, the visible section is moved as far as possible in the direction of the movement.
4. Next, release the left mouse button at the desired location. The selected data is then copied to the selected location. Any previously existing data will be overwritten.

Note If during the drag-and-drop operation you have to switch to a different window, move the mouse briefly to the window tab that you want to bring to the foreground.

Filling Columns with Values

1. In data view, select the empty columns you want to fill with values by dragging the mouse over the column headers or selecting a cell range. If you select entire columns, the fill range will be set to the longest data set in the data view.
2. Choose the command Fill Column with Values from the Data Set menu or from the context menu. You can request further help in the dialog box that appears.

Calculating Values in Columns

1. In the Data View select the columns that you want to calculate by dragging across their column headers or selecting a cell range. If you select entire columns, the range to calculate is set based on the longest data set in the Data View.
2. Click Data Design[Columns] > Calculate Values. You can request further help in the dialog box that appears.

3.7 Data Query

The data query is a data object 118 used to search for objects with shared properties and to return their data. Data on the hard drive (indexed files) or in the project database can be queried.

For instance, it is possible to search for all objects that have a particular name or a particular unit and return their data as a list 268.

You have the option of concatenating the results of multiple data queries 147. This is useful if you want to perform several data queries using shared search criteria. You can do this by having a data query return the keys of the found data objects and then using them in another data query.

Proceed in a similar manner to limit the data query to particular folders, such as to folders in which a data set exceeds a maximum value. Using your first data query, search for data sets that exceed the maximum value and have it return the keys of the parent or grandparent folders of these data sets. In a second data query, use these keys to limit the search to found parent or grandparent folders.

Consequently, long-term measurements are often stored as files that each contain a time segment of the long-term measurement. Using a data query for long-term measurements 148, you can query for time segments, which span several files, as a data set.

If for the data query you use Output as a list as the result, then you can specify which attribute to use for the list element names. All found often channels have the same name but come from different files. In this case, instead of the channel name, for instance, the file name is used for the list element.

For the data query you can use search criteria that refer to data, such as on the maximum of a data set. When searching for formulas and analysis objects in the project database, it is important that the current result has been indexed. You can make sure of this by making the first criterion for the data query a search criterion that does not refer to the data, e.g. the name, and selecting for the search criterion the option Update and index objects that meet the search criteria before evaluating additional search criteria. This ensures that the data in the index is current when you use a data criterion in the next step.

To be able to use the data query option, indexing must be enabled.

- To query data on the hard drive, file indexing 99 must be set up.
- To query data in the project database, data indexing for the project database 74 must be enabled.

Note Querying data on the hard disk is only possible when the Data Explorer Option 95 is installed.

* This object is not available in FlexPro View and FlexPro Basic.

Working with Data Queries

Setting Up a Data Query

1. In the Folders window, select the folder in which you want to add the new data query.
2. Click Insert[Data] > Data Query.

A data query object is created under a unique name in the displayed folder. Click ? in the dialog box for more help about the setting options.

Concatenating Data Queries

1. Add a new DataQuery 147 data object.
2. On the Options tab, select Search in Project Database or Search in Indexed Files.
3. Select All indexed objects.
4. Define one or more search criteria.
5. Select Objects with Keys as the Result.
6. Add another DataQuery 147 data object.
7. On the Options tab, select Search in Project Database or Search in Indexed Files.
8. Select Objects with keys and select the first data query with the found keys.
9. Define one or more search criteria.
10. Select as the Result Values as list.

11. Repeat steps 6 through 10 for additional data queries that build upon the first data query.

Data Query for Long-Term Measurements

Do the following to request any data set time segment from a long-term measurement stored as a series of files:

1. Add a new DataQuery 147 data object.
2. On the Options tab, select Search in Project Database or Search in Indexed Files.
3. Select All indexed objects.
4. Define the search criterion that selects the data set, such as the data set name.
5. Define two additional search criteria that specify the desired time segment. For this purpose, use for instance the search items Minimum of X component and Maximum of X component.
6. For Sort, select Minimum of X component.
7. Select as Output concatenated (to data series or signal) or Output concatenated and X values of signals increasing.

3.8 Unit Manager

The value of a (measurement) quantity is expressed as a product of a number and unit. FlexPro features a Unit Manager that handles direct import, analysis and presentation of these types of quantities. The customizable Unit Manager in FlexPro is based on the SI unit system and includes the following features:

- Compatibility check and automatic adjustment when calculating quantities of different units.
- Percentage calculation with % and ppm units.
- Ability to select output unit separately for display of quantities in diagrams and tables.
- Support for additional popular unit systems (Gaussian unit system, US unit system).
- Open system architecture in relation to unknown units.
- Manual and automated unit correction during import.

- Ability to define and use custom units.
- Support for the ISO 80000 standard International System of Quantities (ISQ).

The SI Unit System

The International System of Units, "Système International d'Unités" (SI) was introduced in 1960 and is the most widely used unit system for physical quantities. It is managed and internationally standardized by the International Bureau of Weights and Measures, "Bureau International des Poids et Mesures" (BIPM).

Since its revision in 2018, the SI system of units is based on seven defining physical constants and the seven units that define them:

Defining Physical Constant	Symbol	Numeric Value	Unit
Hyperfine transition frequency of Cs	$\Delta\nu_{Cs}$	9 192 631 770	Hz
Speed of light in a vacuum	c	299 792 458	m s^{-1}
Planck constant	h	$6.626\ 070\ 15 \times 10^{-34}$	J s
Elementary charge	e	$1.602\ 176\ 634 \times 10^{-19}$	C
Boltzmann constant	k	$1.380\ 649 \times 10^{-23}$	J K^{-1}
Avogadro constant	N_A	$6.022\ 140\ 76 \times 10^{23}$	mol^{-1}
Photometric radiation equivalent	K_{cd}	683	lm W^{-1}

These physical constants can be used to define the seven base units of the SI system of units:

Base Unit	Symbol	Base Quantity	Symbol	Dimension Symbol
Meter	m	Length	l, x, r, etc.	L
Kilogram	kg	Mass	m	M
Second	s	Time	t	T
Ampere	A	Electric current	I, i	I
Kelvin	K	Temperature	T	Θ
Mole	mol	Amount of substance	n	N
Candela	cd	Luminous intensity	Iv	J

Additional coherent SI units are derived exclusively from these base units. The coherent SI unit of electrical resistance, the ohm with the unit symbol Ω, is, for instance, uniquely defined by the relation $\Omega = m^2\ kg\ s^{-3}\ A^{-2}$, which follows directly from the definition of the physical quantity. Coherent in this context means that no factors other than one occur in the product of powers.

Although this distinction between base units and derived units is no longer necessary due to the redefinition of the SI via physical constants, it has been retained for historical reasons.

For some of these derived units, special names and symbols, such as Pascal (Pa), are in turn used; others are assembled from the existing names and symbols, such as the Newton meter (N m). The notation of the symbols is case sensitive. Symbols derived from a proper name start with a capital letter, such as Newton (N) or Pascal (P). Symbols where this is not the case are written in lower-case, e.g. second (s).

Some measurement quantities span several orders of magnitude. To avoid extra large or small numeric values when displayed, you can place an SI prefix in front of the unit symbol. A unit defined this way, however, is no longer coherent, since the prefix introduces a factor that is not equal to one. Each SI prefix corresponds to a particular power of ten, so for instance the millimeter (mm) is equal to 10^{-3} m, or the kilovolt (kV) is equal to 10^{+3} V.

The appearance of a unit as a product of powers of the base units defines the SI dimension of the respective quantity. This SI dimension is important when checking units for compatibility. Compatible, i.e. interconvertible units, such as km/h and m/s, always have the same SI dimension, as in the example $M^1\ T^{-1}$. Unfortunately, this is not the case the other way round, i.e. units of different types can have the same SI dimension. A familiar example of this are the units Newton meter (N m) for rotational speed and joule (J) for energy. Both have the dimension $L^1\ M^1\ T^{-2}$.

Some derived quantities, in particular quantities defined as ratios of quantities of the same kind, have an SI dimension in which all exponents are equal to zero. Such quantities are simply numbers. The assigned unit is the unit one with the symbol 1, which is usually not written explicitly. The product of powers is thus equal to one, and therefore these quantities are denoted as of dimension one or as dimensionless.

The exponents of the base units usually are whole numbers in the range from ±4. In certain cases, however, fractions can also occur as exponents. Therefore, this results in rational exponents, for instance, when some Gaussian unit system units are displayed in SI base units.

Calculating and Converting Quantities

When making calculations with quantities, you need to account for more than just their numeric values. Thus, for instance, 1 N + 400 mN neither results in 401 N nor in

401 mN, but instead results in 1.4 N. Even the statement 1 N > 400 mN would provide an incorrect result if purely accounting for the numeric values. Other operations are physically useless and should therefore be recognized as such, e.g. 1 V + 1 A. On the other hand, for other calculations a new unit appears for the result, e.g. 6 m / 2 s = 3 m/s or 5 m ^ 2 = 25 m². The unit in which the data is output should also be independent of the unit in which it was acquired, e.g. a speed may be measured in m/s, but output in km/h. The FlexPro Unit Manager covers all of these scenarios. All FPScript mathematical operators as well as all built-in analysis functions not only account for the values of quantities, but also their unit, consisting of the SI dimension, factor, offset and unit symbol. FlexPro is able to compare units prior to calculation, determine the correct unit for the result of a calculation, recognize nonsensical operations, i.e. incompatible units, and transform units for output.

Unit Types

FlexPro makes a distinction between units with known and unknown SI dimensions. Units with an unknown SI dimension cannot be transformed, which means in particular that when using addition and subtraction as well as comparable operations, the units of both operands must match exactly. Multiplication, division and exponentiation, however, can be used without this limitation, and the symbol of the resulting unit is also formed correctly. The FPScript Debugger displays units with an unknown SI dimension in quotation marks, e.g. "people/h". "People/h".

Expandable Unit Collections

FlexPro supports a basic set of units from which you can derive new units through multiplication, division and exponentiation. In addition, you can also use unit collections for the US unit system, the Gaussian unit system and a set of other popular non-SI units, which you can enable as another option.

You can expand on this set of units by registering your own custom units and specifying their SI dimensions. FlexPro manages a unit table in the current project database, in the personal template database, in the global template database and in a number of shared template databases, the locations of which you can specify on the Unit Tables tab of the FlexPro Organizer dialog box. You and your colleagues can store units that you want to share in shared template databases. The unit tables contained within are read by FlexPro when the program is launched.

You can also add units with unknown SI dimensions to the unit tables. This will ensure that FlexPro does not convert the unit or calculate it with other units.

When translating a unit, FlexPro first searches through the unit tables in the order mentioned above and attempts to find an exact match. If no entry is found, the unit is being decomposed into its element. For elements that are not recognized as SI unit symbols, FlexPro searches through the tables again.

Example: FlexPro first searches through the unit tables for the entire unit "Vrms A". If this cannot be found, it will be decomposed into the elements "A" and "Vrms". "A" is the known SI unit Ampere and is accepted as such. "Vrms" is not known to FlexPro, which is why FlexPro searches through the tables again to find it. If an entry is found, the symbol is accepted. The SI dimension specified in the entry is then multiplied with the SI dimension of "A". Otherwise, the entire unit "Vrms A" is viewed as an unknown unit.

Unit Monitoring Modes

You can configure one of the following unit monitoring modes for each project database:

- None
  FPScript ignores all units and performs calculations with numeric values only. Some analysis objects determine the unit of the result using the units entered in the header data of the arguments. In this mode, FPScript is compatible with the previous versions of FlexPro up to version 8.
- Tolerant (preset for new project databases)
  FPScript performs calculations with quantities, i.e. values with a unit, adjusts units to each other during processing, and checks units for compatibility before processing. When output to diagrams, curves of an axis are set to the same unit before being displayed. This mode also permits units whose SI dimension is unknown to FlexPro. These types of units must match exactly when processing two data sets. These types of units must match exactly when processing two data sets. In addition, in this mode, compound units without spaces or multiplication signs between the element units are permitted, such as "Nm" instead of correctly as "N m" or "N·m".
- Moderate
  Specifies the Tolerant setting with the limitation that compound units without spaces or multiplication signs between the element units are not permitted.
- Strict
  Specifies the Moderate setting with the additional limitation that units whose SI dimension is unknown to FlexPro are not permitted.

Note The default setting for newly created project databases is set on the Unit Manager tab of the FlexPro Options dialog box. You can configure the setting for the open project database on the tab of the same name in the project database Properties dialog box.

Physical Quantities

The ISO 80000 series of standards define a large number of physical quantities and their SI units from different domains. FlexPro provides you with assistance when entering a unit and physical quantity for a data set. When you specify a unit, only the physical quantities compatible with this unit are displayed in the physical quantity list box. Conversely, when you choose a physical quantity, only the units compatible with this physical quantity are displayed in the unit list box.

You can specify which domains FlexPro should include on the Unit Manager tab of the FlexPro Options dialog box.

Unit Correction during Data Import

Many measurement systems either do not store the physical unit in the data files at all or they store it in an incorrect notation. A typical mistake, for instance, is to use the symbol S instead of the correct symbol s for the Second unit. In the SI unit system, "S" stands for the unit "Siemens", which is used for conductance. Other examples include RPM instead of 1/min, grdC instead of °C, sec instead of s, and so on. Often, they do not even distinguish between the unit name and symbol, as in the case of VOLT or Volts instead of V, for instance.

This is why FlexPro can provide you with a list of all units found in the data as you start importing it so that you can review and correct them as necessary. During this process, you can add your corrections to a correction table so that FlexPro can make these corrections automatically during subsequent imports.

FlexPro manages a unit table in the current project database, in the personal template database, in the global template database and in a number of shared template databases, the locations of which you can specify on the Unit Tables tab of the FlexPro Organizer dialog box. You and your colleagues can store units that you want to share in shared template databases. The unit tables contained within are read by FlexPro when the program is launched. For automatic unit correction, FlexPro uses the correction tables in the order mentioned above.

Internal Representation of Units

To calculate quantities accurately and convert them into other units, FlexPro stores the following additional quantity attributes in addition to the numeric value(s).

- The unit symbol as a string,
- the denominator and numerator of the exponents of the seven SI base units,
- a factor, which comprises all prefixes used in the symbol or represents the factor between the existing unit and SI base unit (e.g. 1 min = 60 s), and
- an offset, which represents a distance between the unit within the quantity and the respective SI base unit (e.g. 1 °C = (1 + 273.15) K).

FlexPro can use this data to convert a quantity from the existing unit into a compatible SI unit:

$$SIValue\ [SIUnit] = Value\ [Unit] \cdot Factor \left[\frac{SIUnit}{Unit}\right] + Offset\ [SIUnit]$$

with

$$SIUnit := \mathrm{m}^{\frac{N_1}{D_1}}\mathrm{kg}^{\frac{N_2}{D_2}}\mathrm{s}^{\frac{N_3}{D_3}}\mathrm{A}^{\frac{N_4}{D_4}}\mathrm{K}^{\frac{N_5}{D_5}}\mathrm{mol}^{\frac{N_6}{D_6}}\mathrm{cd}^{\frac{N_7}{D_7}}$$

The inverse operation can be used to convert a quantity in the SI unit to another unit with a known SI dimension. Both steps together result in the following formula for converting a quantity from one unit to another:

$$Value_2[Unit_2] = Value_1[Unit_1] \cdot \frac{Factor_1}{Factor_2}\left[\frac{Unit_2}{Unit_1}\right] + \frac{Offset_1 - Offset_2}{Factor_2}[Unit_2]$$

In addition, FlexPro can check units for compatibility prior to calculation. Units are considered compatible if their SI dimensions match. This check is sufficient (the Newton meter and joule units, for instance, have the same SI dimension), but it catches the majority of errors.

You can obtain the two vectors with the denominators and numerators of the seven exponents as well as the factor and offset of a unit in FlexPro using the SIUnits function. The expression `SIUnits(1 km/h)` provides, for instance, a list with the following four elements:

```
Scale = 0.277777777777778

Offset = 0

ExpNum = {1, 0, -1, 0, 0, 0, 0 }

ExpDenom = {1, 0, 1, 0, 0, 0, 0 }
```

The 1 km/h expressed in SI base units is 1000 m / 3600 s = 0.277777777777778 $m^{1/1}$ $s^{-1/1}$.

For non-existing base units, the exponential numerator is equal to zero. The upshot is that for a dimensionless unit, all exponential numerators are always equal to zero. In this case, the exponential denominator is interpreted by FlexPro in a special format in order to distinguish between dimensionless units of varying types. The unit of the solid angle, the steradian (sr), for instance, has the following exponents:

```
ExpNum = {0, 0, 0, 0, 0, 0, 0 }
ExpDenom = {2, 0, 0, 0, 0, 0, 0 }
```

These are interpreted as m^2/m^2. Similarly, the unit of the angle, the radian (rad), provides the following exponents, which are interpreted as m/m:

```
ExpNum = {0, 0, 0, 0, 0, 0, 0 }
ExpDenom = {1, 0, 0, 0, 0, 0, 0 }
```

Since FlexPro compares the denominator and numerator for compatibility when comparing two units, it recognizes that the operation 1 rad + 1 sr is not permitted, even though both have the same SI dimension 1.

References

- Bureau International des Poids et Mesures (2019). *The International System of Units, 9th Edition.* https://www.bipm.org/en/publications/si-brochure
- ISO 80000-1 to -13 series of standards.

Working with the Unit Manager

Setting Up the Unit Manager

Setting up the Unit Manager for the open project database

1. Click File > Info > Project Database Properties.
2. In the Project Database Properties dialog box click on the Unit Manager tab.
3. Here you can change the unit monitoring settings, percentage calculation and unit collections to be activated as desired. If necessary, obtain further help in the dialog box.

Note: Your changes to the settings for FPScript formulas in the project database will take effect immediately. You will have to check and, if necessary, make changes to the FPScript code.

Setting up the Unit Manager default settings for new project databases that you create.

1. Click on File > Options.
2. In the Options dialog box click on the Unit Manager tab.
3. Here you can change the Unit monitoring settings, percentage calculation and unit collections to be activated as desired. If necessary, obtain further help in the dialog box.

Selecting the categories of measurement quantities to be displayed in list boxes.

1. Click on File > Options.
2. In the Options dialog box click on the Unit Manager tab.
3. Select the categories relevant for your work in the Offer quantities from the following categories list.

Checking Imported Units

Depending on the settings of the Units option for data import, the Verify Units dialog box appears. Proceed as follows to check and, if necessary, to correct the imported units.

1. First specify in the Action list box whether you want the units to be edited automatically or manually. Select Edit units to check the units individually and, if necessary to correct them.
2. Point the mouse to an imported unit in the first column of the list to obtain a summary of the data sets that use this unit.
3. Now check the displayed SI dimension. If Unknown appears here, then you should correct the unit. Even if an SI dimension is displayed, you should check it carefully to ensure that the unit is interpreted as expected. If the unit is correct, you can skip the next item.

4. Click on an incorrect unit in the column Replace With and specify the correct unit symbol. Confirm your entry by pressing the ENTER key. Recheck the SI dimension.
5. If nothing appears in the Physical Quantity column, then you should specify a quantity. To do this, click on the empty field in this column and choose one of the quantities displayed. You can also enter a name of your own choice.
6. If the corrections made are to be carried out automatically for future imports, select a storage location in the Save changes list box. Your replacements will then be added to the selected table when the dialog box is closed. You can use the Edit selected table button to check the current content of the table.
7. Click Close to close the dialog box. The selected action is then carried out.

Registering Custom Units

1. Click File > Info > Project Database Properties.
2. In the Project Database Properties dialog box that appears, click on the Unit Manager tab.
3. Click on the button Edit table of custom units on the tab to open the Table of Custom Units dialog box.
4. From the Table list box, choose the table that you want to edit.
5. Click on Insert item or press the INSERT key.
6. In the Unit column of the list, enter the symbol of the new unit using the correct case-sensitive. You cannot use symbols that are already assigned to other units.
7. Now click on the field to the right of the symbol and enter the representation of the unit in SI units. Please note the following rules for entering a unit:
   - You can use any SI units you want during input and not just the seven base units, such as "N" or "W".
   - Enter "1" for a dimensionless unit.
   - Unit symbols are case-sensitive, which means, for instance, that "S" is the symbol for the unit "Siemens" and "s" is the unit for seconds.
   - Use a space for multiplication, e.g. "N m" instead of "Nm".
   - Use the '^' sign when using powers, e.g. "m^2".
   - Use the '/' sign for division, e.g. "m/s^2".
   - Use parentheses correctly, e.g. "kg/(kW h)".

- You can specify an SI prefix or a factor, e.g. "mm" or "0.001 m".
- You can specify an offset, e.g. "K−273.15".

8. If you want to allow the use of an SI prefix for your new unit, click in the Allow SI Prefix column and select Yes.
9. Click Close to accept your changes.

Note: You cannot edit the unit tables in the shared template databases directly. For instance, to add a unit to a shared template database, you first have to register it in the current project database or the personal template database as described here and then copy it to the shared template database, as described in Organizing Units 159.

Editing Tables for Unit Correction during Data Import

1. Click File > Info > Project Database Properties.
2. In the Project Database Properties dialog box that appears, click on the Unit Manager tab.
3. Click on the button Edit table for correcting units during data import on the tab to open the Data Import Unit Correction Table dialog box.
4. From the Table list box, choose the table that you want to edit.
5. Click on Insert item or press the INSERT key.
6. In the Unit column of the list, enter the symbol of the new unit using the correct case-sensitive.
7. Now click on the field to the right of the symbol and enter the symbol that will replace this unit. If the unit you enter is known, a check mark will appear at the beginning of the row. Otherwise, an exclamation mark will appear.
   - Unit symbols are case-sensitive, which means, for instance, that "S" is the symbol for the unit "Siemens" and "s" is the unit for seconds.
   - Use a space for multiplication, e.g. "N m" instead of "Nm".
   - Use the '^' sign when using powers, e.g. "m^2".
   - Use the '/' sign for division, e.g. "m/s^2".
   - Use parentheses correctly, e.g. "kg/(kW h)".
   - You can specify an SI prefix or a factor, e.g. "mm" or "0.001 m".

- You can specify an offset, e.g. "K−273.15".

8. Click Close to accept your changes.

Note: You cannot edit the unit tables in the shared template databases directly. For instance, to add a unit to a shared template database, you first have to register it in the current project database or the personal template database as described here and then copy it to the shared template database, as described in Organizing Units 159.

Unit Organizer

Use the Organizer dialog box to delete units from the custom units table or from the unit correction table during data import or to copy units from one database to another.

Opening the Organizer dialog box

1. Click on File > Info > Organizer.
2. In the Organizer dialog box click on the Unit Tables tab.

Opening the current project database or a template database for Organizing

- Choose a template database in one of the Items available in list boxes.

Opening a shared template database or any other database for organization

1. Click on one of the Close Project Database buttons to close the database currently displayed in the list.
2. Click on the Open Project Database button and select a project database on your hard drive or from the network.

Copying units from one project database to another

1. First, in the Table field, choose whether you want to edit the custom units table or the unit correction table for data import.
2. Now select the units that you want to copy and click Copy.

Deleting units from a project database

1. First, in the Table field, choose whether you want to edit the custom units table or the unit correction table for data import.
2. Now select the units that you want to delete and click Delete.

3.9 Data Import

FlexPro offers you various data import options, which are presented below:

Importing Data from Binary Files

FlexPro supports binary file formats from many different measuring systems and data acquisition programs and can import these directly. A generic import filters for binary data is also available. For most data formats, you can choose whether you want to import individual channels from a file or whether to import the entire file. FlexPro can read several files of different formats in a single step. Optionally, a folder of the same name can be created in the project database for every file to be imported.

Importing Data from Text Files

You can import text files using the Text Data Import Wizard. All settings that you specify while importing a file can be saved as a new import schema. This allows you to call up any number of files with the same structure at a later stage without having to use the wizard again. You can obtain additional information by accessing help in the Text Data Import Wizard.

Importing Data from Excel Files

You can use the Excel Data Import Wizard to import data that is arranged in columns or rows directly from Microsoft Excel XLS or XLSX files. The wizard works similarly to the Text Data Import Wizard and also supports import schemas. Microsoft Excel does not need to be installed on the machine for this type of file import.

Data Link Object

Not only can you copy the data from binary, text and Excel files into the FlexPro project database, but you can also create links to the original data. Links are created via data link objects, which read the data of a channel from a file, if necessary. You can use the data link object as you would a data set or a formula. Data link objects are special formulas that have parameters you can set using a dialog box.

If the file you want to import contains several data sets, such as a multi-column text file, then an individual data link object is set up for each data set. This object only reads the section assigned to it from the file. You cannot to set up the data link objects manually. FlexPro automatically creates these when you select the Create links option during import. FPScript Read...File 339 import functions are available for various data formats that read out a channel from the file. These are used in the formulas of data link objects.

Since data link objects maintain the link to the file to be imported, you can therefore exchange data dynamically between the application that writes to the file and FlexPro. You can transform data link objects into data sets at any time. When updating, the formulas of data link objects are only considered invalid if the external file has been changed since last read in. This ensures that unnecessary re-calculations are avoided.

Some data link objects can read comments and units from the files to which they are linked. You can activate this feature by leaving the corresponding input fields empty in the Properties dialog box of the import object.

Channel Selection Filters

Many FlexPro import filters support saving the current channel selection as a channel selection filter. When importing the same files repeatedly, you can then select this type of channel selection filter when you don’t want to import all of the data in the file.

FlexPro saves the channel selection filters in your personal template database, which you can managed under File > Info > Organizer. Here you have the option to copy a channel selection filter to a different database in order to provide the channel selection filter to all users, for instance. You can import or export a channel selection filter as an XML file as well.

A file-based channel selection filter saved as an XML file can also be used on other computers. To do this you only have to copy the file to the following folder: To do this you only have to copy the file to the folder C:\ProgramData\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Templates\Import\Filter.

Importing Data from ODBC Data Sources

FlexPro can import data from ODBC data sources.

Open Database Connectivity (ODBC) is an international standard for manipulating relational data using the SQL syntax. ODBC offers a standard interface for accessing relational data sources using what are known as ODBC drivers, which exist for various project database systems.

ODBC is part of MDAC (Microsoft Data Access Components), which is supplied in different versions with the different Windows operating systems. You can view a list of the ODBC drivers available to you by opening the Control Panel, clicking on Administrative Tools and under Data Sources (ODBC) clicking on the Drivers tab. You can download the latest version of MDAC from Microsoft’s website.

Importing Data Using Data Explorer

If the Data Explorer option is installed, FlexPro can index all your data on the hard drive. You can use the Data Explorer to search through the data and import it into the object list using drag & drop, for instance.

Importing Data Using DataQuery

If the Data Explorer option is installed, you can use the DataQuery object to directly access all data that FlexPro has found and indexed on your computer. The object directly imports the data to which the query applies and passes it as a list 124.

Importing Data from ASAM ODS Data Sources

ASAM ODS is a standard used in the automotive industry for data management. If the ASAM ODS Data Import option 170 is installed, FlexPro can create links to and import data from ASAM ODS conforming data sources.

Importing Data from ISO/TS 13499 MME Data Sets

The ISO/TS 13499 MME format is widely used in the automotive industry as a multimedia format for crash tests. FlexPro can import data, pictures and videos from ISO/TS 13499 MME data sets. Pictures and videos are added as media objects that remain linked to the original files.

Importing Data via OLE

OLE (Object Linking and Embedding) is a standard protocol for data exchange in Windows. To import data using OLE, the application that is to supply the data must have an OLE server interface.

In FlexPro, you use OLE data links to access data from the OLE server, e.g. Microsoft Word. You can use these links like normal data sets within FlexPro. Whenever the data in the OLE data link is accessed, the request is passed on to the server application.

The server application must offer the data in text format so that FlexPro can convert the data. If you transfer a single column, the OLE data link provides this column in the form of a data series. If you link several columns, then they are provided as a data matrix. Values specified with "#NV" are imported as void values.

Importing Data

Importing Data from Files of Other Programs

a) Using the Import command:

1. Select the folder containing the data that you want to import into the Folders window.
2. Click on the Data tab of the ribbon.
3. The options in the Import Settings group correspond to those on the Import dialog box. You can adjust the settings here or in the dialog box.
4. Now click on Binary Data in the Import from Files and Databases group.
5. In the dialog box that appears, select the file format that you want to import from the File Type list. Select All Files if you want to import several different file types. Select Text Files or Binary Files if the specific format is not listed.
6. Select the file(s) you would like to import.
7. Click on Open to import the data. A dialog box or wizard appears depending on the selected format and import settings. Access Help if you require additional assistance.
8. Depending on how the Units option is set, the Verify Units dialog box appears. For information on how to use this dialog box, see Checking Imported Units 156.

Notes:

- The file type Text Files is already selected if in step 4 you click on Data[Import from Files and Databases] > Text Data instead of Data[Import from Files and Databases] > Measurement Data.
- The availability of the import settings described above depends on the import format selected.

- If you import read-only data sets, then the data editor and any other operations that could change the data are disabled. For instance, you will not be able to eliminate or interpolate any outliers with the cursors.
- When specifying the text data setting, the text data must be arranged in columns. Each column should have a title to which the physical unit can, as an option, be appended in square brackets. The decimal symbol must correspond to the Regional and Language Options set in the Windows Control Panel.

b) Importing data using drag-and-drop:

You can import data very quickly if you use drag-and-drop to drag the data directly from Windows Explorer to the object list.

1. Select the files in Windows Explorer that you want to import.
2. Drag the data from Windows Explorer to the object list. If you want to link the data instead of copying it, you hold down the CTRL and SHIFT keys while you drop the files into the object list. A linking symbol should appear on the cursor.
3. All imported units are first corrected automatically using the unit correction tables in the project database and template databases. If units remain that are unknown to FlexPro, the Verify Units dialog box appears. For information on how to use this dialog box, see Checking Imported Units 156.

Note Not every import filter supports linking. Importing using drag-and-drop always imports the entire files. Otherwise, the import is done via the Import Settings set on the Data tab.

Importing Excel Data

1. In the Folders window, select the folder where you want to place the object.
2. Click Data[Import from Files and Databases] > Excel Data.
3. Select the file(s) you would like to import.
4. Click Open.
5. If necessary, request further help in the wizard that now appears.

Importing Data via the Clipboard

a) Importing data into the FlexPro project database

You can import values or ranges of values from other applications directly into the project database. The data will then be stored as new data sets in the object list. The pre-condition for this is that the external application must be able to provide the values in text format in the clipboard or by using a drag-and-drop operation. To import values from Microsoft Word, for instance, do the following:

1. Start Microsoft Word and open the document where the data you want to import into FlexPro is located.
2. Select the text or the table to be transferred and select Home[Clipboard] > Copy.
3. Switch to FlexPro and click in the object list.
4. Use the command Home[Clipboard] > Paste.
5. A dialog box now appears where you can specify how the data is to be imported. Select the options you wish to use and close the dialog box by clicking on OK.

...or use drag-and-drop.

1. Start Microsoft Word and open the document containing the data you want to import into FlexPro.
2. Select the text or table to be transferred.
3. Move the mouse cursor onto the selected area. Hold down the left mouse button and the CTRL key and move the mouse onto the object list while keeping the CTRL key and mouse button pressed. When the mouse pointer turns to a + symbol, this indicates that the data is being copied.
4. Now release the left mouse button and then the CTRL key.
5. A dialog box now appears where you can specify how the data is to be imported. Select the options you wish to use and close the dialog box by clicking on OK.

Notes

- You can select single values or one or more columns of values and import them in a single operation. If you just selected two columns, you can import them as a signal, for instance. In this case, the first column specifies the X values and the second specifies the Y values of the signal. If you have selected multiple columns, you can import them as a data matrix 123 or as separate data series 123. A data matrix is a two-dimensional data set that is suitable for a matrix of values, for instance. If you import several data series, then a separate data set is created for each column selected. Here you can specify if a specific column is to be used as an X data set. The name of this data series is then entered as an assigned data set 125 in the headers of all Y data series.

- If you import data series or signals, the first line of the selected text may contain the captions of the columns. FlexPro then uses them as the names of the data sets to be created. As an option, you can append the physical unit set off by square brackets, such as "Temperature [°C]"

b) Importing data into an open data set or the data view of an open folder

You can import values or ranges of values from other applications into an open data set or the data view of an open folder. Also in this case, the values must be provided in text format. To import values from Microsoft Word, for instance, into a data set window, do the following:

1. Start Microsoft Word and open the document containing the data you want to import into FlexPro.
2. Select the text or the table to be transferred and select Home[Clipboard] > Copy.
3. Switch to FlexPro into the data view or the data set window where you want to insert the values.
4. Select the cell that is to represent the top left corner of the range or field to be inserted.
5. Use the command Home[Clipboard] > Paste.

...or use drag-and-drop.

1. Start Microsoft Word and open the document containing the data you want to import into FlexPro.
2. Select the text or table to be transferred.
3. Move the mouse cursor onto the selected area. Press the left mouse button and move the mouse onto the data view or the data set window where you want to insert the values. The area of insertion is indicated by a rectangular frame.
4. Release the left mouse button after you have selected the insertion point. The values are moved from Word into the selected area or are copied there if you keep the CTRL key pressed as you release the left mouse button.

Note When importing data from external applications, the import process fails if FlexPro cannot convert the data into the data type of the import target area. In addition, empty lines between the values are imported as void values. Empty lines at the beginning are ignored.

Importing Data Using Data Explorer

"Indexed Data" View

Only one file or channel can be selected in the tree view, whereas multiple channels can be selected at the same time in the list view.

In contrast to File Explorer, this view allows importing of complete files as well as individual channels.

a) Importing Data via the Clipboard

1. Select the file or channels in Data Explorer that you want to import.
2. Right-click with the mouse and from the context menu choose Copy.
3. Move the mouse cursor over the object list, and from the context menu choose Paste.

b) Importing data using drag-and-drop

You can also import data if you use drag-and-drop to drag the data directly from Data Explorer to the object list.

1. Select the file or channels in Data Explorer that you want to import.
2. Drag the data from Data Explorer to the object list. If you want to link the data instead of copying it, you hold down the CTRL and SHIFT keys while you drop the files into the object list. A linking symbol should appear on the cursor.

c) Importing data using Data Explorer commands

1. Select the file or channels in Data Explorer that you want to import.
2. Right-click with the mouse and from the context menu choose Copy to Project Database. If you want to link the file instead of copying it, choose the following from the context menu: Create Link(s) in Project Database.
3. Copy All Channels of Selected File(s) to Project Database or Create All Channels of Selected File(s) As Link(s) in Project Database. This allows you to select one or more channels and import all channels of the associated files.

"All Files" View

Importing data from Data Explorer is handled similarly as importing from Windows Explorer.

Only one file can be selected in the tree view, whereas multiple files can be selected at the same time in the list view.

a) Importing Data via the Clipboard

1. In Data Explorer select the files that you want to import.
2. Right-click with the mouse and from the context menu choose Copy.
3. Move the mouse cursor over the object list, and from the context menu choose Paste.

b) Importing data using drag-and-drop

You can also import data if you use drag-and-drop to drag the data directly from Data Explorer to the object list.

1. In Data Explorer select the files that you want to import.
2. Drag the data from Data Explorer to the Object List. If you want to link the data instead of copying it, you hold down the CTRL and SHIFT keys while you drop the files into the object list. A linking symbol should appear on the cursor.
3. All imported units are first corrected automatically using the unit correction tables in the project database and template databases. If units remain that are unknown to FlexPro, the Verify Units dialog box appears. For information on how to use this dialog box, see Checking Imported Units 156.

Note Not every import filter supports linking. Importing using drag-and-drop takes into account some of the Import Settings set on the Data tab.

Importing Data Using DataQuery

You can import several data sets as a list by querying the external data (indexed data).

Note: To do this, indexing must be enabled. Querying external data is only possible when the Data Explorer option is installed. Select the search criteria for the data query so that the result does not contain too many channels/data objects, since this can have a strong impact on the processing speed.

Importing sample data

Import the ultrasound signals from the file samples provided. The prerequisite is that the folder `C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples\Data` has already been indexed.

1. Add a new DataQuery 147 data object.
2. On the Options tab, select Search in Indexed Files.
3. Select Objects in folder: `C:\Users\Public\Documents\Weisang\FlexPro\%VERSION_COMMERCIAL%\Examples\Data`
4. Select Include subfolders in search.
5. Define a search criterion Name is Ultrasound Signal.
6. Under Result and Output Format Values and Output as a list.

Single channels / output data objects

1. Use the command Run Query to test the data query. Use the Show Legend button display the results of the data query in a list or tree view.
2. Select Output selected items and then click on Select Items.
3. Select only one of the found ultrasound signals.

Importing Data from ODBC Data Sources

Importing data using ODBC is similar to the process of importing data from files:

1. Select the folder containing the data that you want to import into the Folders window.
    1. Click on Data[Import from Files and Databases] > ODBC Data Source.
    2. Select a data source from which you want to import data in the Select Data Source dialog box, or create a new data source.

3. The ODBC Data Import dialog box appears where you can select the columns to import from the tables present in the data source. If necessary, call up Help in the dialog box by clicking on '?'.
4. The import process starts when you press Copy or Link.

Some data formats, such as Microsoft Excel workbooks (extension .XLS), Microsoft Access databases (extension .MDB) and also ODBC data source files (extension .DSN), can also be selected and imported directly as files.

ASAM ODS Data Import Option

ASAM ODS is an automotive industry standard and defines a generic data model for the interpretation of data, interfaces for managing models, saving data, accessing data as well as exchanging data.

The data model consists of a base model, an application model and instances. The base model is the same for all applications and is defined by ASAM ODS. It consists of base elements and base relations (relationship between two base elements). For instance, AoMeasurementQuantity is a measured physical size. The model classifies the available information into different areas:

- Test result management: AoTest, AoSubTest, AoMeasurement
- Test result storage: AoSubMatrix, AoLocalColumn
- Information: AoUnitUnderTest, AoTestEquipment, AoTestDevice etc.

The application model is application-specific, which means that different applications are described by different application models. It is based on the base model and determines what type of elements are actually represented in the data storage.

Both models contain meta information about the structure of the stored data. The data is stored in instance elements. An instance element is clearly identified by a name and ID. A value is stored in an instance element for each application attribute. In addition, an instance element can contain attributes that were not defined in the application model (instance attributes). The instance elements can have relationships to other instance elements.

These instances (instance elements, attributes, relationships) are displayed in the Windows ASAM ODS Data Source. You can define particular views that display only part of the information. You can also search for instance elements.

- Click Data[Other Data Sources]> ASAM ODS.

Note If an ATF or ATFX file is loaded via the Import dialog box, the ASAM ODS Data Source window appears automatically.

Select Connection in the toolbar of the ASAM ODS Data Source window or in the context menu.

In the Connection dialog box, you can select an ASAM ODS data source to be displayed in the ASAM ODS Data Source window. The following settings are possible:

Connection type

- No connection: No connection to an ASAM ODS data source is established.
- CORBA: Establishes a connection to an ASAM ODS server via a CORBA interface.
- ATF: Establishes a connection to an ASAM Transport Format (ATF) file.
- ATFX: Establishes a connection to an ASAM Transport Format in XML (ATFX) file.

File

Specifies the path of the ATF or ATFX file to be displayed in the ASAM ODS Data Source window.

Server

The name of the machine on which the ASAM ODS server is located.

Port

Specifies the port.

Factory name and type

Every ASAM-ODS server that is available in the network and is registered with an ORB offers a so-called factory interface and thus enables clients to obtain general information about the server and to establish a connection with this server. The factory interface is uniquely described by name and type. Existing factory interfaces are automatically listed.

User

Specifies the user of the ASAM ODS server.

Password

Specifies the user password.

Read entire data source

Specifies whether the data source is read in completely or whether only the instance element specified via the ASAM path, including its subelements, is read in.

ASAM Path

Determines the ASAM path of the instance element that is to be read (see Reading the Data Source Completely).

Importing Measurement Data

You can import all measurement data of an instance element marked in the ASAM ODS Data Source window. The selected instance element must have the value AoTest, AoSubTest, AoMeasurement, AoMeasurementQuantity, AoSubmatrix or AoLocalColumn as base element. The system searches for all value attributes of the instance elements that were derived from the base element AoLocalcolumn and that are subordinate to the selected instance element. For elements of type AoTest, AoSubTest and AoMeasurement, subfolders are created in FlexPro so that the measurement data is stored hierarchically during import.

1. Click with the left mouse button on the instance element that contains the data to be imported.
2. Select the Import Data command from the toolbar of the ASAM ODS Data Source window or from the context menu.

If you want to import attributes automatically in addition to the measurement data, select the Import attributes option in the ASAM ODS Options dialog box.

If you want to import additional information automatically in addition to the measurement data, select the Import additional information option in the ASAM ODS Options dialog box.

Importing Data with Special Properties

It is also possible to search 176 for instance elements with certain properties and then import the measurement data containing the instance elements found. If the instance elements you are looking for do not contain any measurement data, no data will be imported.

1. Search for specific instance elements by selecting the Search command on the toolbar of the ASAM ODS Data Source window or by clicking Find in the context menu and defining a search filter.
2. Left-click the Search Results node that contains the instance elements.
3. Select the Import Data command from the toolbar of the ASAM ODS Data Source window or from the context menu.

1. In the ASAM ODS Data Source window, right-click the instance element whose attributes you want to import.
2. In the context menu, choose Import Attributes.

You can use the mouse to copy ASAM ODS attributes from the ASAM ODS Data Source window to the FlexPro object list:

1. Select the attribute to be inserted in the ASAM ODS Data Source window.
2. Drag the data from the ASAM ODS Data Source window to the object list.
3. If you want to link instead of copy, you can activate the Import data as links option under Options 177 in the ASAM ODS Data Source window.

1. Click Data[Import from Files and Databases] > Binary Data.
2. In the dialog box that appears, select the ASAM ODS file format from the File Type list.

3. Select the file you want to import.

The ATF/ATFX file is now displayed in the ASAM ODS Data Source window. You can select individual attributes and copy them to the FlexPro database.

- Select one of the defined views from the list box on the toolbar of the ASAM ODS Data Source window.

- Select Customize View from the toolbar of the ASAM ODS Data Source window or from the context menu.

In the ASAM ODS View dialog box you can set which information of the ASAM ODS data source is displayed. The complete application model of the current ASAM ODS data source is displayed. This model can be manipulated, i.e. elements, attributes and relationships can be hidden or moved.

A defined view refers to a specific ASAM ODS data source because each data source has its own application model.

Before the application model for the view can be manipulated, a name for the view must be entered and the view saved.

The following commands are available:

Save View

Saves the current view under the specified name.

Delete View

Deletes the selected view.

Hide Element

Removes the element selected in the tree (application element, attribute, relationship) from the view. If the element has subelements, these are also hidden.

Show Elements

Displays all subelements of the selected application element.

Root Element

Moves the application element to the root or to its original position.

Show Application Attributes

Shows or hides all application attributes.

Show Instance Attributes

Shows or hides all instance attributes. Instance attributes are attributes that are not defined in the application model.

Show Relations

Shows or hides all relationships.

Reset

Restores the original application element.

In the ASAM ODS Data View, you can display the instance elements that have a relationship to a selected instance element.

1. In the ASAM ODS Data Source window, select the relationship node of an instance element.
2. Click Show Relationship in the context menu.

The corresponding instance element is selected in the data view. If the instance element does not exist in the view, it is appended to the relationship node as a subnode.

You can search for specific instance elements and display the search result as a separate tree in the ASAM ODS Data Source window.

1. On the toolbar of the ASAM ODS Data Source window, choose Search or click Search in the context menu.
2. Enter a name for the search filter in the combo box or select an existing search filter.
3. In the list box, select the type (application element) to search for. As the result, all instance elements will be listed that originate from the selected application element and that fulfill the defined search conditions.
4. Define one or more search conditions. If several search conditions are defined, you can logically combine the search conditions. You can influence the priority of the logical combinations by using parentheses. If no parentheses are set, the conditions are linked one after the other (the first two results are linked, the result is linked to the third condition, and so on).
5. Click the Save Search Filter button if you want to save the search filter. Since the search filter depends on the application model, the search filter depends on the active ASAM ODS data source.
6. Click the Find button. The result is displayed in the ASAM ODS tree as a new node.

You can create, edit, and delete search conditions.

You can select different search conditions in a dialog box. The search conditions always refer to a selected attribute, i.e. the selected attribute must fulfill a certain condition.

The following settings are possible:

- Attribute: The system searches for instance elements that contain an attribute that fulfills the search conditions.
- Parent element: The system searches for instance elements whose parent element contains an attribute that fulfills the search conditions.
- Child element: The system searches for instance elements whose child element contains an attribute that fulfills the search conditions.

- Relationship to element: The system searches for instance elements whose relationship element contains an attribute that satisfies the search conditions.
- Relationship from element: The system searches for instance elements whose relationship element contains an attribute that satisfies the search conditions.
- Mutual relationship: The system searches for instance elements whose relationship element contains an attribute that satisfies the search conditions.

1. Select an existing search filter in the Search dialog box.
2. Click the Remove Search Filter button.

The following settings can be made in the ASAM ODS Data Import Options dialog box:

Import Data as Links

If you do not want to copy the ASAM-ODS data but want to link it, select the Import data as links option.

Import Attributes

If you want to import attributes automatically in addition to the measurement data, check the Import attributes option. If you execute the Import Data command, additional data objects are created in FlexPro. An object contains the value of an attribute that contains the selected instance element or one of the sub-elements. The base element of the instance element must be AoTest, AoSubtest or AoMeasurement.

Import Additional Information

If you would like to import other information automatically in addition to the measurement data, check the Import additional information option. If you execute the Import Data command, additional data objects are created in FlexPro. An object contains the value of an attribute of an instance element that was derived from one of the tables AoUnitUnderTest, AoUnitUnderTestPart, AoTestEquipment, AoTestEquipmentPart, AoTestDevice, AoTestSequence, or AoTestSequencePart and has a relationship to the measurement to be imported.

Use ID in Name of Instance Elements

Select this option if you want the ID of the instance elements to be displayed next to the name in the tree view.

Importing Data Using OLE

In this section, you will create an OLE data link. The process depends on the server application from which you would like to import the data. Here we describe the process using the popular word processor Microsoft Word.

1. In Word, open the document from which you would like to import the data.
2. Select a text or a table with numerical data.
3. Use the command Home[Clipboard] > Copy to copy the OLE object to the clipboard.
4. In the Folders window, select the folder where the OLE data link is to be inserted.
5. Open the list box under Home[Clipboard] > Paste and select Paste Special to insert the OLE data link.
6. In the Paste Special dialog box, select Link.
7. Click on OK.

Working with Channel Selection Filters

Creating a Channel Selection Filter

During File Import

Some import filters support saving the current channel selection as a channel selection filter.

1. Click on the Data tab of the ribbon.
2. Under Import Settings > Channel selection select Manual.
3. Now click on Binary Data in the Import from Files and Databases group.
4. In the dialog box that appears, select the file format that you want to import from the File Type list.
5. Select the file that you want to import.
6. Click on Open to import the data.

7. In the Channel selection dialog box, select the data channels that you want to include in the channel selection filter.
8. In the Channel Selection Filters list box, select <Save selection...>.

If no list box with Channel Selection Filters is available, then the selected import filter does not support saving channel selection filters and you will need to create the filter using the Organizer dialog box as described below.

---

9. In the Save dialog box, enter a name for the channel selection filter and click OK.
10. In the Channel selection dialog box, click Finish if you want to import the data now. Otherwise, click Cancel.

Note: A channel selection filter that you create this way for a certain file format will not always work with other file formats. This is especially true in the case of formats that organize data hierarchically. It is better to create cross-format channel selection filters in the Organizer dialog box.

Using the Organizer Dialog Box

1. Click on File > Info > Organizer.
2. In the Organizer dialog box, click on the Channel Selection Filters tab.
3. Select a template database in which you want to create the channel selection filter from one of the Items available in list boxes.
4. Click Add.
5. In the edit Channel Selection Filter dialog box, create the channel selection filter and give it a name. If necessary, obtain further help in the dialog box.

Using Channel Selection Filters

1. Select the channel selection filter from the Data[Import Settings] > Channel selection list box.
2. Import the data on which the channel selection filter will be used.

Organizing Channel Selection Filters

The Organizer dialog box is used to delete channel selection filters or to copy them from one database to another.

Opening the Organizer Dialog Box

1. Click on File > Info > Organizer.
2. In the Organizer dialog box, click on the Channel Selection Filters tab.

Opening the Current Project Database or a Template Database for Organizing

- Choose a template database in one of the Items available in list boxes.

Opening a shared template database or any other database for organization

1. Click on one of the Close Project Database buttons to close the database currently displayed in the list.
2. Click on the Open Project Database button and select a project database on your hard drive or from the network.

Copying Channel Selection Filters from One Project Database to Another

1. Select the channel selection filters that you want to copy.
2. Click on Copy.

Deleting Channel Selection Filters from a Project Database

1. Select the channel selection filters that you want to delete.
2. Click on Delete.

3.10 Data Export

Exporting Data to a File

FlexPro offers you the option of exporting data sets, formulas and related objects in diverse formats.

When saving in CSV or text format, you have the option of exporting several objects below one other or side-by-side to a single file. You have the flexibility to compile header data for export, select different character encodings, specify the decimal separator and whether the formula result or text should be output. CSV stands for Comma Separated Values. In this format the values are not separated by tabs, but are instead separated by commas or semicolons.

The same flexibility also makes it possible for FlexPro to save data sets directly in the Microsoft Excel XLSX format.

Only signals and signal series with two or more signals are suitable for export as an audio file. Signals are treated as mono signals, signal series with two signals are treated as stereo signals, and signal series with more than two signals are treated as multi-channel audio signals. The X component of the data must correspond to one of the following sampling rates supported by the audio file format: 11025 Hz, 22050 Hz or 44100 Hz.

Note: Please note that, depending on the file format, when exporting data, it is possible that data could become lost when rounded. Some file formats save the data using a lower resolution.

FlexPro Data Exchange Format

The FlexPro data exchange format is similar to that variant of the text format, where the data of several data sets are stored side-by-side and with all of the header information. Here, however, an additional row with a data format identifier is then output. Regardless of the system settings, floating point values are always output with a point as the decimal symbol. Calendar times are also always output in the same format, regardless of the system settings. This format has the following characteristics:

- The structure of the data format is simple, in that it is relatively easy to integrate corresponding export and/or import routines into programs that want to exchange data with FlexPro on a file basis.
- The complete header information including parameter lists is stored. This means that no information is lost when data is exchanged.
- The data format corresponds to a single table in ANSI code. The tab character (09 Hex) serves as column separator. The table is completely square, meaning that all rows have the same number of columns. This is why the format is particularly suitable for exchanging data with spreadsheet programs.
- The data format is independent of the Region and Language system settings. Floating point values are always output with a decimal point, and calendar settings are always output in the format MM/DD/YYYY HH:MM:SS in local time.

- Formulas can either be stored like data sets with data or purely as formulas without data. During import, data sets are created for the first case, and in the second case, formulas are created.
- Data with linearly increasing values can be stored using the two parameters called starting value and increment, which means that in this case the corresponding data columns may remain empty.

The following table describes the data format in detail:

Row	Contents
1	Version identifier: FlexPro Text Data File Version 5.0 at UTC+/-TimeShift The TimeShift number following UTC specifies the delay in hours relative to UTC (Coordinated Universal Time). This allows calendar times in the file to be converted back to UTC.
2	Names of the data sets and formulas contained inside. The names are shown in the first column of the applicable data set. If a data set has several data columns, several column separators can come between its name and that of the neighbor to the right.
3	String with data structure identifier for each data set. The format of the identifier is <data structure> <number of columns> <number of rows> <Y data type> <Y formatter> <X data type> <X formatter> <Z data type> <Z formatter>. The following keywords are used for the data structure: Scalar, DataSeries, DataMatrix, Signal, SignalSeries, SignalSeriesZ, SignalSeriesX, SignalsSeriesXZ, SpaceCurve and Formula. The X following SignalSeries specifies that it has a two-dimensional X component, i.e. a separate X column exists for every Y column. The Z following SignalSeries specifies that it has a Z component. The number of rows and columns is only shown if the number of dimensions of the data structure requires this. The Formula keyword specifies that only a formula but no data has been stored for the corresponding name. In this case, data types and dimension lengths are not shown. Only the three formatters follow the keyword. This type of formula without data is stored like a space curve with three columns, which, however, remain empty. The following keywords are used for the data types of the individual components: Integer16, Integer32, Integer64, FloatingPoint32, FloatingPoint64, ComplexInteger16, ComplexInteger32, ComplexInteger64, ComplexFloatingPoint32, ComplexFloatingPoint64, Boolean, String, DateTime, DurationHMS, DurationDHMS. If a component contains a data series with linearly increasing real values, the keyword is followed by the starting value and increment for the row in curly brackets, e.g. FloatingPoint64{0 0.1}. The starting value and increment always have to be specified as numerical values, meaning that calendar time values and durations are not allowed. In this case, the corresponding data columns may

Row	Contents
	remain empty, since they are ignored during import. The formatters correspond to the entries on the Formatting tab of the FlexPro Properties dialog box of the data sets. These are output as strings and must be set off by quotation marks. Quotation marks appear as doubled, since the complete data structure identifier is also output as one large string. Data types and formatters for X and Z components are only output if they exist in the corresponding data structure.
4	Comments for the individual components of the data sets and formulas. Output occurs in the order X, Y, Z, with the X and Z component only shown if they are present. The Z component is only output here for space curves and formulas. For signal series with Z components, this is not a column, but a row and its comment is output separately.
5	The hyperlinks assigned to the data sets and formulas.
6	Details on the individual components of data sets and formulas. For the X and Z component, the specifications for Row 4 apply.
7	The description of the physical quantity for the individual components of the data sets and formulas. For the X and Z component, the specifications for Row 4 apply.
8	The lower limit ranges for the individual components of data sets and formulas. For the X and Z component, the specifications for Row 4 apply. The values are formatted as specified under 28.
9	The upper limit ranges for the individual components of data sets and formulas. For the X and Z component, the specifications for Row 4 apply. The values are formatted as specified under 28.
10	Absolute times for the individual components of the data sets and formulas. For the X and Z component, the specifications for Row 4 apply. The values are formatted as specified under 28.
11	Curve colors of data sets and formulas as a 6-digit hexadecimal number of the RGB value in the form of BBGGRR or #NV for "Automatic".
12	Creation date and time of the data sets and formulas are in the format MM/DD/YYYY HH:MM:SS. Output is in local time.
13	Date and time of the last modification to data sets and formulas.
14	Author of the data sets and formulas.
15	Origin of the data sets and formulas.
16	Component selection: None, Y, X, Z

Row	Contents
	Specifies whether a data set contains all data or is to be interpreted as the component of a composite data structure.
17	Name of the assigned X data set in case a data set is marked as a Y component.
18	Name of the assigned Z data set in case a data set is marked as a Y component.
19	A list of parameters for each data set and each formula. For each parameter, an expression in the format `<data type> <name>{<value>}[<unit>]` is output. The data type is coded as specified under 3. The values are formatted as specified under 28. For complex values, however, real and imaginary components are separated with a space, and strings are always set off by quotation marks. These are then doubled, since the whole list is seen as one large string. The unit, if present, is placed in brackets following the value when output.
20	The comments of the Z components of all signal series with a Z component. See also Row 4.
21	The units of the Z components of all signal series with a Z component. See also Row 4.
22	The names of the physical quantities of the Z components of all signal series with a Z component. See also Row 4.
23	The lower range limits of the Z components of all signal series with a Z component. See also Row 4.
24	The upper range limits of the Z components of all signal series with a Z component. See also Row 4.
25	The absolute times of the Z components of all signal series with a Z component and formulas. See also Row 4.
26	The FPScript codes of formulas.
27	The data of the Z components of all signal series with a Z component. The values are formatted as specified under 28.
28...	The data of the data sets and formulas. For complex data, the real part is output first and the imaginary part is output to a second column. For signals and signal series, output occurs in the order X, Y. For space curves, the order is X, Y, Z. Signal series with a two-dimensional X component are output in the order X0, Y0, X1, Y1 ... Xn-1, Yn-1. Void values are output as #NV. Floating point values are output with a decimal point. For Boolean values, TRUE or FALSE is output. Calendar time values are shown in local time as

Row	Contents
	MM/DD/YYYY HH:MM:SS. If applicable, seconds are shown with fractional digits.

All rows are always output, even if they contain nothing but column separators.

The number of columns of the data table not only depends on the number of data sets to be exported, but also on their data structure and data type.

When strings are stored, they are placed in quotation marks if they contain at least one quotation mark or any control character. If a string is placed in quotation marks, all quotation marks contained within are doubled. Control codes within the string are not changed. This means that carriage returns included in a string must not be interpreted as a new row of the FlexPro data table.

The current version of the format is 5.0. The following table shows the version change history:

Version	Changes
1.0	Initial version.
2.0	The data types `Integer64` and `ComplexInteger64` were added.
3.0	The rows 7, 8, 9, 22, 23 and 24 specified above were added in which the data range limits and names of physical quantity of the individual components are stored. For parameters with units, they are now appended in brackets.
4.0	The rows 10 and 25 specified above were added in which the absolute times of the individual components are stored.
5.0	The rows 11 and specified above were added in which the curve colors of the data sets and formulas are stored.

Exporting Data via OLE

OLE (Object Linking and Embedding) is a standard protocol for data exchange in Windows. To export data via OLE, the application that is to receive the data must have an OLE client interface. FlexPro can provide the data in CSV and text formats.

Exporting Data via the Clipboard or via Drag-and-Drop

You can export individual data values or ranges from a data set into other applications using the clipboard or via drag-and-drop. FlexPro provides the data in CSV and text formats.

Exporting Data

Exporting Data Objects as a File

1. Select all of the objects in the object list that you want to export to the file. Select a folder in order to export all data objects contained within it (not supported by all formats).
2. Right-click with your mouse to open the context menu.
3. Click Data[Export] > Export.
4. Select the target folder and specify the file name for the file.
5. Under File Type select the format for the file.
6. Select any export options for the file.

Exporting Data Objects via OLE

The way the link is inserted depends on the external application into which the data is to be inserted. The process described here is therefore an example of exporting to a Microsoft Excel spreadsheet.

1. Select a data set, a formula or a related object in the object list.
2. Use the command Home[Clipboard] > Copy to copy the OLE object to the clipboard.
3. In Excel you open the table into which you want to insert a link to FlexPro files.
4. Select the cell that corresponds to the upper left corner of the column or field to be inserted.
5. Open the list box under Home[Clipboard] > Paste window and select Paste Special.
6. In the Paste Special dialog box, select the format CSV and select the option Link.

Exporting Data via the Clipboard or via Drag-and-Drop

Inserting the values depends on the external application into which the data is to be inserted. The process described here is therefore an example of exporting to a Microsoft Excel spreadsheet.

1. Open the data set from which you want to export the values.
2. In the data set window or in an open folder's data view, select the values that you want to export.

3. Use the command Home[Clipboard] > Copy to copy the selection to the clipboard.
4. Use Excel to open the spreadsheet into which you want to insert the values.
5. Select the cell that corresponds to the upper left corner of the column or field to be inserted.
6. Use the command Home[Clipboard] > Paste.

or

1. Open the data set from which you want to export the values.
2. Select the values that you want to export from the data set window.
3. Move the mouse cursor to the edge of the selected area. The cell cursor turns into an arrow cursor. Press the left mouse button and drag the mouse to the Excel worksheet into which you would like to insert the values. The insertion area is shown by a rectangle.
4. Release the left mouse button after you have selected the insertion point. The values are copied from FlexPro into the selected area.

4 Analyzing Data with Cursors

FlexPro allows you to measure curves in 2D and 3D diagrams and videos using cursors.

More specifically, the following options are available:

- Show one or two cursors using different display formats.
- Free and bound cursors.
- Harmonic cursors for measuring secondary harmonics in spectra.
- Sideband cursors for measuring sidebands in spectra.
- Order tracking cursor for measuring orders in speed-frequency spectra and frequencies in order spectra of machine ramp-ups.
- Slope cursor for measuring the slope of the curve.
- Show values under the cursors, their differences and custom coordinates.
- Zooming and scrolling of the curves.
- Move to individual frames in videos.
- Animate a cursor to play back curves.
- Translate curves in 2D diagrams.
- Navigate to peak values.
- Select individual points or ranges of one or more curves.
- Edit the values under the cursors.
- Remove outliers from curves.
- Form new data sets from sections of curves.
- Dimension curves.

You can use the cursors in 2D and 3D diagrams 381, in media 464, in documents 470 and in worksheets 493. If you use the cursors in a worksheet or document, you can simultaneously measure multiple diagrams and analyze videos that you have recorded synchronously with physical quantities. However, the worksheet is generally better suited than the document as an environment for using the cursors, because it takes better advantage of the screen. While the cursors in the worksheet are always active, you first have to activate them in the diagram, media or document.

Cursor Types

You can use one or two cursors for each diagram. The leading cursor is displayed as a solid line and is used to measure curves. The origin cursor is shown as a dashed line. It specifies the origin for the calculating differences.

FlexPro offers you free and bound cursors. You can move a free cursor anywhere on the plane of the diagram, while a linked cursor always follows the points of the curve. Free cursors are particularly suitable for measuring parts of the curve where there are no points or for measuring distances to a free reference position. You individual curves of a diagram from cursoring. The cursor can then no longer be positioned on the curves, and their values are not displayed in the coordinates window.

You can display the cursors as a vertical line or as a cross-hair. You can also move the cross-hair cursor vertically with your mouse.

Media objects support only a cursor that represents the time of the currently displayed frame. Use the horizontal scrollbar so you can move to an image and thus move the cursor.

In addition to the main cursor, you can activate one of the following auxiliary cursors:

Harmonic Cursors

FlexPro offers you harmonic cursors that are specifically designed for measuring secondary harmonics in relation to a primary harmonic in spectral signals. If you activate this option, in addition to the active cursor, additional cursors are displayed, which are positioned at multiples of the current X cursor position. You can specify the spacing between harmonic cursors in octaves, where one octave corresponds to a factor of two.

Sideband Cursors

These are also suitable for evaluating spectra. The sideband cursors appear at a fixed distance to the left and right of the current X cursor position. You can easily adjust the spacing of the sideband cursors with the mouse by moving a sideband cursor.

Order Tracking Cursor

Speed-frequency spectra or order spectra are often used to evaluate machine ramp-ups. The speed determines the basic frequency of the machine. Since parts of the machine usually rotate at multiples of the speed, amplitudes also appear in the spectrum at these multiples, which are called orders. The order tracking cursor appears in the speed-frequency spectrum as a diagonal line and marks locations of the same order. In order spectra, it appears as a hyperbola and marks locations of the same frequency.

Slope Cursor

The slope cursor appears as a tangent to the curve and is used to measure the slope of the curve.

Synchronizing Cursors

If you use the cursors in a worksheet or document with multiple diagrams and/or videos, FlexPro can synchronize the cursors between the individual objects. Synchronization can be carried out via the X index, i.e. via the number of the active value in the curve or frame in the video. Alternatively, FlexPro can synchronize the cursors via the X value. In a video, this is the time corresponding to the displayed frame; in a curve, this is the X value of the active point.

FlexPro can also synchronize the cursors while you play a video or audio signal. In this case, the cursors in the diagrams are animated and move in sync with the video or audio signal over the curves. Conversely, you can move a cursor in a diagram to display the relevant frame in the video, e.g. for a position at which you identified a point of interest in a measurement signal.

Zooming and Scrolling

You can zoom into any section of the diagram's plane or show the area between the two cursors as spread out in order to take a closer look at interesting signal segments. All processes that change the section displayed in the diagram achieve this by manipulating the axis scale end values in the diagram. In addition, automatic scaling on the axis concerned is disabled. This modification takes place directly on the axis in the diagram concerned. If you edit a linked diagram with cursors, zooming therefore affects all documents where you have inserted a link to the diagram. You can undo any step in the process of zooming and scrolling or restore the diagram to its original state. If you move the cursor on a curve you have zoomed into, it might leave the visible section. In this case, you can determine whether the image section should be automatically expanded or moved.

Animating a Cursor

You can animate any cursor, which means you can let it run automatically over a curve. If the curve under the cursor is a time signal, the animation takes place in real-time or at a multiple rate of this speed. When viewing an image you have zoomed into, the section of the image may be automatically expanded or moved when the cursor leaves it. The signal then scrolls across the screen.

Markers

When measuring curves, you can select individual points or ranges, for instance, to highlight prominent points in the curve. When setting a marker, you can determine whether the marker is to be displayed in the current view of the diagram or in all views that display the diagram. FlexPro offers you a variety of templates for markers that you can use to mark not only data points, but also slopes, waypoints, orders and frequencies. Of the marker has a text box, it can be placed either on the edge of the diagram or directly on the marked point. In the text box you can display any text in which you can embed the coordinates of the selected points or, for example, the slope using fields.

Use a multi-marker to select all curves at a specific X position in one step. If you place the marker field of the multi-marker at the diagram edge, the coordinates of all curves are combined in one field.

You can mark a curve range, the area under a curve range, or the rectangular range of a surface with a range marker. This highlights a section of a curve or surface in color.

An image marker places an image at the highlighted position. If you have recorded videos synchronously with measurement data, FlexPro can automatically take a still image for the highlighted X position from an assigned media 464 and use it for the image marker. Alternatively, you can specify an image file from which the image is to be imported. If you use a chart with map display and add an image as an image marker in which the longitude and latitude of the recording are stored, then you can decide whether the image marker should be positioned at this location instead of the current cursor position.

FlexPro offers a variety of marker templates to choose from with attributes that are stored in the FlexPro user profile. Each of the above marker types is also available as a custom marker that you can design freely. The attributes of the custom markers are stored in the diagram where you use the markers.

However, after you have set a marker, you can always edit or reposition it. FlexPro takes design attributes that you set to Automatic from the user-defined template in the diagram. The attributes of the template thus apply to all marks for which you do not make individual settings.

You can store the marker positions as a data set or formula in the project database, which you can access in your analyses.

Changing Data Set Values

If the curve where the cursor is located obtains its data from a data set, then you can manipulate its values directly using the cursors. The cursors offer you the option of changing a single value or the range between the cursors to a different value, interpolating them or setting them to void. The first option is ideal for the visual adjustment of a calibration characteristic curve, for instance, and the two others are good for removing outliers from the data set. Setting outliers to void is generally preferred over interpolation, since in this case no false data is being introduced. However, the advantage of interpolation is that the resulting data cannot contain void values, and thus calculations can still be made using algorithms, such as the Fourier transform.

Translating Curves

You can translate the curves of 2D diagrams using the cursor, e.g. to compensate for time lag when measuring. In this case, the curves are only visually translated. The base data remains unchanged. The offset is entered as Y offset and X offset on the Data tab of the curve.

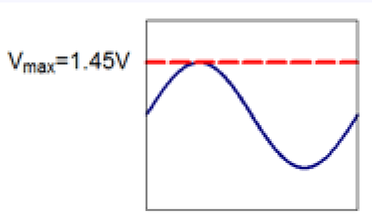
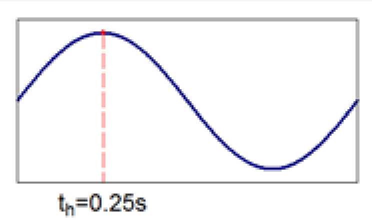
Coordinates Window

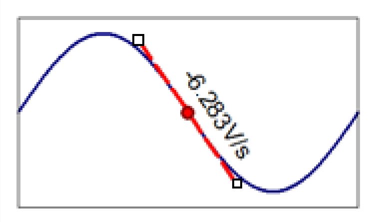
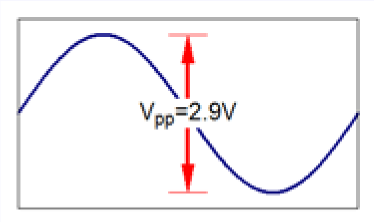
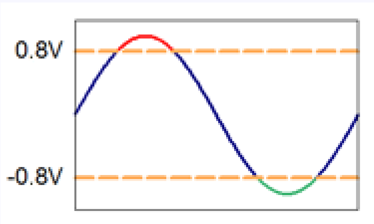
In the coordinates window, the coordinates of the cursors and their differences are displayed. To do this, you can specify whether only the coordinates of the active diagram or the coordinates of all diagrams in a document or worksheet are to be displayed. You can copy the contents of the coordinates window to the clipboard as a static picture or into a diagram, document or worksheet as a dynamic object. In addition to the coordinates window, FlexPro displays an information window with the scaled coordinates of the current mouse pointer position if you move the pointer over the diagram.

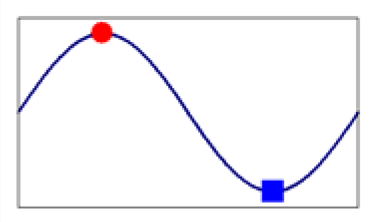
You have two options available for customizing the coordinates window. Instead of displaying the predefined coordinates, you can display any presentation object in the coordinates window, such as a table with custom results related to the cursors. FlexPro updates this object automatically whenever cursors are moved. The second option is to extend the coordinates window to include custom coordinates. These are special FPScript formulas that access the cursors and perform particular calculations, such as finding the area under the curve between the two cursors. FlexPro provides you with several predefined coordinates. However, you can also program your own coordinates in FPScript.

Dimensioning curves

FlexPro offers you a variety of dimensioning elements that you can use to dimension the curves of diagrams:

Dimensioning	Description
 Boundary Line	A horizontal or vertical boundary line.
 Perpendicular Line	A perpendicular line from a particular point on the curve to one of the edges of the diagram.

Dimensioning	Description
 Tangent	A tangent at a point on the curve.
 Dimension Line	A dimensioned distance in the Y direction, X direction or in any direction.
 Tolerance Band	A horizontal or vertical tolerance band. A color change can be enabled for the curve when the tolerance is overshot or undershot.

Dimensioning	Description
 Symbol	A symbol that highlights a particular point on the curve.

Dimensioning is added as special markers in the diagram so that they are always correctly scaled with the diagram when subsequently zooming into the curve. Like other markers, you can reposition or design dimensions afterwards.

The current cursor positions are used as the initial points for the coordinates used for new dimensioning. These do not necessarily have to be on a curve. You can work with free cursors.

Accessing Cursors and Markers with FPScript

You can access the current cursor positions and markers with FPScript 246 and thus use these within analyses. For instance, using both cursors you can select a range for which an analysis is to be calculated, or you can select events in a signal using markers that you can then access in FPScript formulas. To do this, the diagram, document and the worksheet provide you with Properties 279 with which you can access the cursors and markers. For example, the following FPScript formula accesses the current cursor positions of the second pane of a worksheet in order to form a signal segment:

```
Signal[Worksheet.WKS.Cursor(fpCursorMinX, 1).PositionIndex, _
    Worksheet.WKS.Cursor(fpCursorMaxX, 1).PositionIndex]
```

4.1 Activating and Setting Up Cursors

Toggling Cursors in a Document or Diagram

- Use the command Design[Cursors] > On/Off or Cursors[Cursors] > On/Off.

Note You cannot disable cursors in a worksheet.

Toggling Auxiliary Cursors On and Off

- Click the arrow next to the button Cursors[Cursors] > Toggle Cursors and select or deselect one of the options in the lower section of the menu.

Changing the Number and Spacing of Auxiliary Cursors

1. Right-click with your mouse on the diagram, worksheet or document to change its appearance.
2. From the context menu that appears, select the Properties command to open the Properties dialog box 107.
3. Select the Cursor Appearance tab.
4. Under Auxiliary Cursors specify the desired number and spacing of the cursors in octaves.

Toggling Coordinates Displayed at Cursor On and Off

- Select or deselect the option Cursors[Cursors] > Show Coordinates.
- To adjust the position of the coordinates, click the arrow next to the button Cursors[Cursors] > Show Coordinates and select one of the options in the menu.

Excluding a Curve from Cursoring

When cursors are enabled:

- Open the Cursor [Curves] > Cursor menu and deselect the curves you want to exclude.

With cursors disabled:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on the curve that you want included when cursoring in order to select it.
3. Right-click with your mouse to open the context menu.
4. Uncheck the option Account for When Using Cursors.

Customizing the Display of Cursors

To change the cursor type:

- Click the arrow next to the button Cursors[Cursors] > Toggle Cursors and select the desired cursor type from the menu.

Note You can quickly switch between crosshairs and line cursor by clicking the button under Cursors[Cursors] > Toggle Cursors.

To change the line attributes and color:

1. Click the box in the lower right corner of Cursors[Cursors] to open the Properties dialog box.
2. Select the Cursor Appearance tab and make the desired settings.
3. If you use the cursors in a worksheet or document and want to apply your settings to all diagrams, select the option Apply to all diagrams before clicking on Apply.

4.2 Moving the Cursors

Moving the Cursors Using the Keyboard

Use the following keys to move the cursors:

Navigation Keys	Function
RIGHT	Moves the cursor in the positive X direction.
LEFT	Moves the cursor in the negative X direction.
UP	Moves a bound cursor to the previous curve and a free cursor in the positive Y direction.
DOWN	Moves a bound cursor to the next curve and a free cursor in the negative Y direction.
PAGE UP	Moves a bound cursor to the previous curve and a free cursor in the negative Z direction.
PAGE DOWN	Moves a bound cursor to the next curve and a free cursor in the positive Z direction.

You can also press the following keys to affect the behavior:

Additional Key(s)	Function
CTRL+	Moves by an average increment.
SHIFT+	Moves by a large increment.
CTRL+SHIFT+	Jumps to the edge.
ALT+	Moves both cursors.
SHIFT+ALT+	Moves to event.

If you want swap both cursors when swapping the curve without having to press the ALT key, select this option first: Cursors[Navigation] > Move Both Cursors.

Select the option Cursors[Cursors] > Synchronize Cursors if the diagram is located in a worksheet or a document and you want to synchronize the movement of the cursors of the other diagrams as well.

The following table shows additional key assignments:

Key	Function
TAB	Switches between origin cursor and leading cursor.
F6	Switches to the next diagram in the document or worksheet.
SHIFT+F6	Switches to the previous diagram in the document or worksheet.
CTRL+W	Moves the cursor to the next event.
CTRL+Q	Moves the cursor to the previous event.

Note You can easily view the key assignment for cursor navigation using the buttons under Cursors[Navigation]. Pressing the SHIFT key and/or the CTRL key changes the appearance of these buttons. Now point with your mouse to one of the buttons in order to view a popup window showing the key assignment.

Moving the Cursors with the Mouse

You can basically move free and bound cursors the same way. Moving bound cursors, however, is limited, since they must always follow the curve.

Moving	Procedure
Placing a cursor on a different curve	In the left-hand column of the coordinates window, click on a curve to set the leading cursor on it. Clicking in the right-hand column moves the origin cursor accordingly.
Activating the leading or origin cursor	Click with your mouse in the background of the relevant coordinates window column. The left-hand column activates the leading cursor and the right-hand column activates the origin cursor.
Moving a cross-hair cursor or line cursor	Use your mouse to drag the vertical line of the cursor to the desired position.
Moving both cross-hair cursors or line cursors horizontally	Use your mouse while holding down the ALT key to drag the vertical line of a cursor to the desired position.
Moving a cross-hair cursor vertically	Use your mouse to drag the horizontal line of the cursor to the desired position.
Moving both cross-hair cursors vertically	Use your mouse while holding down the ALT key to drag the horizontal line of a cursor to the desired position.
Freely moving a cross-hair cursor	Use your mouse to drag the position where the two lines of the cursor intersect to the desired position.
Freely moving both two cross-hair cursors or line cursors	While holding down the ALT key, drag the rectangle surrounded by the cross-hairs or the vertical band bordered by the line cursor in the desired direction.
Set a cursor on a particular point on the curve.	Click with your mouse on the desired point. The mouse cursor must appear as an arrow. If not, such as because the point is under a cursor line, press and hold down the SHIFT key.

Note If you want to set a bound cursor on a particular point on the curve, just click on the location. FlexPro will then have to search for the point on the curve next to the point you clicked on. You can choose between the following options: To nearest point and Use X position. When the first option is selected, FlexPro searches the entire curve to find the point with the shortest distance from the position you selected. In this mode it is very easy to select a peak value, for instance. In the case of large data sets, however, this method is too processor-intensive, so you should select the other option instead. Then only the horizontal component of the selected position is taken into account and directly from that the index of the point to be selected is calculated. This method is very fast because FlexPro doesn’t have to search through the data set, but it only works for equidistantly sampled 126 signals. In the Properties dialog box, select the options on the Cursor tab of the diagram, worksheet or document with the cursor you are using.

Select the option Cursors[Cursors] > Synchronize Cursors if the cursors of the other diagram should be moved synchronously in a document or worksheet.

Navigating to Peak Values or Markers

You can use the commands Cursors[Event] > Previous and Cursors[Event] > Next to move the active cursor from peak value to peak value or from marker to marker. On the Cursors tab of the diagram's Properties dialog box 107 you can specify whether you want to go to local maxima, local minima, minima and maxima or markers.

Here you can also enter a hysteresis, which determines what should or should not be recognized as a peak value. If some are skipped when navigating the peak values, you need to decrease the hysteresis. If small waves or noise in the course of the signal are recognized as peak values, then you need to increase the hysteresis.

Navigating to a Specific Point in a Diagram

Use the Cursors[Event] > Go To command to place the cursor specifically on a point of the curve or a point in the diagram plane. Use the options under Cursor[Zoom Curve] > Adjust Image Section to determine whether and how the image section should be adjusted when necessary.

You can also navigate to a point using your mouse by simply clicking on the point.

Navigating to a Frame in a Video

Use the Cursors[Event] > Go To command to navigate to a frame in a video. Alternatively, you can use the horizontal scrollbar.

Scrolling Along a Curve

If you display the curve spread out in the diagram as follows and then animate the cursor, you will obtain a scrolling effect:

1. Hold down the CTRL key and use your mouse to drag a vertical band from left to right in order to display a section of the curve as spread out.
2. Open the list box under Cursors[Zoom Curve] > Adjust Image Section and select Fixed Cursor Position.
3. Uncheck the option Cursors[Cursors] > Cross-Hair Cursor to display the cursor as a line cursor.
4. Now click Cursors[Play] > Play or Cursors[Play] > Reverse to allow the cursor move to the right or left.
5. Click Cursors[Play] > Stop to stop scrolling.

Notes

- The option in the Fixed Cursor Position list box Cursors[Zoom Curve] > Adjust Image Section causes the cursor to appear at a fixed location and the curve to move under the cursor. Alternatively, you can use the Move option.
- You can change the speed of the cursor in the Cursors[Play] > Playback Speed field.
- You can set the cursoring increment in the Playback increment field of the Properties dialog box.

Synchronizing Cursors

To synchronize the cursors of diagrams in a worksheet or document:

1. Click on the diagram with the cursors that are to be moved at the same time as when the cursors are moved in a different diagram.
2. Select the option Cursors[Cursors] > Synchronize Cursors.

4.3 Changing the Image Section

Stretching the Image Section

1. Place the two cursors on the left and right side of the section that you want to display as stretched.
2. Choose the command Cursors[Zoom Curve] > Spread. You can run the command from the keyboard by using the PLUS key (+) on the numeric keypad.

Note Spreading takes place in such a way that both cursors end up on the left and the right side of the diagram.

Synchronizing Zoom Actions

1. Right-click on the diagram that is to be zoomed synchronously when you zoom another diagram and select Properties.
2. In the Properties window, on the Cursor Settings tab, check Synchronize Zooming.
3. If zoom operations are to be synchronized in the Y direction as well, also select the option Also sync. zoom in Y direction.

Stretching and Compressing the Curve

- Use the commands in the selection fields Cursor[Zoom curve] > Stretch and Cursor[Zoom curve] > Compress to stretch or compress the curve in the desired direction.

Note For the axis concerned, automatic scaling is turned off and the axis end values are reduced or increased by 10 % of the range of values displayed.

Zooming with the Mouse

To display any rectangular section as a close-up:

- Use your mouse to drag a rectangle from the top left to the bottom right of the desired section to enclose it.

To zoom in or out on any location:

1. Use the mouse point to the position where you want to zoom in or out.
2. Hold down the CTRL key and then turn the mouse wheel forward or backward.

To enlarge the graphic in the X direction:

- Press and hold the CTRL key and use your mouse to drag a vertical band from left to right.

To enlarge the graphic in the Y direction:

- Press and hold the SHIFT key and use your mouse to drag a band from top to bottom.

Scrolling

- If the active curve is not completely visible, you can scroll through the section of the image displayed in the diagram. To do this, you can use the scrollbars displayed at the bottom and to the right of the worksheet or the active diagram in the document.
- Use the buttons under Cursors[Move Image Section] to scroll at a fixed increment.
- You can use the mouse wheel to scroll instead of the scrollbars. You can scroll vertically by holding down the SHIFT key while using the mouse wheel. Otherwise, scrolling takes place horizontally.

Adapting the Image Section to the Curve

- Click Cursors[Zoom Curve] > Complete to display the entire curve, or choose Full Height or Full Width from the list box.

Note: When using the command for the first time, the corresponding axis of the diagram is switched to autoscaling via the data range lower limit/upper limit. If the data range limits are specified in the data set, they are then used as the default for autoscaling the axis. If the command is executed again, it switches to autoscaling of the axis via the data minimum/maximum.

Adjusting the Image Section when Moving Cursors

Use the options in the Cursors[Zoom Curve] > Adjust Image Section list box to determine how the image section should be adjusted when you move a cursor:

- Do Not Adjust leaves the segment unchanged. Use this option if you do not want to change the configured axis scaling.
- Fixed Cursor Position adjusts the image section so that the cursor appears in a fixed location. For a cross-hair cursor the adjustment is in the X and Y direction, and for a line cursor it is only in the X direction.
- Expand automatically expands the image section when the cursor leaves.
- Move automatically moves the image section when the cursor leaves.

Undoing Zoom Actions

- You can undo all actions in increments that change the section displayed in the diagram using the command Cursors[Zoom Curve] > Previous.
- Use the command Cursors[Zoom Curve] > Restore Image Section to restore the axes of the diagram to their original state.

Note You can run the command from the keyboard by using the MINUS key (-) on the numeric keypad.

Automatic Zooming of Segments

You can zoom closer to the area between the cursors of a diagram in a second diagram. Follow these steps:

1. Select the diagram in which you want get a close-up of a segment.
2. Select the option Cursors[Zoom Curve] > Automatic to enable automatic zooming for this diagram.
3. Now switch to a different diagram and move the cursors.

Note If you create a worksheet with two areas for a data set and select the arrangement marked with a magnifying glass in the menu, FlexPro inserts two diagrams for the same data set into the worksheet, with the second diagram representing a section enlargement of the first.

4.4 Working with Markers

Setting and Deleting Markers

To set a marker:

1. Place the active cursor at the location that you want to mark.
2. Open the menu Cursors[Marker] > Place.
3. Check the option Set Markers in All Views if the marker is to appear in all views of the diagram.
4. In the menu Cursors[Marker] > Place click on a marker in order to set it. Set other markers of the same type by clicking on Cursors[Marker] > Place or pressing the ENTER key.

Note In the menu Cursors[Marker] > Place you will find predefined and custom markers. You can specify the appearance of the custom markers on the Markers tab of the object by setting the markers. These attributes are not used for the predefined markers. However, after you have set a marker, you can always edit it with the command Cursors[Marker] > Edit.

To set a range marker:

1. Set the two cursors to the right and left of the range of the curve you want to mark.
2. Open the menu Cursors[Marker] > Place.
3. Check the option Set Markers in All Views if the marker is to appear in all views of the diagram.
4. In the menu Cursors[Marker] > Place click on Curve Range, Area Under Curve or Custom Range Marker. Set other markers of the same type by clicking on Cursors[Marker] > Place or pressing the ENTER key.

Note You can also set a range marker with the mouse by holding down the CTRL and SHIFT keys and then dragging a range with the mouse.

To set an image marker:

1. Place the active cursor at the location that you want to mark.
2. Open the menu Cursors[Marker] > Place.
3. Check the option Set Markers in All Views if the marker is to appear in all views of the diagram.
4. In the menu Cursors[Marker] > Place click on Image Marker or Custom Image Marker. Set other markers of the same type by clicking on Cursors[Marker] > Place or pressing the ENTER key.
5. If you specified a media on the diagram's Cursors tab, a still image will be used from that media. Otherwise, a dialog box appears where you can select an image file.

To delete a marker:

- Delete the marker closest to the active cursor using the command Cursors[Marker] > Delete.
- You can delete all of a curve's markers by placing a cursor on the curve and then in the Cursors[Marker] > Delete list box choose the Remove All for Cursor command.
- You can delete all unbound markers by disabling the option Cursors[Cursors] > Bind Cursor to Curve and then using the command Cursors[Marker] > Delete > Remove All For Cursor.
- You can delete all markers of the diagram by using the command Cursors[Marker] > Delete > Delete All.
- You can delete any marker while the cursors are active by right-clicking with your mouse to select Delete Marker from the context menu.
- If you disable the cursors by clicking on Cursors[Cursors] > On/Off, you can select one or more markers and delete them using the DEL key. If the diagram is in a document or worksheet, you will first have to open it by double-clicking on it.

Positioning Markers

To adjust a marker label (marker field):

1. Use your mouse to point to the marker label that you would like to position and keep the mouse key pressed.
2. Now drag the marker label to the desired position.

To change the positioning of a marker label:

1. Double-click on the marker label that you want to position.
2. In the Edit Marker dialog box, use the Marker field list box to select where to position the marker label.

To change the position on the curve of a marker:

a) When cursors are enabled:

1. Point the mouse at the place where the marker sits on the curve and keep the mouse button pressed.
2. Now drag the marker to the desired position.

or

1. Place the cursor near the marker you want to reposition. The marker closest to the cursor is displayed with a marker in each case.
2. Click Cursors[Marker] > Position and enter the desired index or value in the dialog box.

b) When cursors are disabled:

1. Select the marker whose position you want to change.
2. Now change one of the values in the Properties window X position or X index to move the marker in the Y direction. In the 3D diagram, you can also move the marker in the Z direction by changing one of these values: Z position or Z index.

Note When you position a marker for display in all diagram views, then it appears in all views at the new position.

Assigning Media as a Source for Image Markers

When cursors are enabled:

1. Click on Cursors[Marker] > Markers Properties.
2. On the Markers tab of the Properties dialog box that appears, select Custom image marker under Template.
3. Under Marker field select Media from the Source field.
4. Now in the Media field enter the media whose frames should serve as the source for the image markers in the diagram.

In editing mode:

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select Properties.
3. On the Markers tab of the Properties dialog box that appears, select Custom image marker under Template.
5. Under Marker field select Media from the Source field.
6. Now in the Media field enter the media whose frames should serve as the source for the image markers in the diagram.

Editing Marker Text

To edit the default text to be displayed in all markers to which you did not assign an individual text:

When cursors are enabled:

1. Right-click with your mouse on a marker and select Marker Properties.
2. Enter the desired text on the Markers tab of the Properties dialog box that appears. You can embed fields for the index, X, Y and Z values of the marker, and for the corresponding units and data set names.

In editing mode:

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select Properties.

3. Enter the desired text on the Markers tab of the Properties dialog box that appears. You can embed fields for the index, X, Y and Z values of the marker, and for the corresponding units and data set names.

To assign individual text to a marker:

When cursors are enabled:

1. Right-click with your mouse on the marker and select Edit Marker.
2. In the Edit Marker dialog box, you can assign text to a marker or edit existing text.

In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a marker.
3. In the Edit Marker dialog box, you can assign text to a marker or edit existing text.

To reapply the default text to a marker that displays unique text:

When cursors are enabled:

1. Right-click with your mouse on the marker and select Edit Marker.
2. Delete the existing text in the Edit Marker dialog box.

In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a marker.
3. Delete the existing text in the Edit Marker dialog box.

Note You can also assign text to a range marker, but this will not be displayed in the curve.

Changing Marker Size

To change the size of all markers to which you have not assigned any individual size attribute:

When cursors are enabled:

1. Right-click with your mouse on a marker and select Marker Properties.
2. On the Markers tab of the Properties dialog box that appears, enter the desired Width and Height. For text markers, these are the minimum values to be exceeded if the text is larger.

In editing mode:

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select Properties.
3. On the Markers tab of the Properties dialog box that appears, enter the desired Width and Height. For text markers, these are the minimum values to be exceeded if the text is larger.

To change the size of a single marker:

When cursors are enabled:

- Click on an edge of the marker and drag it while holding down the mouse button.

or

1. Right-click with your mouse on the marker and select Edit Marker.
2. Enter the desired Width and Height in the Edit Marker dialog box. For a text marker, these are the minimum values to be exceeded if the text is larger.

In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a marker to select it.
3. Drag the rectangle of the marker field with the mouse to the desired size.

To reassign the default size to a marker with an adjusted size:

When cursors are enabled:

1. Right-click with your mouse on the marker and select Edit Marker.

2. In the Edit Marker dialog box, enter 0 in the Width and/or Height field.

In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a marker.
3. In the Edit Marker dialog box, enter 0 in the Width and/or Height field.

Note You can also assign text to a range marker, but this will not be displayed in the curve.

Changing Marker Formatting

To change the formatting for all markers to which you have not assigned any individual text:

When cursors are enabled:

1. Right-click with your mouse on a marker and select Marker Properties.
2. On the Markers tab of the Properties dialog box that appears, set the insertion point in the Text input field on the placeholder of the value whose formatting you want to change, such as %<YValue> for the Y value.
3. Now click on Edit Formatter and set the formatting desired.

In editing mode:

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select Properties.
3. On the Markers tab of the Properties dialog box that appears, set the insertion point in the Text input field on the placeholder of the value whose formatting you want to change, such as %<YValue> for the Y value.
4. Now click on Edit Formatter and set the formatting desired.

To change the formatting for a single marker:

When cursors are enabled:

1. Right-click with your mouse on the marker and select Edit Marker.

2. In the Edit Marker dialog box that appears, set the insertion point in the Text input field on the placeholder of the value whose formatting you want to change, such as %<YValue> for the Y value.
3. Now click on Edit Formatter and set the formatting desired.

In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a marker.
3. In the Edit Marker dialog box that appears, set the insertion point in the Text input field on the placeholder of the value whose formatting you want to change, such as %<YValue> for the Y value.
4. Now click on Edit Formatter and set the formatting desired.

Customizing the Display of Markers

To change the display attributes of all markers to which you have not assigned any individual display attributes:

When cursors are enabled:

1. Right-click with your mouse on a marker and select Marker Properties.
2. On the Markers tab of the Properties dialog box that appears, set the attributes for the desired marker type.

In editing mode:

1. If the diagram is in a document, select the diagram.
2. Now right-click with your mouse on the diagram and select Properties.
3. On the Markers tab of the Properties dialog box that appears, set the attributes for the desired marker type.

Note You can automatically color the markers in the color of their respective curve by selecting Automatic as the background color.

To assign individual display attributes to a marker:

When cursors are enabled:

1. Right-click with your mouse on the marker and select Edit Marker.
2. Set the desired attributes in the Edit Marker dialog box.

In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a marker.
3. Set the desired attributes in the Edit Marker dialog box.

To restore default attributes assigned to a marker that uses individual display attributes:

When cursors are enabled:

1. Right-click with your mouse on the marker and select Edit Marker.
2. In the Edit Marker dialog box, select Automatic for the corresponding attributes or delete the text.

In editing mode:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a marker.
3. In the Edit Marker dialog box, select Automatic for the corresponding attributes or delete the text.

Copying Markers

A data set with the markers is stored in the folder in the object list. This folder contains the object in which you use the cursors.

To copy the markers of a specific curve or all curves:

1. If applicable, toggle the cursors using the command Design[Cursors] > On/Off.
2. Place the cursor on the curve whose markers you want to copy.
3. Select Cursors[Copy] > Markers.

4. In the dialog box that appears, select the option Copy markers of all curves, if you want to copy not only the markers of the curve under the cursor, but of all curves.
5. Select whether you want to save the markers in the order in which you had set them, or whether they should be sorted by value indices.
6. Now click on one of the options to save the markers in the desired format.

To copy all the markers of the graph that are not bound to any curve:

1. If applicable, toggle the cursors using the command Design[Cursors] > On/Off.
2. Unbind the cursor by unchecking Cursors[Cursors] > Bind Cursor to Curve.
3. Select Cursors[Copy] > Markers.

4.5 Working with Data

Editing Data

To change the Y value under the active cursor:

1. Set the cursor to the point on the curve whose Y value you want to mark.
2. Click Cursors[Edit] > Edit Value. You can also press the INSERT key instead.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

To increase or decrease the Y value under the active cursor:

1. Set the cursor to the point on the curve whose Y value you want to mark.
2. Press CTRL+'+' to increase the value or CTRL+'-' to decrease it.

To change the Y value under the active cursor using the mouse:

1. Hold down the CTRL and ALT keys and move the mouse pointer to the location where the vertical and horizontal cursor lines cross. The mouse pointer changes to a double arrow with an x in the middle.
2. Now hold down the left mouse button and drag the point under the cursor to the desired Y position.

To change the Y values of all data between the cursors:

1. Set both cursors on the left and right edge of the area whose Y values you want to change.
2. Click Cursors[Edit] > Edit Range.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

To move the Y or X values of all data between the cursors:

4. Set both cursors on the left and right edge of the area whose Y values you want to move.
5. Click Cursors[Edit] > Edit Range > Move Range in Y Direction or Cursors[Edit] > Edit Range > Move Range in X Direction.
6. In the dialog box that appears, enter a Y or X offset in the physical unit of the data upon which the curve is based.

Note You can only manipulate data directly if the curve is based on a data set and not on a formula. If the curve is based on a formula, you first have to convert it into a data set 131.

The manipulation takes place directly in the data set and cannot be undone.

Setting Data to Void

To make the Y value under the active cursor void:

1. Set the cursor to the point on the curve whose Y value you want to make void.
2. Open the selection field Cursor[Edit] > Edit Value and select the command Void Value.

To make the Y values of all data between the cursors void:

1. Set both cursors on the left and right edge of the area whose Y values you want to make void.
2. Open the selection field Cursor[Edit] > Edit Range and select the Void Range command.

Deleting Data

To delete the data point under the active cursor:

1. Set the cursor on the point on the curve that you want to delete.
2. Click on the arrow next to Cursors[Edit] > Edit Value and select the command Delete Data Point.

To remove all data points between the cursors:

1. Set both cursors on the left and right edge of the range with data points that you want to delete.
2. Click on the arrow next to Cursors[Edit] > Edit Range and select the command Delete Range.

Interpolating Data

To interpolate the Y value under the active cursor:

1. Set the cursor to the point on the curve whose Y value you want to interpolate.
2. Open the list box Cursor[Edit] > Edit Value and select the command Interpolate value.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

To interpolate the Y values of all data between the cursors:

1. Set both cursors on the left and right edge of the area whose Y values you want to interpolate.
2. Open the list box Cursor[Edit] > Edit Range and select the command Interpolate range.
3. In the dialog box that appears, enter a new Y value in the physical unit of the data upon which the curve is based.

Undoing Data Editing

- Open the list box Cursors[Edit] > Edit Value or Cursors[Edit] > Edit Range and choose the command Undo.

Note Undoing a data manipulation is only possible if in the Options dialog box of FlexPro on the Operation tab, the option Allow actions to data sets to be undone is selected.

Copying Data

To copy the Y values to the Clipboard that are currently displayed in the coordinates window for the active cursor as a data series:

- Click on Cursor[Copy] > Value(s).

To copy the section of the curve between the cursors to the object list:

1. Click on Cursor[Copy] > Range.
2. In the dialog box that appears, specify whether the current cursor positions are to be inserted as fixed or variable values. In the latter case, the range provides new data when you move the cursor. Thus, for example, you can combine the manual selection of a signal section with an automated analysis.

Note The formula or data record is stored in the object list in the folder containing the object in which you are using the cursors.

Displaying Data of a Curve

1. Set the cursor on the curve with the data you want to display.
2. Click Cursors[Edit] > Open Data Set.

4.6 Working with the Coordinates Window

Displaying and Hiding the Coordinates Window

- To display or hide the coordinates window, check the option Cursors[Cursors] > Show Coordinates Window or uncheck the option.

Changing Coordinates Window Display Options

1. Click the arrow to the right of Cursors[Cursors] > Show Coordinates Window.
2. In the menu that appears, you can set the desired options.

Changing the Coordinates Window Output Format

You can change the output format in the coordinates window for the data of one or more curves. The formatter that determines the output format is stored in the diagram on the General tab of the curve.

1. Right-click with your mouse on the value in the coordinates window for which you would like to change the output format.
2. Choose the command Change Output Format.
3. Set the desired formatting in the Formatter Properties dialog box and click on OK to close the dialog box.
4. If several curves are to be displayed in the coordinates window, the Assign Formatting dialog box appears. Here you can specify whether the new formatting is also to be used for the additional curves.

Opening a Data Set Displayed in the Coordinates Window

- Double-click in the coordinates window on the name of the curve based on the data set or right-click with the mouse on the name and select Open Data Set from the context menu.

Copying a Value from the Coordinates Window

To copy a value to the clipboard:

1. Use the right mouse button to click on the value in the coordinates window.
2. From the context menu, choose the Copy command to copy a value to the clipboard.

To copy a value using drag-and-drop:

1. Use the right mouse button to click on the value in the coordinates window.
2. In the context menu, select the Copy Coordinate command to copy the value as a formula to the object list.

To copy a value as a data set into the object list:

1. Use the right mouse button to click on the value in the coordinates window.
2. From the context menu, choose the Copy command to copy a value to the clipboard.
3. Right-click with your mouse in the object list.
4. In the context menu select the Paste command to paste the value from the clipboard.

To copy a value via drag & drop:

1. Use the left mouse button to click on the value in the coordinates window.
2. While holding down the mouse button, drag the value to the desired target object and then release the button. If the window of the target object is not visible, move the mouse to the object’s tab and wait until the window appears in the foreground.

Note You can copy the value to the object list, to the data view or to any presentation object, such as a table or text object.

Copying the Contents of the Coordinates Window to the Clipboard

You can copy the content of the coordinates window to the clipboard as a picture and insert it into a diagram, a table or a document. This is a static copy, which means that the figures in the copy will not change if you move the cursor at a later stage.

1. Use the right mouse button to click on the coordinates window.
2. From the context menu, choose the Copy Coordinates command to copy the content to the clipboard.
3. To insert it in the target object, open it and select Home[Clipboard] > Paste.

Inserting Dynamic Coordinates into a Diagram, Document or Worksheet

You can insert a dynamic copy of the coordinates window content into a diagram, document or worksheet. The figures in the copy will not change if you move the cursor at a later stage.

1. Open the list box under Design [Diagram Layout] > Add Diagram Element window and select Coordinates.
2. Use your mouse to position the copy.

Note: If you deactivate the cursor, the inserted coordinates will be fixed and the last content shown will remain the same without being updated.

Changing the Font Size of the Coordinates Window

You can change the font of the coordinates window in a diagram, document or worksheet:

1. Select the coordinates window in a Document, Diagram or Worksheet.
2. Change the font size in the Format[Font] > Font Size list box.

See Also

Analyzing Data with Cursors 188

Coordinates Window 192

Inserting Dynamic Coordinates into a Diagram, Document or Worksheet 220

Adding a Custom Coordinate

1. With cursors active, right-click with your mouse on the coordinates window and select Add Coordinate. You can also find the command in the ribbon under Cursors[Cursors] > Show Coordinates Window > Hide Coordinate.
2. In the Select Coordinate dialog box, choose one of the coordinates provided from the list or choose (New custom coordinate) to program a coordinate in FPScript.
3. To program your own coordinate, follow the instructions displayed in the Custom Coordinate Properties dialog box.

Editing a Custom Coordinate

1. With active cursors, right-click with your mouse in the coordinates window on the coordinate to be edited and choose Edit Custom Coordinates . Edit Coordinate The command can also be found in the ribbon under Cursors[Cursors] > Show Coordinates Window > Add coordinate.
2. Edit the coordinate in the Custom Coordinate Properties dialog box.

Note Text the FPScript code with the Debugger before closing the dialog box.

Displaying and Hiding a Custom Coordinate

To hide a coordinate:

- Click with your right mouse button in the coordinates window on the coordinate you want to hide and choose Hide Custom Coordinate. Hide Coordinate You can also find the command in the ribbon under Cursors[Cursors] > Show Coordinates Window > Hide Coordinate.

To delete a coordinate permanently:

1. Use your right mouse button to click on the coordinates window and select Coordinates Properties. You can also find the command in the ribbon under Cursors[Cursors] > Show Coordinates Window> Coordinates Window Properties.
2. On the Coordinates Window tab, select the coordinate to delete from the Custom Coordinates list.
3. Click on Remove Coordinate.

To display the hidden coordinate again:

1. Use your right mouse button to click on the coordinates window and select Coordinates Properties. You can also find the command in the ribbon under Cursors[Cursors] > Show Coordinates Window> Coordinates Window Properties.
2. On the Coordinates Window tab, select the coordinate that you want to show from the Display & Name column of the Custom Coordinates list.

Displaying a Presentation Object in the Coordinates Window

In the coordinates window, you can display any presentation object, such as a table with embedded calculation results. The text is then automatically updated each time the cursor is moved, so that embedded calculations referring to the current cursor positions always remain current.

Using the Properties dialog box:

1. Use your right mouse button to click on the coordinates window and select Coordinates Properties. You can also find the command in the ribbon under Cursors[Cursors] > Show Coordinates Window> Coordinates Window Properties.
2. On the Coordinates Window tab, select the option Object under Display in window.
3. In the text box, specify the name of the object to be displayed with the extension corresponding to the object type, or select an object from the list.

Using drag-and-drop:

1. Enable the cursors in the object in whose coordinates window you would like to display a presentation object so that the coordinates window appears.
2. Now drag the object that should appear in the coordinates window from the object list into the coordinates window. Make sure that you do not activate the object list by clicking on it, otherwise the coordinates window will disappear.

4.7 Working with Curves

Dimensioning Curves

1. Enable the cursors of the diagram whose curve you would like to dimension or of the document where the diagram is located.
2. If the diagram is in a document or worksheet, click on it to select it.
3. Position the cursors on the curve to be dimensioned.
4. Open the list box under Cursors[Marker] > Place and select a dimensioning.
5. If you selected a horizontal or vertical dimension line, you can drag it with your mouse to move it vertically or horizontally.

Translating Curves

To translate a curve in increments using the keyboard:

1. Set the cursor on the curve you want to translate.
2. If necessary, press the NUM LOCK key on the numeric keypad of your keyboard in order to turn off NUM LOCK mode.
3. Use the arrow keys on the numeric keypad correspond to the numbers 2, 4, 8 and 6 in order to translate the curve in the particular direction. You can increase the increments by holding down the CTRL or ALT key. Relevant buttons can be found under Cursor[Translate Curve].

To translate a curve using the mouse:

1. Hold the CTRL key down and move the mouse to the horizontal or vertical line of the active cursor or to the location where the two lines cross, depending on whether you want to translate the curve vertically, horizontally or in both directions.
2. Now hold down the left mouse button and drag the curve with the mouse to the desired position.

To translate a curve around a predefined numerical value.

1. Disable the cursors using the command Cursors[Cursors] > On/Off.
2. Click on the curve or on its axis label to select it.

3. In the Properties window, enter the desired values in the fields X offset and Y offset in the unit of the data upon which the curve is based.

To align two curves:

1. Place the original cursor on the curve that is not to be translated, but is to serve as a reference for the translation.
2. Move the cursor to the desired reference point on this curve.
3. Now activate the leading cursor and place it on the curve to be translated.
4. Move the cursor to the point on the curve to be aligned with the reference point of the other curve.
5. Open the list box under Cursors [Translate Curve] > To Cursor and select one of the following commands: Horizontal to Cursor, Vertical to Cursor or Both to Cursor.

Note The data of the translated curve is not changed during translation. The curve just appears as translated. The original data of the data set upon which the data is based continues to be displayed in the coordinates window as well. By translating the curve, its data deviates from the axis scaling.

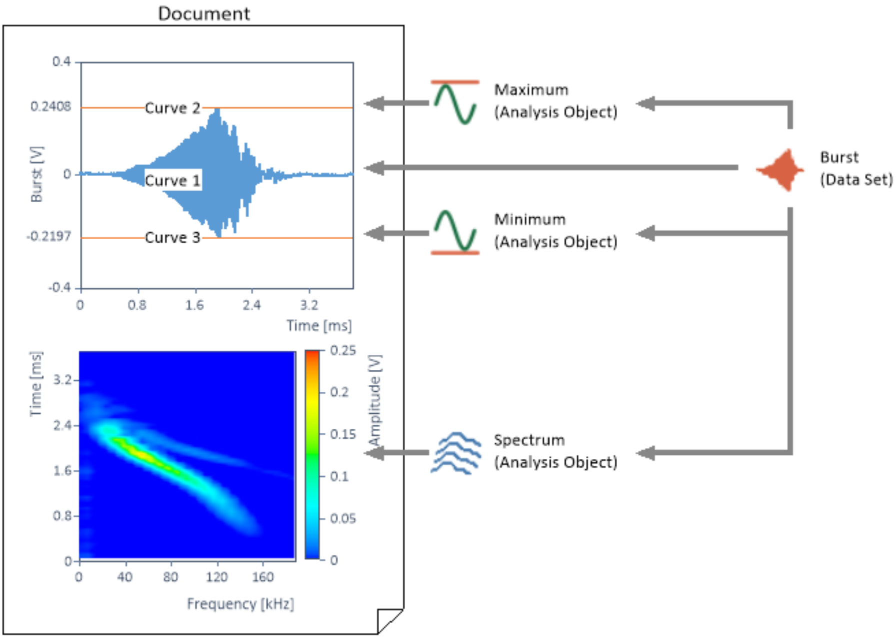
Canceling a Curve Translation

1. Place the cursor on the curve whose translation you want to undo.
2. Open the list box under Cursors[Translate Curve] > Reset and select one of the following commands: Reset Horizontal, Reset Vertical or Reset Both.

5 Analyzing Data Mathematically

Traditional applications for data analysis use a number of analysis commands to generate new data sets from the raw data sets available, which then contain the results of the analysis. In FlexPro, however, you set up a dynamic network of analysis objects 293 and formulas 235 to evaluate your data. Each analysis object and each formula represent the result of an analysis, such as the mean value or the frequency spectrum of a signal. Formulas are programmed in the FlexPro FPScript 246 analysis language. Analysis objects are special formulas that are managed by FlexPro and for which you can set parameters via the Properties dialog box 107. To calculate an analysis result, the FPScript code of the formula or the analysis object is processed to calculate the result. When processing the formula, FlexPro accesses the source data on which this formula is based and to which the formula is linked. For an analysis object, specify the source data to be used on the Data tab of the Properties dialog box. For a formula, use the source data as you would a variable in FPScript code. You can nest the object network that you are creating as deeply as you like. Not only can the source data of an analysis object or a formula be data sets 132, but the data can also be different analysis objects or formulas. For all objects that can supply data, i.e. for formulas, analysis objects and data link objects 160, FlexPro uses the term data object 118. You can use these types of data objects like data sets and display them as a diagram or table, for instance.

The following illustrates a simple analysis network:



The object-oriented data analysis from FlexPro has two distinct advantages over the conventional approach. First, you can modify the analysis objects and formulas at any time to optimize the analysis; second, you have the option of using the analysis that was created just once for as many new data sets as you would like without having to repeat all of the steps in the analysis. All settings that you make during the analysis remain stored in FlexPro's network of data objects. After changing a setting or after importing new data, you only need to use the Update command 105 to update all documents with the latest information.

If you are new to FlexPro and used the conventional methods for data analysis in the past, the object-oriented application from FlexPro will require some changes in your thought process. However, you will learn to appreciate the advantages of this method rather quickly.

Analysis Wizard

The Analysis Wizard offers you a wide range of ready-made analyses to choose from and creates a complete evaluation of the data sets selected by you. The final analysis does not only contain the desired calculation results, but as an option also contains specially adapted diagrams and tables that present these results. If required, the wizard can even create a finished document for your analysis.

Analysis Templates

You can store analyses created one time as an analysis template in the current database or in a template database. This type of analysis template consists of any network of interdependent objects. In the example above, these would be the document, the diagram within it, and three analysis objects. Objects in the lowest hierarchical level, i.e. those that do not depend on any other objects, are replaced by selected objects during subsequent use of the template, as long as they were not marked as belonging to the template when it was created. In the example above, this would be the data set. However, this can also be any other type of object, such as a diagram. You can then use the analysis templates in the Analysis Wizard to create new analyses or to apply it directly to the recently imported data during data import.

Analysis Automation

All analyses in FlexPro are created using analysis objects or formulas, allowing for automation of the analysis without any programming.

To update a document or worksheet, you only need to run the Update command. FlexPro then updates all diagrams, text and tables in that object. If, for example, a formula or an analysis object has been entered as a curve, FlexPro checks to see whether a new calculation is required and, if necessary, carries out the re-calculation. If, in turn, the formula uses other formulas and/or data sets in calculations, then this network is completely processed.

If you have set up data link objects 160 for importing external data, the original data is accessed with each update, as long as it has changed since last updated. Changes thus affect the results immediately.

Smart Update

When you update a document or diagram, all of the formulas and analysis objects upon which this document is based are also updated. This type of network can quickly cover several hundred formulas. FlexPro "Smart Update" technology ensures that only the formulas with indeterminate results are re-calculated. If a formula was already calculated during a previous updating process, then the last result is re-used, if this is at all possible.

FlexPro checks the formula code here for vagueness, which means that re-calculation *could* provide a different result. Only those formulas are re-calculated for which this is the case. A typical example of a formula with a vague result is a formula that uses the Noise function or the Input function. Formulas that refer to data sets or results of other formulas are re-calculated if at least one of the data sets has been edited or if one of the formulas used has to be re-calculated.

FlexPro also ensures that each formula is not calculated more than once within a single update. You can therefore assume that a random signal used in several formulas represents the same instance in each of these formulas. An exception to this are formulas with arguments 289. These are always calculated.

However, "Smart Update" includes not only formulas, but presentation objects as well. For a diagram, FlexPro checks for each individual curve, for instance, to see whether it must be re-plotted.

5.1 Working with Analyses

Structuring the Data Analysis

The most important task when analyzing data using FlexPro is to structure the analysis correctly, which means that the analysis problem should ideally be broken down into a network of analysis objects and formulas. Provided that you are working exclusively with the analysis objects provided by FlexPro, this structure is automatic. Just select the data set to be analyzed and choose an analysis object to obtain the relevant result. For instance, if you want to display the signal minimum and maximum in your document, create two analysis objects Statistical Quantity, one for the maximum and one for the minimum.

If, however, you use your own algorithms, i.e. formulas that you would like to program in FPScript, you should carefully plan this structure. The following elements are available to help you build your analysis network:

Analysis Object 293: If an analysis object is available for your analysis, it is recommended that you use it. Analysis objects have an advantage over conventional FPScript formulas in that you can set parameters via a dialog box.

Formula 235: Calculates a result that you would like to display or that you need for further calculations.

FPScript Function 289: Calculates a result for one or more arguments returned when calling the function.

To structure the analysis, use the following procedures:

1. First plan an analysis object or a formula per result that you would like to display.
2. Analyze each formula that you planned for multiple instances of code elements, which are completely identical and which calculate a single result in one of the FPScript data structures. For these types of code elements, create additional formulas whose results you then access in the formulas already present.
3. Now you can analyze your code for recurring elements that use different data, but otherwise have the same structure and calculate a single result in one of the FPScript data structures. For these types of elements, create FPScript functions that you call from existing formulas and that return the data to be included as arguments.

A carefully planned analysis should be as efficient as possible and should not have redundant code elements. Avoid redundant code elements by "outsourcing" them in separate formulas or FPScript functions. You can also achieve efficiency by avoiding unnecessary loops in the FPScript code.

Using the list data structure correctly

As described above, each formula should return as its result exactly one of the final results to be displayed. In rare cases, however, it is impossible to separate calculation of several results without considerable loss of efficiency. If an algorithm returns, for instance, several results, then in principle it is always possible to separate the calculation of the individual results by using the same algorithm in one formula for each result. However, this leads to repeated processing of the algorithm.

In the List 124 data structure you can bundle any number of calculation results. The most frequent use of the list is for multichannel analyses. The same calculation is carried out for n channels and the result is a list with n elements. But you can also use lists for the case mentioned above:

1. Create a formula using the appropriate algorithm to calculate the results.
2. Return all results as a list data structure.
3. Name the elements of the list according to the calculated partial results.

Analyzing Multiple Measurements

Using an Analysis on Several Similar Measurements

Analyzing multiple, similar measurements is one of the most important automation tasks in measurement applications. Since FlexPro manages each analysis as a dynamic object network, you only need to "exchange" the data sets upon which the analysis is based.

To do this, you set up several folders in the project database. Each folder contains the respective data sets from a certain measurement. FlexPro automatically creates these folders when importing measurement data files if you select Per file under Data[Import Settings] > Subfolders. Here, it is important that the data sets from the various measurements are always given the same name. You create the documents and analysis objects in the parent folder.

Now for example, before you update a document and thus trigger a calculation, activate the subfolder from which the data is to be taken by selecting it in the Activate or deactivate subfolder drop down list at the top of the Folders window. All of the objects in the activated folder are then displayed in the parent folder. FlexPro first searches in the activated folder for all data sets referenced in formulas, curves, etc., and then, if they are not found there, it searches within the parent folder.

For this to work for all references to the data, only the name and not the entire path name of the data set must be entered. In the curve, for example, Signal should be entered as the data set and not \Measurement1\Signal. In the latter case, this would mean that the data set would continue to be taken from Measurement1 after the folder Measurement2 is activated. You can make sure that this is the case by activating a measurement folder already before creating the analysis and documentation and then selecting the data sets blended-in, e.g. when creating a diagram. FlexPro recognizes that a simple name is enough to refer to the data set and then only enters the name as a reference.

There is an Analysis.fpd project database located on your hard drive. In the project database there is an Automated Analysis folder where you can find an example covering the topic above.

The project database path name is usually `C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples\Analysis.fpd` or `C:>Users>Public>Public Documents>Weisang>FlexPro><%VERSION_COMMERCIAL%>>Examples>Analysis.fpd`.

Creating an Analysis Covering Several Measurements

This type of analysis lets you find trends and outliers, for instance, in a series of measurements. What is important in this case is that the analysis is dynamic in relation to the number of measurements to be analyzed, which means that it is not changed after completed on additional measurements, but only has to be updated.

You can achieve this using FlexPro indexing and the DataQuery 146 object. The Indexer scans all of your measurements and stores the collected data in a database. You can now use the data query to access any data sets, such as all channels called "Pressure" with a minimum value above 10 Pa. The data query provides the results as a List 124. You can then use this list as a starting point for analyses and presentations. When you add a list containing source data into a FlexPro analysis object, a list with the same number of results is output. When you display a list in a diagram, the diagram shows the same number of curves as items contained in the list. It is therefore possible to create an analysis that analyzes exactly the data sets you require and that is dynamic in relation to the quantity of data.

A more detailed tutorial is available under: Test Series Analysis Tutorial.

Efficient Data Analysis

The FlexPro's data analysis is based on the powerful FPScript 246 analysis programming language. In addition to the optimum setting 68 of the FPScript runtime environment and the FlexPro object database, using FPScript correctly plays a decisive role in efficient analysis. Avoiding loops is the most important factor when creating an analysis algorithm using FPScript. To achieve this, the following FPScript features are available:

- FPScript offers the ability to calculate complete data sets in simple arithmetic expressions.
- For all common analyses, FPScript offers a variety of integrated functions 254.
- The event isolation 352 functions localize events such as extreme values, slopes and level crossings and allow you to split and extract sections in lists.
- The Index operator extracts individual values or ranges of data.
- Operators and functions are automatically applied to all elements of a list.

With the above tools, loops can almost always be avoided, especially across all values in a data set. If a loop over individual values cannot be avoided in FPScript, then you should use the For Each Value...End statement, which is the fastest of all FPScript loops.

Example

For individual waves from a power signal, the RMS value should be calculated and output as a data series 123. This task can be solved as follows:

1. With the LevelCrossings function, the positions (indices) of all positive zero crossings in the signal are determined as data series.
2. Using a For loop, run across all values in the data series.
3. For every two neighboring positions with the Index operator, take the range between these two indices from the signal.
4. For each signal segment formed this way, calculate the square mean (RMS).
5. With the Append operator, collect the RMS values calculated this way in a data series.

The example above uses a loop, not across the individual values of the data set, but instead only for the periods that occur therein. It is simpler and more efficient to use event isolation with subsequent quadratic averaging:

1. The Event Isolation analysis object with splitting into segments without borders can deliver the individual periods as elements of a list with the search for positive level crossings.
2. The Statistical Quantity analysis object for calculating the root mean square value then converts the list of periods into a list of the individual RMS values.
3. A small FPScript formula using the LevelCrossings function generates a data series from this.

You can find both examples in the FPScript Example database 236 in the Wave Analysis folder.

5.2 Working with Analysis Templates

Creating an Analysis Template

1. Select all of the objects in the object list that you want to save as an analysis template.

Note It is sufficient if you select only the objects at the highest hierarchical level. FlexPro automatically adds linked objects to the template.

---

2. Now click Home[Selected Objects] > Save as Template.
3. On the first screen of the Template Wizard that appears, for the Template Type, choose Analysis.
4. Follow the wizard's next steps and use Help in the wizard if you require additional assistance.

See Also

Analysis Templates 227

Organizing Analysis Templates 234

Template Databases 49

Using an Analysis Template

Use the Analysis Wizard

1. To select the objects from the object list which you want to create an analysis.
2. Select Insert[Analyses] > Analysis Wizard > From Template.
3. Follow the steps of the Analysis Wizard and use the wizard's Help if you require additional assistance.

Data Import

1. Select the analysis template from the following list box: Data[Import Settings] > Analysis template.
2. Import the data on which the analysis will be used.

See Also

Analysis Templates 227
Creating an Analysis Template 233
Organizing Analysis Templates 234
Template Databases 49

Organizing Analysis Templates

Use the Organizer dialog box to delete analysis templates or to copy templates from one data base to another.

Opening the Organizer dialog box and selecting the template type

1. Click on File > Info > Organizer.
2. In the Organizer dialog box, click on the Templates tab.
3. In the Template type list box select Analysis.

Opening the Current Project Database or a Template Database for Organizing

- Choose a template database in one of the Items available in list boxes.

Opening a shared template database or any other database for organization

1. Click on one of the Close Project Database buttons to close the database currently displayed in the list.
2. Click on the Open Project Database button and select a project database on your hard drive or from the network.

Copying templates from one project database to another

1. Select the templates you would like to copy.
2. Click on Copy.

Deleting templates from a project database

1. Select the template that you would like to delete.
2. Click on Delete.

5.3 Formula

A formula is a special data set 132 where the value is not fixed when stored, but is instead replaced by calculation operations that can calculate the value when required. You usually write these formulas in the script language FPScript 246. FlexPro offers you special analysis objects 293 for common analyses so that you do not have to learn how to use FPScript.

Formulas are the main tools for automating the analysis. In the nested network of formulas for a project, the entire "knowledge" is saved so that you can repeat the analysis as often as you like, and the data to be analyzed can be assigned 230 at the click of a button.

You can use formulas wherever data sets are also permitted, such as for a curve in a diagram.

FlexPro compiles the formula into binary code, which guarantees rapid processing. The FlexPro "Smart Update" technology 227 ensures that unnecessary re-calculations are avoided. The result of the last calculation is stored temporarily and re-used for as long as possible.

Note With FlexPro View, the analysis options are limited to basic mathematical operations and simple, statistical evaluations.

Elements of FPScript Formulas

- Constants 271: These are constant expressions such as values in the various data types.
- Variables 266: You can use variables in a formula to save interim results. This allows you to save computing time by only performing a calculation once for terms that are used several times. You can also structure your formulas better by first assigning parameters to a variable and then continue on with the formula using the variable.
- Data sets 132: Data sets are used like variables. If a data set name is used in a formula, this name represents the data that is read when the formula is later calculated. This data can also be formulas. The formulas are then first evaluated to obtain the data. You cannot make assignments to formulas and data sets.
- Pre-defined constants 276: Some mathematical constants, such as PI or E, as well as constants, which are used as function parameters, are available as pre-defined constants. You cannot use these names for data set names and/or variables.

- Operators 276: You can use operators to calculate and modify data. FlexPro provides you with a variety of operators with which you can also calculate complete signals.
- Statements 262: Each formula consists of one or more statements, where one line is used for each statement. However, you can also write several statements in a single line, if you separate the statements from each other using a semi-colon ';'. You can also split a statement over several lines. For lines that are to be merged with the lines following, you have to enter a backslash '\' or an underscore character '_' as the final character. In FlexPro formulas you can use different loops and conditional statements, for example, to be able to perform a complex analysis for all of the data series in a signal series.
- Functions 254: FlexPro features a variety of functions that you can use to generate signals and carry out analyses. You can also define custom functions 289. These are formulas that you provide with arguments and that you can call like a function in other formulas.
- Object properties 279: The various objects in FlexPro provide special functions for accessing their attributes. For example, you can access the comments for a data set.
- Comments 262: These start with "//" and then extend to the end of the line.
- Python Functions und Constants You can use Python functions and constants directly in your FPScript formulas.

Examples of FPScript Formulas

The project database FPScript is located on your hard drive and contains various FPScript programming examples.

The project database path name is usually C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples\FPScript.fpd or C:>Users>Public>Public Documents>Weisang>FlexPro><%VERSION_COMMERCIAL%>>Examples>FPScript.fpd.

Python Formula (FlexPro Professional, Developer Suite)

If you have licensed and activated the Python interface of FlexPro, you can also create Python formulas in addition to FPScript formulas. Python formulas are written in the open-source programming language Python and give you access to the extensive function libraries of this language directly in FlexPro. FlexPro uses the numerical data types of Python's NumPy library to exchange data between FlexPro and Python. NumPy is a library that must be imported into a Python program using the import command before it can be used.

In a Python formula, you can use the flexpro module to access data from other data sets or formulas, aggregate Python data into the composite data structures used in FlexPro and return the result of the Python formula to FlexPro.

Unlike FPScript, Python cannot calculate with physical values. However, it is possible to read in the units of data sets and assign units to the result of the Python formula.

Smart Update does not work automatically with Python, i.e. FlexPro cannot recognize whether a Python formula needs to be recalculated even if the data it is calculating has not changed. You do this by setting the Indeterministic option on the Result tab of the formula's Properties dialog box.

Important note In order for FlexPro to ensure the integrity of the object network, formulas in FlexPro must be programmed so that they only have their result as output. A formula must therefore not produce any "side effects". The range of functions and commands in FPScript ensures this for FPScript formulas. You can only have read access to other data sets and cannot perform operations, such as writing files. In Python, these restrictions do not exist and it is therefore up to the programmer's discipline to adhere to this basic rule.

Python Prolog Code

Libraries that you want to use in Python must first be imported using the import command. So that you do not have to do this in every Python formula, you can specify a prolog code under File > Project Database Properties > Python, which is always executed before the actual code of the formula. A typical prolog code looks like this, for example:

```
import flexpro

import numpy as np
```

You must always import the flexpro and numpy modules.

Using Further Python Modules

The numpy and flexpro modules are already pre-installed. If you want to use other modules, you must tell FlexPro where to find them. You can specify the hard disk folders in which FlexPro should search for Python modules under File > Options > Python > General settings.

Important note In contrast to FPScript, Python can be used to write programs that manipulate files and the operating system, provided your user permissions allow it. Therefore, only import modules that originate from trustworthy sources and only specify hard disk directories to which there is no unrestricted access! If in doubt, contact your company's IT security officer.

Working with Formulas

Creating a Formula

To create an empty formula:

1. In the Folders window, select the folder in which you want to place the new formula.
2. Click Insert[Data] > Formula.

A new formula is created that you can edit.

Create a formula that aggregates data into a list.

1. In the Folders window, select the folder in which you want to place the new formula.
2. Select several data sets or other formulas from the object list.
3. Open the list box under Insert[Data] > Formula and select List.

A new formula is created. The `SourceData` variable contains a list.

Note You can do the same thing to create a formula that concatenates data sets (Concatenation) or bundles data sets into a data matrix or signal series (Bundle).

Editing a Formula

Before you can edit a formula, you first have to open it. To do this, double-click on the formula in the object list. A window appears where you can then edit the formula. Use the scripting language FPScript 246 for the code of the FPScript formula and the Python 286 language for the code of a Python formula.

The window for editing formulas has three views. Edit the formula in the Code view. In the Data view, the current result of the formula is displayed numerically. The Data view corresponds in its functionality to a read-only Data window. In the Presentation view, the result is displayed as a curve and you can use the cursors to take measurements.

The formula editor works like a common input box. Use the following keys to navigate:

Key	Function
CURSOR KEYS	Move the cursor in the corresponding direction.
CTRL+RIGHT	Moves the cursor to the beginning of the next word.
CTRL+LEFT	Moves the cursor to the beginning of the previous word.
HOME	Moves the cursor to the start of the line.
END	Moves the cursor to the end of the line.
CTRL+HOME	Moves the cursor into the first line.
CTRL+END	Moves the cursor into the last line.
TAB	Indents the selected lines.
SHIFT+TAB	Unindents the selected lines.

Note If you hold down the SHIFT key while navigating, the passage you move over is selected.

Inserting Code Elements into a Formula

For all common elements that you use to construct your formula, FlexPro offers wizards that free you from a large part of the coding workload. Under Formula tools/Design[Insert Code Element] you will find wizards for inserting functions 254, object properties 279, operators 276, constants 271, instructions 262, data sets 132 and index operations.

Note Not all wizards are available for Python formulas.

Inserting a Function

1. Place the cursor at the position where you would like to insert a function.

2. Click Formula Design[Insert Code Element] > Function or press CTRL+SPACE to view a list of available functions.
3. Select the desired function from the list and press ENTER to insert it into the formula.
4. The FlexPro Assistance Window will then appear, which will assist you in entering the function arguments. Another list appears for data arguments that contains data sets in the folder of the formula whose data structure is valid for the argument. For arguments requiring a constant, a corresponding list appears with the valid constants. Use the UP key to switch to the list and to select an entry.
5. After selecting an argument, add a comma, if applicable, to enter additional arguments.

Note To view the assistance window for the argument at a later time, place the cursor on the argument and press SHIFT+CTRL+SPACE.

Inserting an Object Property

a) Using the Formula Editor's Assistance function

1. Next, specify the object by entering one of the keywords This, ThisFpObject, ThisObject or Application or an object reference, such as '\2D-Diagram.2D'.
2. Now enter a period to view the list of available properties.
3. Use the DOWN key to switch to the list and to select an item from the list. Transfer the item using the ENTER key.

b) Using the Object Properties Wizard

1. Click on the position where you would like to insert an object property.
2. Click Formula Design[Insert Code Element] > Object Property
3. A wizard appears to assist you with the remaining steps.

Inserting a Unit Symbol

1. Click on the position where you would like to insert a unit symbol.
2. Click Formula Design[Insert Code Element] > Unit Symbol
3. In the dialog box, select either the required unit symbol directly or first select the physical quantity and then the appropriate unit symbol.

4. Click OK to confirm and apply your selection.

Inserting Data Sets from the Object List

Data sets, formulas and analysis objects that you want to use for the formula calculations can be transferred from the object list into the formula. Simply drag the desired data set with the mouse from the object list to the position in the formula where the reference is to be inserted. A reference to the object is then placed in the formula. If the inserted object is located in the same folder or in an activated subfolder 230, only the name is entered. Otherwise, the path name 259 of the object is entered. The object has to be located in the same project database as the formula. You can also drag a cell range selected in the data view of a folder or a data set window into the formula. An FPScript expression is then inserted that extracts the selected section from the data set.

Search and Replace in a Formula

You can search for text passages in a formula and replace these passages with other text. Use the commands Formula Design[Edit] > Find and Formula Design[Edit] > Replace. Dialog boxes then appear where you can specify the required text. You can use the command Formula Design[Edit] > Find Next to run the last search in each case several times.

Debugging Formulas

After you have written a formula, you should check that the formula works correctly. Three types of errors can occur in formulas: Syntax errors, runtime errors and semantic errors.

Types of Errors

A syntax error occurs when you violate the syntax of the FPScript language, such as if you have written a keyword incorrectly. FlexPro recognizes this type of error as soon as the formula is compiled into binary code. You can use the command Design[Debug] > Syntax Check at any time to check if the syntax in a formula is correct. If there is a syntax error, this is displayed in the Event Log, and you can select the defective section of code by double-clicking on the message.

A runtime error occurs when there is a problem during calculation of the formula, such as if you were to divide by zero. The calculation of the formula is then canceled and the error is displayed in the Event Log. You can select the defective passage in the code by double-clicking on the message. You can use the command Design[Debug] > Run to calculate a formula on a test basis and thus check whether runtime errors occur. The result of the calculation is displayed in a window.

A semantic error occurs when a formula does not work as the programmer expected. These types of errors are often difficult to find because FlexPro cannot localize them on its own. However, FlexPro offers you the Formula Debugger that you can use to step through the formula and analyze all of the interim results.

Formula Debugger

The formula debugger is a powerful tool with which you can execute and test formulas step by step. Once started, the debugger can be in Single Step mode or in Run mode. You can execute single statements in Single Step mode and review their results as well as the contents of local variables. In Run mode, the formula is processed at normal speed until a breakpoint is reached, it is canceled by you, or the end of the formula is reached.

FlexPro displays the content of selected variables and the result of the statement just executed in the Watch window while you are debugging a formula.

The sequence of called formulas from the formula for which you started the debugger up to the current stop position is displayed in the Call Stack window and you can easily switch between the code windows.

All breakpoints currently set in formulas are displayed in the Breakpoints window and you can manage them centrally.

The breakpoints have an optional hit counter. You can define a stop condition, e.g. that a local variable must have a certain value. You can also output a message with embedded variable content when the breakpoint is reached.

Working with the Formula Debugger

Starting the Debugger

To perform controlled execution of a formula until you reach a breakpoint using the debugger:

1. Open the formula which you want to execute under control of the debugger.
2. Click Design[Debug] > Debug/Continue.

Note: The formula is executed immediately and ends when the end of the formula or a breakpoint is reached. You can cancel its execution at any time by selecting Design[Debug] > Break.

To step into a formula using the debugger:

1. Open the formula which you want to execute under control of the debugger.
2. Click Design[Debug] > Step Into.

The debugger marks the first statement in the formula, but will not start executing it.
Next: Stepping into Statements 243.

Step-by-Step Execution of Statements

The debugger must be in Single Step mode. If the debugger is in Run mode, click Design[Debug] > Break.

To execute the highlighted statement and descend into an FPScript/Python function or formula called in it:

- Click Design[Debug] > Step Into.

To execute the entire highlighted statement:

- Click Design[Debug] > Step Over.

To execute the remaining code of the current formula and stop at the next line of the calling formula:

- Click Design[Debug] > Step Out.

To continue execution until the next breakpoint:

- Click Design[Debug] > Debug/Continue.

To trace which formulas were called to reach the current stop position:

1. Bring the Call Stack window to the foreground by clicking on its tab.
2. Click on one of the formulas displayed to bring its window to the foreground. The line with the call of the next formula is highlighted.

Setting and Deleting Breakpoints

To set a breakpoint:

1. Open a formula in which you want to set a breakpoint.
2. Click on the left-hand border of the line for which you want to set a breakpoint.
3. A breakpoint symbol appears at the left border of the line.

To delete a breakpoint:

a) In the code editor

1. Open the formula containing a breakpoint that you want to remove.
2. Click on a breakpoint to delete it.

b) In the Breakpoints window

1. Select one or more breakpoints that you want to delete.
2. Click on Delete Breakpoints in the toolbar.

To temporarily disable a breakpoint:

a) In the code editor

1. Right-click on the breakpoint.
2. In the menu that then appears, click on Enable/Disable Breakpoint.

b) In the Breakpoints window

1. Select one or more breakpoints that you want to disable.
2. Click on the checkbox of one of the selected breakpoints or on Enable/Disable Breakpoints in the toolbar.

To edit the properties of a breakpoint:

a) In the code editor

1. Right-click on the breakpoint.
2. In the menu that then appears, click on Breakpoint Properties.
3. In the dialog box that now appears, you can request further help if necessary.

b) In the Breakpoints window

1. Select the breakpoint that you want to edit.
2. Click on Breakpoint Properties in the toolbar.
3. In the dialog box that now appears, you can request further help if necessary.

To delete all breakpoints in a formula:

1. Open the formula containing all breakpoints that you want to delete.
2. Click Design[Debug] > Delete All Breakpoints.

Monitoring Variable Content

The debugger must be in Single Step mode. If the debugger is in Run mode, click Formula Design[Debug] > Break.

To display all local variables of the current formula in the Watch window:

- Select the Show Local Variables option in the toolbar of the Watch window.

To display the content of an FPScript or Python expression in the Watch window:

1. Select the expression to be monitored in the formula.
2. Click Formula Design[Debug] > Add Watch.

or

1. In the Watch window, click in the empty Expression field at the bottom of the list.
2. Enter the expression to be monitored and press the ENTER key.

To remove a variable from the Watch Window:

1. In the Watch window, select the line(s) you want to remove.
2. Click on Delete Watch in the toolbar of the Watch window.

Note You can only remove watch entries that you have previously added manually. You can hide the automatic monitoring of the current formula's local variables by deselecting the Show Local Variables option.

To display the result of the previous instruction in a data view:

- Select the Show Data View option in the toolbar of the Watch window.

Note To be able to see the result of a Python statement, you must assign it to `this.data`.

To view non-scalar data of a watch entry in detail:

1. In the Watch window, select the entry whose data you want to view by clicking on the border to the left of the line.
2. In the Data Preview window, the data is displayed numerically and as a diagram and you can view it in detail.

FPScript

FPScript is a programming language that is particularly easy to learn. Use FPScript when you want to program your own formulas or functions or as embedded FPScript 380 for accessing data or attributes in presentation objects.

The FlexPro analysis objects 293 are also based on FPScript. However, if you use an analysis object, you do not have to program the FPScript code yourself. Simply configure the analysis via the analysis object's Properties dialog box and FlexPro automatically creates the appropriate FPScript code.

FPScript has been developed specifically for technical/scientific data analysis and offers powerful operators that you can use to calculate individual values as well as entire data series, signals, etc. in a single line. FPScript formulas can provide each of the data structures 122 and data types 118 valid for data objects as the result. In particular, composite data structures consisting of up to three components 121 can be calculated in a single expression.

FPScript not only uses numbers for calculations, but also physical quantities composed of the value and unit. During arithmetical operations the units are adjusted automatically, if necessary, and the unit of the result is derived correctly from the units of the arguments. This eliminates the need for using scaling factors in FPScript expressions as they are required to adjust units when using purely numeric programming languages. Another source of errors is in calculating incompatible units. FPScript recognizes this and outputs an error message.

FlexPro handles formulas that you have written using the FPScript language just like data sets. For instance, you can display the results in diagrams or tables or calculate them in additional formulas or analyses. Therefore, the most common statements in FPScript programs are arithmetical expressions that you can use for calculations. As opposed to common programming languages such as Basic, FPScript allows you to calculate whole data series, signals etc., in addition to scalar values, in one single statement. You do not have to use loop constructs to do this.

Arithmetical Expressions

An arithmetical expression consists of operators 276, which determine the operations, and operands, which supply the data to be calculated. Most operations have two operands. The operator is then written between the operands. For instance, to add two figures, write:

```
1.5 + 2.3
```

For the following expression, you would have to program a loop already if you were using Basic:

```
DataSeries * 2
```

DataSeries is the name of a data set that could, for instance, contain a series of measurements with 10,000 measurement values. This is multiplied by a factor of 2. FPScript carries out this multiplication on a per-element basis, which means that each value in DataSeries is multiplied by 2. The result of the expression is then once again a data series with 10,000 values.

Expressions with the following structure are also very common:

```
DataSeries1 + DataSeries2
```

Here, two data series are added. Again, FPScript proceeds on a per-element basis, i.e. the first value from DataSeries1 is added to the first value from DataSeries2, etc. The result is once again a data series. The two operands should have the same number of values. If this is not the case, the number of calculations matches the number of values in the shorter data set.

You can also string several operations together in a single expression; for instance:

```
2.5 + 3 * DataSeries
```

The above example shows linear scaling of a data series. The Y axis division is 2.5 and the gradient is 3. When concatenating operations, operator priority and associativity 276 determine the order in which the individual operations are carried out. Multiplication always takes precedence over addition. Therefore, FPScript first multiplies the DataSeries by the value 3 and then adds 2.5 to the result. If you would like to determine the order of calculation yourself, you have to use parentheses, as you would in standard mathematics:

```
(2.5 + 3) * DataSeries
```

In this case, all values in DataSeries are multiplied by the value (2.5 + 3), which equals 5.5.

The associativity of operations plays a role if several operations with the same priority are strung together. Most operations are left-associative, meaning that the operations are carried out from left to right:

```
DataSeries1 + DataSeries2 - DataSeries2
```

corresponds to the following expression using parentheses

```
(DataSeries1 + DataSeries2) - DataSeries2
```

Some operations only need one operand. In this case the operator is written before the operand:

```
-DataSeries
```

negates all values in DataSeries, for instance.

Quantities in FPScript

Physical Quantities

One of the strengths of FPScript is the ability to perform calculations using physical quantities composed of a value and unit. FPScript not only manages the unit symbol, but also the SI dimension, which makes it possible to check and, if necessary, transform different units for compatibility.

Entering Quantities

By appending a unit to a constant, the constant becomes a quantity. The value of a quantity can be of the 32-bit or 64-bit floating point data type and real or complex. Quantities of the integer data type are not supported.

The unit is written after the constant and separated by spaces, such as in:

```
1.3 N
```

Degrees (°), minutes (´) and seconds (´´) are an exception. These units are written immediately after the numeric value without spaces:

```
15° + 30´ + 45´´
```

or

`15° + 30' + 45"`

New units can be formed from most units using an SI prefix:

`1 kHz`

`1 MHz`

`1 µm`

FPScript also accepts the notation frequently used with u instead of µ for the prefix "micro":

`1 um` equals `1 µm`

However, u is written alone as the symbol for interpreting the atomic mass unit:

`1 u` equals `1 Da` equals `1 Dalton`

FPScript accepts the symbol or the name of a unit. Symbols are case sensitive, but case is irrelevant for names. The following are correct forms of notation:

`5 V`

`5 mV`

`5 Volt`

`5 volt`

`5 Millivolt`

`2 Ω`

`2 Ohm`

The following, on the other hand, are not accepted:

`5 mVolt`

`5 Volts`

Derived units can be constructed as a product of powers of known units.

`1 N m`

`1 N*m`

`1 N·m`

Exponents are introduced with the ^ character:

`1 m s^-2`

For the exponents 2 and 3, FPScript also accepts the superscript numbers:

`1 m²`

`1 m³`

Instead of negative exponents, you can also use the / character (division):

```
1 m/s^2
```

```
1 1/min
```

Make sure to use parentheses correctly:

```
1 m/(s kg)
```

If you use Unit monitoring on the Unit Manager tab of the project database Properties dialog box and set it to Tolerant, you can omit the empty spaces or multiplication sign between the basic units:

```
1 Ws
```

However, the result of this is that it could be interpreted incorrectly, such as in the following:

```
5 Volts
```

is then accepted as

```
5 V·o·l·t·s
```

equals 5 volts·octet·liter·ton·second.

Note: You can avoid misinterpretations by using unit symbols only and separating them with spaces. Use µ and not u for the "micro" prefix. Check your use of upper/lower case in the prefix and unit symbol. Typical errors include S (Siemens=conductance) instead of s (seconds=time) or M (mega=10^6) instead of m (milli=10^{-3}).

You can also use rational exponents. The following example depicts the franklin from the Gaussian unit system in SI units:

```
1 g^(1/2) cm^(3/2) s^-1
```

Units may contain a prefactor and an offset. The following depicts the unit ° Fahrenheit in the Kelvin SI unit:

```
1 0.555555555555555K+255.3722222222222
```

In the event of ambiguities, you will have to place the unit between apostrophes:

```
Dim s = 2
```

```
v = 3 'Vs'*s
```

This is also required for a negative prefactor:

```
5 '-1.602176487E-19 C'
```

Checking the Unit Symbol

FlexPro breaks a unit down into its elements during compiling and stores the unit symbols of the elements individually with their corresponding exponents. In addition, the unit symbol as specified is saved as a custom unit symbol. For

```
1 N m/(1/min)
```

the custom symbol "Nm/(1/min)" is saved with the exponents 1, 1, 1 as the symbols of the basic units "N", "m", "min". When calculating different units, the custom symbol is discarded and the result unit symbol is reconstructed from the symbols of the basic units:

```
1 N m/(1/min) * 1 min^-1
```

results in:

```
1 N m
```

You can replace the custom symbol with the ChangeUnitSymbol function without affecting the unit. You can also use the same function to delete it. FlexPro will then regenerate the unit symbol from the basic units. For the example above, the result is "N m min".

Using Quantities for Calculations

FPScript adjusts the units to each other before processing the quantities. The following FPScript expression

```
1 V + 2 mV
```

for example, returns the result 1.002 V and not the value 3, which a calculation without taking the unit into account would return. The right operand is transformed to the unit of the left operand prior to addition.

The same applies to comparison operations:

```
5 mV < 1 V
```

provides TRUE, for instance, and not FALSE.

Another frequent source of error when performing purely numeric calculations of quantity comparisons is calculating incompatible units. The expression

```
1 A + 2 V
```

generates an error message in FPScript, since the units Ampere and Volt have different SI dimensions.

```
1 km + 5 NM
```

is, on the other hand, permitted, since the nautical mile can be converted to km.

When values, i.e. numbers only, are calculated with quantities, the value implicitly uses the unit of the quantity:

```
1 V + 2 = 2 + 1 V = 3 V
```

This basic principle, which applies to comparison and other operations as well as to function arguments, makes it easier to write "unit-neutral" FPScript code.

For some operations, the unit of the result is formed from the units of the arguments:

```
1 m / 2 s
```

provides 0.5 m/s as the result.

```
(1 m) ^ 2
```

provides 1 m^2 as the result.

For vector data, i.e. data series and data matrices, all elements always have the same unit. The individual components of a composite data structure can, however, have different units, thus, for instance, the Y component of a signal can have the unit Nm and the X component can have the unit s.

The units are accounted for not only in FPScript operators, but also in all FPScript functions.

```
Derivative(Path)
```

provides, for instance, the speed in the unit m/s for the case in which Path is a time signal with the X unit s and the Y unit m.

Depending on the unit monitoring setting, FPScript also permits unknown units for the result, although they cannot be transformed.

```
1 Dummy + 3 Dummy
```

results in 4 Dummy.

The value of a quantity determines the Value operator:

```
Value 5 mm
```

returns the 64-bit floating point value 5.

With the Unit operator you can transform the unit of a quantity:

```
Unit<mm> 2 cm
```

provides 20 mm as the result.

Note: FlexPro's behavior with regard to processing units depends on the Unit Manager tab of the project database Properties dialog box Unit monitoring settings.

Unit Calculation

With the Unit function you can extract the unit of a quantity:

```
Unit(5.3 V)
```

A data series with two values is returned here:

```
Unit({56 V, 7.8 V})
```

Making these two calls provides 1 V as the result.

Thus, you can carry out unit calculations and use the AdjustUnit function to transform a quantity to the result unit:

```
AdjustUnit(Frequency, 1 / Unit(Time))
```

The unit of Frequency is transformed to the reciprocal value of the unit of Time.

Percentage Calculation with the Units % and ppm

When the option below is enabled File > Info > Project Database Properties > Unit Manager > Percentage Calculation with the Units % and ppm the basic arithmetic operations addition and subtraction are calculated in a special way for operands with the unit % or ppm on the right side

```
200 m + 10 % = 200 m + 200 m * (10/100) = 220 m
200 m - 10 % = 200 m - 200 m * (10/100) = 180 m
```

For all other operations, the units % and ppm are first converted to the unit 1:

```
200 m * 20 % = 200 m * (20/100) = 40 m
200 m / 20 % = 200 m / (20/100) = 1000 m
```

The only case in which the units % and ppm remain is if the right operand has no unit:

```
10 % * 3 = 30 %
```

For the unit ppm, the calculation is carried out in a similar manner with the factor 1,000,000 instead of 100.

Exception Handling

FPScript exception handling allows you to react flexibly to errors in your formulas. Exceptions are all errors that may occur during the execution of FPScript code as well as exceptions that have been explicitly generated using the Throw statement.

The following example carries out a division calculation and generates an exception if the divisor equals zero.

```
Arguments X, Y
If Y == 0 Then
    Throw "Divide by zero"
End
X / Y
```

With the Try...Catch statement, you can catch these types of exceptions and react accordingly. In the case of an error in a calculation, for instance, you can use an error message to prevent the calculation from being aborted. Instead, you can catch the error and, for example, return an empty result:

```
Try
    Return Integral(x)
Catch Exception
    Return Empty
End
```

The variable specified behind the Catch statement is assigned the "thrown" value when generating the exception using Throw. This can be of any data structure. The functions and operators built into FPScript pass a string with a description of the error as a value of the exception.

Exceptions are a powerful concept for error processing by distinguishing between a regular result and an exception. You should, however, only use exceptions to handle errors and not as an alternative for passing results. The reverse is also true. You should not use the formula's resulting value to pass error messages, but generate an exception for this instead.

Functions

FPScript offers you a host of built-in functions. You use a function by writing the name, followed by a list of arguments, which is to be specified in parentheses, into the formula. The following example integrates a data set:

```
Integral(DataSet)
```

If you have to specify several arguments, these are comma-separated:

```
Expand(DataSet, 2)
```

Even for functions that do not need an argument, the argument list parentheses must be included:

```
CurrentDate()
```

Some functions have optional arguments. In most cases, the argument list is created in such a way that the optional arguments are at the end of the argument list and can be omitted:

`Mean(Signal)`, for instance, is the equivalent of `Mean(Signal, MEAN_ARITHMETIC + PROCESS_ROWS)`.

For some functions, however, an optional argument can be omitted in the middle of the argument list:

`Noise(1, , 0)`, for instance, is the equivalent of `Noise(1, NOISE_NORMAL, 0)`.

Functions account for the units of their arguments and also provide quantities as the result:

`LevelCrossings(Height, 25 cm)` also returns the correct result, for instance, if Height is in the unit m and not, like the threshold value, in the unit cm.

Only a small selection of functions 20 is available in FlexPro View.

Python Functions und Constants

If you have set up the Python interface of FlexPro, you can use the various Python function libraries directly in your FPScript formulas.

To do this, use the `python` keyword in FPScript. The following example calls the `sin()` function in the math module with the constant `pi` from the Python numpy module as an argument:

```
python.math.sin(python.numpy.pi)
```

There are FPScript functions whose result can vary even with the same input data, e.g. the `Noise()` function. If such a function is used in an FPScript formula, it must be recalculated each time it is updated, even if the data it calculates has not changed since the last calculation. FlexPro automatically recognizes this for all FPScript functions, but not for Python functions. When calling a Python function, you must therefore specify whether the function is indeterministic with regard to its arguments or not. You do this by setting the Indeterministic option on the Result tab of the formula's Properties dialog box. An FPScript formula with this option set is recalculated each time it is updated.

Result of Formulas

When a formula is processed, the result of the last statement is saved as the result of the formula and returned to the caller.

The following example generates a signal with a sine wave in several steps:

```
Dim x, y
// Create data series with 100 increasing time values
x = (100, 0 s, 0.1 s)
// Sine with frequency of 3.5 Hz and amplitude of 5 V
y = 5 V * Sin(2 Pi * 3.5 Hz * x)
// Combine X and Y to form signal
Signal(y, x)
```

The formula result is the value that is returned by the Signal function.

If you do not want to pass on the result of the last statement, you must use the Return Statement:

```
If NumberOfRows(Signal) < 2n Then
// If Signal has less than 2 values, then
// Pass void floating point value
    return ?
End
// Otherwise, calculate sampling rate
1. / (Signal.x[1n] - Signal.x[0n])
```

FlexPro very rarely re-calculates formulas. It ensures that each formula is only calculated once during an updating process. Formulas with arguments 289 form an exception. These are calculated with each call, even if the same arguments are passed several times.

If there is already a result present from the previous update procedure, FlexPro first checks this to see if it is necessary to do a recalculation. This is the case when the formula is based on data that has changed since the last calculation or the formula uses functions with a result that is deterministic, e.g., the random function. If the existing result is still valid, then it is re-used without a recalculation.

Extracting Data

In addition to calculating data and creating data structures, extracting parts of data sets is an important use of FPScript; for instance, all local maxima in a signal or the range between two zero crossings. For these types of applications, FPScript offers two very powerful index operators.

Index Operator

The Index Operator extracts individual values or ranges of values using their position in the data structure. All rows in data series and all rows and columns in data matrices are numbered starting with zero. Negative indices are counted from the end, which means that -1 indexes the last row or column. The index operation makes it possible for you to access individual values or ranges using these numbers.

`DataSeries[99n]` reads, for instance, the hundredth value from a data series.

`DataSeries[-2n]` reads the second to the last value from the data series.

`DataSeries[0n, 49n]` creates a data series with the first 50 values from DataSeries.

`DataSeries[49n, -1n]` creates a range from the 50th value to the last value of a data series.

`DataSeries[{0n, 5n, 7n, 9n}]` creates a data series with the values that have indices listed. Here, the Bundle operator was used to bring together the required indices into a data series.

`DataSeries[(NumberOfRows(DataSeries), NumberOfRows(DataSeries) - 1n, -1n)]`

Here, the data series operator is used to generate the indices. The statement reverses the order of values in the data series.

For a data matrix, specify two indices. The first index selects one or more columns and the second selects one or more rows:

`DataMatrix[1n][2n]` reads, for instance, the third value from the second column of the data matrix as a scalar value.

`DataMatrix[-1n]` reads the last column from a data matrix as data series.

`Datenmatrix[][2n]` reads, for instance, the third from a data matrix as a data series.

Alternatively, you can use a 2D Index. This is a data matrix with two rows and n columns. Each column contains the column and row index of a value to be extracted:

`DataMatrix[{ {1, 0}, {0, 1} }]` takes two values from a data matrix as a data series. Corresponds to `{ DataMatrix[1][0], DataMatrix[0][1] }`.

You can also use the index operator on strings to extract a substring or the character code of a single character.

`String[1n, 3n]` extracts a substring with a length of three characters starting from the 2nd character of the string

`String[1n]` takes the character code of the second character from the string. The expression returns a 16-bit integer with the Unicode of the character.

`(Strings[2n])[1n, 3n]` takes the 3rd string from a data series of strings and from this a substring with a length of three characters from the 2nd character. The parenthesis is necessary to separate the data index from the string index.

Value Index Operator

When working with signals and signal series, it often makes more sense to specify the X and Z values directly when indexing sections rather to work with the indices. To do this, FPScript provides the Value Index Operator.

`Signal[[0 s, 4.5 s]]` creates, for example, a partial signal with the points of the first 4.5 seconds.

Note that the operation functions as expected even when the X component of the signal has the unit ms, for instance.

`SignalSeries[[3.5 kHz]]` extracts the signal with the Z value 3.5 kHz from the signal series.

The syntax of the Value Index operator corresponds to that of the Index operator. All you have to do is to include all square brackets twice. FPScript then interprets the indices specified as X or Z values and looks for these in the X or Z component of the signal or signal series in order to obtain the positions.

List Element Operator

You can use the List Element Operator to extract one or more elements from a list, allowing you to work with the list element indices or names.

`List.[1n]` provides, for instance, the second element in the list.

`List.[-2n]` takes the second to last element from a list.

`List.[1, 2]` returns a partial list with the 2nd and 3rd element.

`List.Name` takes the element with name "Name" from a list.

`List.["Current*"]` returns all list elements whose names start with "Current".

Note Please be aware that accessing list items by their names is not always unique, since in a list several items could share the same name.

Accessing Data and Objects

Not only can you access local variables in FPScript, but you can also access external data sets and calculation results. If you use a name such as "DataSeries", FPScript proceeds as follows to find the object with this name. Initially, FPScript checks to see whether a pre-defined constant exists with this name. It then checks to see whether a local variable has been defined with this name in the formula. If neither is the case, and if the FPScript formula is a function with arguments, it checks to see if an argument variable exists with this name. If nothing is found during the search, it searches for a data set or a formula with this particular name. The search takes place in the folder containing the formula. If, however, this folder has an activated subfolder 230, this subfolder takes priority in the search. If the object found is a formula, this is calculated, if necessary, and the result is taken for further calculation.

Path Name

The object path name describes its precise location within the project database.

Absolute path names provide the path from the root folder of the project database to the required object. Folder names are listed as a series, where each folder name ends with a '\'. Since the root folder does not have a name, only a '\' is written for it. Therefore, all absolute path names start with a '\'. The following example locates the object called "DataSet" in the "Measurement1" subfolder of the "Analysis" folder:

```
\Analysis\Measurement1\DataSet
```

The "Measurement1" subfolder is selected this way:

```
\Analysis\Measurement1
```

This access provides all data sets within the folder as a list in the order in which they are displayed in the unsorted object list.

Note If in the example above a data set called "Measurement1" is in the "Analysis" folder, access will be provided to this data set and not to the folder of the same name. To access the folder in this case as well, you need to add the '.fld' extension.

An object called "DataSet" located in the root folder is selected as follows:

```
\DataSet
```

If there are special characters, e.g., spaces, in the path name, or if it begins with a number, then you must specify the path using single quotation marks:

```
'\Analysis\Measurement A\DataSet'
```

`'\01\DataSet'`

To access all data sets in a folder as a list, simply address the folder:

`‘\Analysis\Measurement A’`

If objects are to be accessed that are not data sets, formulas, analysis objects or folders, you have to specify a file name extension 108:

`\Analysis\Document.doc`

You can access the activated subfolder 230 of a folder using the ActivatedFolder keyword, which does not have a '\' at the end:

`\Analysis\ActivatedFolder`

Relative path names provide the path to the target object from the folder from which you are trying to access it. Here, '.\' describes the folder from which you are trying to access it, and '..\' describes the parent folder, which is the folder above it in the folder hierarchy. The following example is accessing a data set located in the neighboring folder called "Measurement2":

`..\Measurement2\DataSet`

This reads the folder name from which access is occurring:

`.\.Name`

You can access a data set located in the "Measurement1" subfolder as follows:

`Measurement1\DataSet`

To access an object located in the same folder, you only need to provide its name:

`DataSet`

or

`Worksheet.wks`

Relative path names never start with a '\', but instead either start with '.\' or '..\' or with the name of the subfolder.

You can use the placeholders '*' and '?' in path names to easily create a list of objects:

`'\Measurement*\Signal'` corresponds to `[\Measurement1\Signal, \Measurement2\Signal, ...]`.

Use the '*' placeholder for a string of any length and the '?' placeholder for a single character.

Note If you use an absolute path name or a relative path name that starts with '.\', then any activated subfolder will be ignored, which means that the path name always describes the specified object exactly, even if there is interference by an object with the same name in the activated subfolder.

Indirect Access

The Indirection Operator ($$) gives you the option of accessing objects using a string with the object name. The following example shows a common use:

```
Name = InputText("Please enter the name of the data set")
Integral($Name$)
```

Object Reference

An object reference is a reference to an object in FlexPro, e.g. to a formula or folder. You can use this type of object reference to access the value of the object reference or to access the properties of the object, e.g. access its name or comments. If you use the path name to a data object in FPScript, it will represent the data object value and will not provide an object reference. To construct it, you need to use the Set statement or the As Object keyword:

```
Dim Obj = DataSet As Object
```

or

```
Set Obj = DataSet
```

The indirection operator ($$) behaves in the same way:

```
Dim Obj = $"DataSet"$ As Object
```

is equivalent to:

```
Dim Obj = DataSet As Object
```

and

```
Set Obj = $"DataSet"$
```

is equivalent to:

```
Set Obj = DataSet
```

and

```
Dim Val = $"DataSet"$
```

is equivalent to:

```
Dim Val = DataSet
```

If a path name does not refer to a data object, but to another object, then it represents an object reference. The following statements are therefore equivalent:

```
Dim Obj = '2D-Diagram.2D' As Object
```

or

```
Dim Obj = '2D-Diagram.2D'
```

FlexPro analyzes an object reference automatically if the value is required for additional calculations. You can use the Value Operator to analyze an object reference explicitly:

```
Dim Obj = $"Signal"$ As Object
```

```
Dim Val = Value Obj
```

Comments

It is important that you add comments to your formulas, particularly for more complex programs. In FPScript, comments start with two forward slashes (//). A comment started this way extends to the end of the line. You can also use empty lines and lines containing only comments to structure your programs. Longer comments extending over several lines start with '/*' and end with '*/'.

Statements

You can structure the FPScript code for your formula in one or more statements. Normally, you write every statement on its own line. You can, however, also write several statements on the same line. These have to be separated with a semicolon ';'. You can also split a statement over several lines. For lines that are to be merged with the lines following, you have to enter a backslash '\' or an underscore character '_' as the final character.

Assignment

You can use an assignment to assign different content to a variable. In a variable containing, for instance, a data series, you can specifically overwrite individual values or sections using the Indexed Assignment. With the Set statement or the As Object keyword you can assign an object reference to a variable. You can then use this variable to access the properties of the referenced object.

Making Decisions with If...Then...Else

The If...Then...Else statement is used to check whether a condition has the value TRUE or FALSE, in order to execute one or more statements conditionally. Normally, the condition is an expression with a comparison operator for comparing two variables or values. You can find information on comparison operators under "Comparison Operators". If...Then...Else statements can be nested to any depth.

Executing a Block When a Condition Results in TRUE

In this case, the keyword Else is omitted:

```
If Value > 2 Then
    Value = 5
End
```

Executing One Block When a Condition is TRUE and Another When the Condition is FALSE

The statements to be executed if the condition results in FALSE are inserted between the keywords Else and End.

```
If Value > 2 Then
    Value = 5
Else
    Value = 0
End
```

Using Loops to Repeat Code

FPScript offers a host of operations to search for events and to process data sets in their entirety. In rare cases, however, you will have to use loops to run across the values in a data series, for instance. With a loop, you usually define a loop variable that enumerates the repetitions. This variable can then be used in an index operation, for instance, to access the corresponding value in a data set:

```
// Declare variable

Dim Result

// Copy data series to be calculated
Result = DataSeries

// Run across all values in the data series
For Each Row i In DataSeries Do
```

```
// Set values less than zero to zero in the result
    If DataSeries[i] < 0 Then
        Result[i] = 0
    End
End
Return Result  // Pass the result data series
```

The above example uses a loop to find a negative value in a data series and to set it to zero. All statements between Do and End are executed several times, where the variable i counts from 0 to the number of values in the data series of data series minus 1. The example also shows a conditional statement. The statement Result[i] = 0. is only carried out if the ith value in DataSeries is less than zero. This example also shows how indents can make FPScript programs easier to read. All statements within the loop were indented by one tab position. The same applies to the conditional statement.

Note You should only use loops if no appropriate function exists and if the problem cannot be described with an arithmetical expression. The integrated functions of FlexPro are many times faster than any loops programmed using FPScript. The example above can, for instance, be re-written as follows:

```
Dim Result, Idx         // Declare variable
Result = DataSeries     // Copy data series to be calculated
// Determine positions of all values < 0
Idx = ValuesAboveLevel(DataSeries, 0, EVENT_INDEX + EVENT_COMPLEMENT)
// Set values at these positions in the result to zero
Result[Idx] = 0
Return Result           // Return the result data series
```

You can use the Break statement to cancel a loop prematurely:

```
Dim Result = 0
For Each Value v In DataSeries Do
    If v == ? Then
        Break
    End
    Result = Result + v
End

Return Result
```

FPScript offers you the following loops:

While...End: Executes a loop while the condition results in the value TRUE.

Do...While: Executes a loop until a condition results in the value FALSE.

For...End: Uses a counter to repeat the statements as often as is specified by the counter.

For Each Value...End: Traverses all values of a data set. No counter is used.

For Each Element...End: Uses a counter to iterate all of the values in a data series or a data matrix.

For Each Row...End: Uses a counter to count all of the items in a list.

For Each Column...End: Uses a counter to iterate all of the columns in a data matrix.

Repeating Statements While a Condition is TRUE

The While...End loop first tests the condition and then executes the statement as long as the condition results in TRUE. The following example searches for the position in DataSet, where the first positive value occurs.

```
i = 0
While DataSet[i] < 0 Do
    i = i + 1
End
```

Repeating Statements Until a Condition is FALSE

The Do...While loop first executes the statement one time and then tests the condition. The following example calculates random numbers until a number greater than or equal to zero is generated.

```
Dim Z
Do
    Z = Noise(0)
While Z < 0
```

Using "For...End"

You can use For...End statements to repeat a block of statements as often as is specified. In loops, use a counter variable with a value that is increased each time the loop is executed.

```
Dim a = "" # 3
For i = 0 To 2 Do
    a[i] = TextInput("Enter text here")
End
```

Using "For Each Row...End"

If, as in the example above, you want to iterate across all elements of a data set, then you can implement this more elegantly with the For Each Row...End loop:

```
Dim a = "" # Input("Count")
For Each Row i In a Do
    a[i] = TextInput("Enter text here")
End
```

Variables

In FPScript, you can use variables for interim results. A variable acts as a placeholder where you can store any results of calculations and recall them when necessary. The name of the variable allows access within the formula to the storage location to which it has been assigned.

Before using a variable for the first time, the variable must be declared using the Dim Statement. This creates storage for one or more variables. A variable, which has not yet been assigned a value, is of the Empty 119 data type. You can use an Assignment to assign a value to it. In a variable containing, for instance, a data series, you can specifically overwrite individual values or sections using the Indexed Assignment.

The range of validity for a variable is restricted to the formula in which you use it. Thus, you can use the same variable name in various formulas without risking conflicts.

The local variable SourceData serves a special purpose. For more information on this, read the Lists in Accessing Header Information 282.

Example

```
// Declares the two variables i and Series.
Dim i, Series
// Initializes a variable i with the integer scalar value 1.
i = 1n
// Creates a data series with 100 zeros.
Series = 0. # 100n
```

For complex calculations, it often makes sense to split up the calculation into several statements 262. In a first statement you could calculate, for instance, a partial expression that you then use in further statements. To do this, it is important to assign the results of the calculations to variables so you can use them later:

```
// Declare variable
Dim Angle
/ Calculation of angle
Angle = 2 Pi * f * t
// Calculation of an overlapped oscillation
2 * Sin(Angle) + 3 * Cos(Angle)
```

Here, the calculation of the angle is performed on a separate line.

Building Data Structures

FPScript provides operators and functions to help you build your own data structures 122.

Linear Data Series

For instance, this Data Series operator creates a data series with ascending values:

`(100, 0., 1.)`

100 is the number of values, 0. is the starting value, and 1. is the increment. If you prefer to specify the starting value, the end value and the sampling interval, use the Series Function:

`Series(0, 100, 1)`

generates a data series with 101 values with a starting value of 0, end value of 100 and sampling interval of 1. Here, the data type is always a 64-bit floating point value.

If you use quantities, you can specify different units, but they must be compatible:

`(2000, 0 s, 1 ms)` or `Series(0 s, 2 s, 1 ms)`

The result contains the unit that is furthest to the left and thus is equal to:

`(2000, 0 s, 0.001 s)`

Data Series and Matrices with Constant Content

The Multiply operator is used to create data series or data matrices with constant content:

`1.5 # 100n`

The above example shows a data series with 100 times the value of 1.5. The scalar value 1.5 has been multiplied 100 times.

`1.5 # 100n # 20n` generates a two-dimensional data matrix with 20 data series of 100 values each.

`TRUE # Shape(Matrix)` generates a matrix of Boolean values whose number of columns and rows corresponds to an existing matrix.

Even here you can multiply quantities:

`1.5 N m # 100n`

Data Series and Matrices with Variable Content

You can also bundle different scalar values into data series and data series into data matrices:

`{ 1.5, 2.6, 7.8 }`

The Bundle operator bundles the scalar values into a data series with three values. The length of the list is not restricted. Of course, in addition to constant scalar values, you can write arithmetical expressions that return scalar values.

`{ DataSeries1, DataSeries2, DataSeries3 }` bundles three one-dimensional data series into one two-dimensional data matrix. The data series should be the same length.

All elements of a data series or data matrix are of the same data type. This does not apply to components of composite data structures; for instance, the X component of a signal can have a different data type than the Y component.

Instead of numeric values, you can also bundle quantities. It is sufficient to specify a unit for the first value; this will then apply to the other values.

`{ 1.5 kg, 2.6, 7.8 }`

If you specify different units, they must be compatible. The result contains the unit of the first quantity:

`{ 50 mV, 0.5 V, -0.7 V }` corresponds to the input of `{ 50 mV, 500 mV, -700 mV }`.

Lists with Multiple Independent Elements

The List operator brings together several values or quantities that, for their part, can have any data structures and data types. Each element of a list has the option of having a name. Since each list element can in turn also be a list, any deeply nested structures can be built.

`[ <Solution> DataSeries, <ChiSquare> ScalarValue ]` compiles the solution vector from a curve fitting and the goodness-of-fit measure into a list. The names provided in angle brackets are optional and describe the list elements.

You can use the placeholders '*' and '?' in path names to easily create a list of objects:

`'\Measurement*\Signal'` corresponds to `[\Measurement1\Signal, \Measurement2\Signal, ...]`.

Use the '*' placeholder for a string of any length and the '?' placeholder for a single character.

Note The use of list element names is no longer recommended in FlexPro version 10 and higher. Instead, use the AssignHeader function to assign header information to the list elements.

Lists are used for multichannel analyses or when a single calculation produces several independent results that cannot be calculated separately.

To access a list element, you can use its name, if available, or the position of the element in the list:

`List.ChiSquare` accesses the second element in the list above.

`List.[1n]` gives you the same result without using the name of the element.

You can use the ListToSeries function to convert a list with items of the same structure into a data series, data matrix or signal series.

Constructing Composite Data Structures

You can use the Signal function to create signal, signal series and space curve composite data structures from the basic data series and data matrix data structures:

`Signal(Amplitude, Time)` joins two data series with measurement values and the corresponding time values to form a signal. Amplitude and Time are then the components 121 of the signal.

Concatenating Data

The Concatenation operator appends data sets to each other:

```
DataSeries1 : DataSeries2
```

You can also append signals to each other. In this case, the X component of the second signal is shifted automatically so that it follows seamlessly after the first.

If you concatenate quantities, the unit of the argument on the right is adjusted to the one on the left.

Void Values

32- and 64-bit floating point values can have the value "void". These types of void values can represent, for instance, outliers in a data series.

Arithmetical operations, such as addition or multiplication with void values provide a void result.

All comparison operations return FALSE if one of the operands is void. The 'is equal to', 'is greater than or equal to' and 'is less than or equal to' comparison operators, however, return TRUE if both operands are void.

With the Analysis of data sets containing void values, these void values are ignored as much as possible. For example, in the case of the formation of a mean value, these values are not taken into consideration.

Due to their structure, some algorithms have the property of dispersing the void attribute. For example, if you calculate an FFT of a time signal with void values, then the result is practically completely void. You should not apply these types of procedures to data sets with void values.

You can display the value “void" in formulas or when editing data sets via a ?:

Constant	Value
?s	void 32-bit floating point value
? or ?L	void 64-bit floating point value
? V	void 64-bit floating point quantity with the unit V
'?.?.?' or '?/?/?'	void calendar time value
'?:?:?' or '?:?:?:?'	Void time span value

You can use Cursors 188 to make the values under the active cursor or the entire range between the cursors void. For example, you can use this option to mark outliers in a measured signal as void.

You can use the SignalCorrection analysis object* to remove void values from data sets or to interpolate them.

The functions SearchVoidValues *, ValuesInInterval * can be used to search for void values in a data set.

* Not available in FlexPro View.

Constants

Constants used in FPScript are subject to specific syntax rules, where, based on the notation used, FlexPro can recognize the data type 118 as well as the value.

Integers can be written in decimal, octal and hexadecimal notation. Octal numbers are written with a leading zero and can only contain the numbers 0 to 7. Hexadecimal numbers are writing with a leading combination 0x and can contain numbers 0 through 9 and A through F. Displaying hexadecimals and octals is interesting for digital data because there is a direct relationship between the numbers and the bits that represent them. For octal numbers, three bits are the equivalent of one digit, and with hexadecimal numbers, four bits are the equivalent of one digit. Please note that the only the notation is different for integers. Internally, all integers are saved in binary form.

FlexPro can process floating point values with 16-bit, 32-bit and 64-bit resolution. You can control the resolution by appending an 'S' or 's' for "short" or an 'L' or 'l' for "long". If you append the suffix 'N' or 'n', then this corresponds to 32-bit or 64-bit integers, depending on which format can store the constant.

Examples of 16-bit integers are:

`123s`	Decimal
`0427s`	Octal
`0x1afs`	Hexadecimal

Examples of 32-bit integers are:

`100000n, 12n`	Decimal
`01024341121, 0123`	Octal
`0x10A00FF, 0x1A, 0x1a`	Hexadecimal

Examples of 64-bit integers are:

`10000000000n, 12L`	Decimal
`071024341121, 0123L`	Octal
`0x1010A00FF, 0x1AL, 0x1aL`	Hexadecimal

Floating point values consist of a mantissa with whole and fractional digits and an optional exponent with a leading e or E. Floating point numbers can also have the value Void 270. This value is displayed as ?.

FlexPro can process floating point values with 32-bit and 64-bit resolution. 32-bit floating point values are accurate to approximately seven decimal places, and 64-bit floating point values have are accurate to approximately fifteen decimal places. You can control the resolution here as well by appending an 'S' or 's' for "short" or an 'L' or 'l' for "long". If you do not append a suffix, then this always corresponds to 64-bit floating point values, which means that the suffix 'L' can always be left off.

For the FPScript code, the decimal point (.) is always the decimal symbol, even if in the Regional and Language Options settings of the Windows Control Panel you have specified the comma (,), which is commonly used in some European countries.

Examples of 32-bit floating point values are:

```
1.23s, 1.234e-12s, 1e+13s, 0.1e12s, 5.s, ?s
```

FlexPro does not recognize whether the range of values or the resolution has been exceeded. If you would like to obtain 64-bit floating point values, you need to omit the 's'.

Examples of 64-bit floating point values are:

```
1.23, 1.234e-12, 1e+13, 0.1e12L, 5.L, ?, ?L
```

In general, spaces may not be used within numbers, not even between mantissa and exponent. If an overflow occurs during a calculation, FlexPro cannot recognize this! You have to ensure that the range of values of the data type is sufficient to perform the calculation.

Example:

`1000s * 1000s`	Generates an overflow because the result, 1,000,000, does not fit into a 16-bit integer! (Solution: `1000n * 1000n`)

Note How decimal numbers that you have not entered with a suffix or a decimal point are interpreted depends on the setting Interpret decimal numbers without a decimal point as floating point values on the General tab in the database properties dialog box. This option is enabled by default, which means that the numbers are interpreted as 64-bit floating point values. This setting prevents accidental rounding errors that may occur when using integer arithmetic. The integral division 1 / 2 results in 0 and not 0.5, for instance.

Complex numbers consist of a real and imaginary part, either in the form of a + bi or separated by commas and placed in parentheses. You can use all of the above notations for constants for the real part and the imaginary part. If this results in different data types for the real and imaginary parts, then these are adapted to each other before being combined into a complex number so that there is no loss of resolution. Within the FPScript, the comma (,) is always the list separator, even if in the Regional and Language Options settings of the Windows Control Panel you specified the semicolon (;), which is commonly used in some European countries.

Examples of complex numbers are:

```
(1.23s, 1.234e-12s) , (1e+13, 0x1), (5.L, ?), 1 + 2i, -3.2 - 7.6i, 5.2i, ?i,
2s + 5si
```

Strings are entered as text set off by quotation marks. There are specific control sequences available for special characters, such as a tab or a return character. These sequences all start with a '\' character.

Note If you want to use the '\' character itself, you must write it twice. The file path `C:\Data.txt` must be written as the string "`C:\\Data.txt`" for example.

Control sequence	Stands for
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\'	Single quotation mark
\"	Double quotation mark

Control sequence	Stands for
\\	Backslash
\oooooo	UNICODE characters in octal notation
\xhhhh	UNICODE characters in hexadecimal notation

Examples of strings are:

```
"A Text", "C:\\Data\\File.txt",", "Two\r\nlines"
```

The **Empty** data type can also be used as a constant to empty a variable used in a formula, for instance. To do this, just write `Empty` :

```
Result = Empty
```

Calendar time values consist of the date and time, with seconds in fractional digits being optional. Only three formats are allowed within FPScript. If you separate the date elements with '.', then the sequence will be Day.Month.Year, and if you use '/' as the separator, then this is interpreted as Month/Day/Year. If you use '-' as the separator, then it is interpreted as Year-Month-Day. The year can be entered as two or four digits. In the two-digit format, for values smaller than 70, 2000 is added, otherwise 1900 is added. For floating point data types, as well as for calendar times, the value is void if '?/?/?' appears. The elements for time are always specified in the format HH:MM:SS.ss. You can omit time elements starting from the right if they are zero. Constants in calendar time format must generally be written using single quotation marks:

Optionally, you can specify the time zone in which the calendar time value constant should be interpreted. To do this, append the time shift in hours from Universal Time Coordinated (UTC), e.g. "UTC+1". If you do not specify a time zone, the calendar time value constants are interpreted in local time, i.e. these are converted into the UTC format used for internal storage taking into account the time zone settings in Windows and, if applicable, daylight saving time.

`'23.10.2004 12:13:24.123'`	Calendar time value with a resolution of 1ms
`'10/23/2011 12:13:24.123'`	Calendar time value written in English notation with a resolution of 1ms
`'2004-10-23 12:13:24.123'`	Calendar time value arranged as Year-Month-Day

`'23.10.2004 12'`	Calendar time value 23.10.2004 at twelve noon
`'23.10.2004'`	Calendar time value 23.10.2004 at midnight
`'1.1.24'`	Calendar time value 1.1.2024 at midnight
`'1.1.98'`	Calendar time value 1.1.1998 at midnight
`'?/?/?'`	Void calendar time value
'1.2.2021 13:14'	1.2.2021, 13:14 in local time
'1.2.2021 13:14 UTC'	1.2.2021, 13:14 In Universal Time Coordinated (UTC)
'1.2.2021 13:14 UTC-2'	1.2.2021, 13:14 In UTC minus 2 hours
'1.2.2021 13:14 UTC+2:30' or '1.2.2021 13:14 UTC+2.5'	1.2.2021, 13:14 UTC plus 2 hours, 30 minutes

Time span values consist of the elements day, hour, minute, second. You can specify seconds using fractional digits as an option. The elements are specified in the format D:HH:MM:SS.ss. You do not have to specify all elements. If you specify two elements, then this is interpreted as HH:MM. If you specify three elements, then they are interpreted as HH:MM:SS, and four elements are then interpreted as D:HH:MM:SS. For output, FlexPro uses three or four elements as the standard. The day is omitted if the time span is less than 24 hours. Constants in calendar time format must generally be written using single quotation marks:

`'12:13:24.123'`	Time span value with a resolution of 1 ms
`'13:24'`	Time span value of 13 hours and 24 minutes

`'2:01:30:00'` or `'49:30'`	Time span value of 2 days, 1 hour and 30 minutes
`'?:?:?'`	Void time span value

Note: The calendar time and time span values have second as the implicit unit. The unit symbol, however, is not displayed.

Pre-defined constants

FPScript recognizes a series of pre-defined constants. The names of the constants cannot be used as object names or as variable names. These are mathematical constants PI = 3.141592654L and E = 2.718281828L as well as a series of constants that have pre-defined meaning as the parameters for functions.

Operators

In FPScript you can use a variety of operators, e.g. for arithmetical and logical operations, comparison and concatenation operations. You can use the operators to calculate individual values or entire data sets, data matrices, signals, signal series or space curves.

Operator Precedence

If several operators occur in an expression, each part is evaluated and resolved in a predefined order. This order is described as operator precedence. You can override the operator precedence using parentheses to force the evaluation of parts of an expression before other components. Operations in parentheses are always executed before the operations that are not in parentheses. However, standard operator precedence also applies within the parentheses.

If an expression contains operators of different categories, then these are evaluated in the order specified below. All comparison operators have the same priority, i.e. they are evaluated from left to right in their order of appearance. Arithmetical and logical operators are evaluated in the following order:

Description	Symbol
Conversion	
Convert to Boolean Value	Boolean
Convert to 16-Bit Integer	Integer16

Description	Symbol
Convert to 32-Bit Integer	Integer32
Convert to 64-Bit Integer	Integer64
Convert to 32-Bit Floating Point Value	FloatingPoint32
Convert to 64-Bit Floating Point Value	FloatingPoint64
Convert to Complex 16-Bit Integer	ComplexInteger16
Convert to Complex 32-Bit Integer	ComplexInteger32
Convert to Complex 64-Bit Integer	ComplexInteger64
Convert to Complex 32-Bit Floating Point Value	ComplexFloatingPoint32
Convert to Complex 64-Bit Floating Point Value	ComplexFloatingPoint64
Convert to String	String
Convert to Calendar Time	CalendarTime
Convert to Time Span	TimeSpan
Convert unit	Unit
Convert quantity to value	Value
Arithmetic	
Unary Negation	-
Exponentiation	^
Multiplication	*
Division	/
Division Remainder	Mod
Addition	+
Subtraction	-
Operational Sign	#
Complex Numbers	
Form Real Component	Real
Form Imaginary Component	Imag
Form Conjugate-Complex Number	*
Comparison *	
Less Than	<
Greater Than	>

Description	Symbol
Less Than or Equal To	<=
Greater Than or Equal To	>=
Equality	==
Inequality	<>
Logical	
Binary Negation	~
Binary Conjunction	&
Binary Exclusion	XOr
Binary Disjunction	|
Logical Negation *	Not
Logical Conjunction *	And
Logical Disjunction *	Or
Data Organization	
Concatenation	:
Multiply	#
Form Complex Number	(,)
Bundle	{ , }
Form Data Series	(, ,)
Form List	[,]
Data Access	
Component	.X .Y .Z
Property	.
List Element	.[ ]
Index	[,] [[,]]
Indirection *	$$

If multiplication and division occur in an expression, each operation is executed from left to right in the order of appearance. The same procedure applies when addition and subtraction or concatenation and multiplying occur in an expression.

Associativity

The exponentiation operator is the only operator that is right-associative. This means that several exponentiations are processed in sequence from right to left. All other operators are processed from left to right.

Processing Logical Expressions

A logical conjunction (Boolean AND operation), or sequences from it are only evaluated until the result is defined, i.e. until the first expression results in FALSE. The same applies to the logical disjunction (OR operation). These are evaluated until the first expression results in TRUE. This does not apply to the bitwise logical operations.

In the following example, `Test(DataSet2)` is only called when `Test(DataSet1)` results in true. If `Test(DataSet1)` results in FALSE, then the final result of the Boolean AND operation is defined as FALSE.

```
If Test(DataSet1) And Test(DataSet2) Then
    ...
End
```

* This operator and all operators of this category are not available in FlexPro View.

Accessing Object Properties

The objects saved in the project database, such as documents or diagrams, have a series of properties. For example, each object has a name or a creation date; data sets have a physical unit, etc. You can use FPScript for read access to these types of object properties. You can, for instance, query the unit name of a data set or the current cursor position in a diagram. To do this, use the following syntax:

[*Object*]**.***Property*[**(***Parameter 1***,**...**,** *Parameter n***)**]

Object is the path name of the object and *Property* is the name of the property that you want to access. If the object is not a data object, i.e. a data set or a formula, then you have to append the file name extension 108 describing the object type to the object name. If you do not specify an object name, then the formula containing the FPScript code is accessed. To access certain objects, which depend on the context of the processed FPScript formula, use the This, ThisFPObject or ThisObject keywords. In the case of embedded FPScript you have to use the `This` keyword.

`Diagram.2D.Name` supplies the name of the 2D diagram Diagram as a string. Name in this case is the name of the property to be read and .2D is the file extension for the 2D diagram.

`.\.FullName` provides the absolute path name for the folder where the FPScript formula is located.

`\.Comments` provides the comments of the project database root folder.

`.UnitY` provides the physical unit of the Y component of the formula containing this FPScript expression.

You can also access object properties using a variable if you have already assigned an object reference to it:

```
Dim Obj = MyDataset As Object
Obj.CommentsX
```

Write-access to object properties is greatly reduced and only permitted for the header information of the formula containing the FPScript code. It is, however, preferred that you handle the corresponding entries in the Properties dialog box of the formula for these properties instead of writing these in the FPScript code. During each update process, formulas that use this type of write access are not considered up-to-date and thus must be recalculated each time. It makes more sense to have write-access to object properties in the embedded FPScript 380.

Instead of directly specifying the object path name or the property name, you can use the indirection operator, which uses a string containing the name. The parameter list is only required for properties that require additional parameter settings. To access the cursor position in a diagram within a worksheet, for example, you have to specify the number of the plane and the number of the cursor.

FPScript allows read access to all object properties that are defined in the FlexPro automation object model 562. It is used very frequently in embedded FPScript formulas 380, which are used, for instance, to label the axes in a diagram. Within FPScript formulas that you use for data analysis, such access is mainly targeting attributes and parameters 107 of objects as well as cursors and markers in diagrams. You can access the FPScript expressions required for this through the Object Properties Wizard of the Formula Editor.

Accessing Header Information

FlexPro can append a data object Header Information 126 reference to an FPScript value that originates from this data object. You can therefore access the data object's attributes via the value. The header information is the properties of the ValueObject object, including the parameter list. If an FPScript value is passed as an argument or as a return value from a value in a different formula, then the value can be used to access the header information without an object reference. FlexPro uses this header information assigned to the data for such tasks as labeling axes in a diagram.

When assigning the header information of a formula to its result value, FlexPro takes into account the Header Information setting on the Results tab of the formula. There you can set whether the header information should only be added or whether existing header information should be replaced or removed.

FPScript functions which only modify the data passed as an argument, such as filter functions, pass the header information from the argument to your result. The same applies to index operations for selecting a data component or a list element.

The following code uses the variable x to access the X component comments of the data set containing the x value:

```
Dim x = \Data\DataSet

x.CommentsX
```

You can also have write-access to an attribute:

```
x.CommentsX = "Time"

x.UpperRangeLimitX = 1.3
```

Write access in this case is not to the data object upon which it is based (\Data\DataSet in the example above). Instead, x is assigned a local copy for the property to be changed. This copy is then set to the assigned value. You can delete the local copy by deleting the associated property:

```
x.CommentsX = ""

x.UpperRangeLimitX = ?
```

The associated attributes will then be removed from the data set at the next read access attempt.

Parameters

The complete parameter list is always provided as a local copy. If when using write access a name is specified that does not have any parameters, the name will be created automatically.

```
x.Parameters("MyParam") = 1.3
```

Apply the Empty value to delete a parameter again:

```
x.Parameters("MyParam") = Empty
```

You can verify whether a parameter is in the list using the Index property.

```
x.Parameters.Index("MyParam")
```

Read access to a parameter only results in its value. You need to read out and assign the unit separately:

```
ChangeUnit(x.Parameters("MyParam"), x.Parameters("MyParam").Unit)
```

Calculations

Only read access is available for calculations. Read access to a calculation only results in its value. You need to read out and assign the unit separately:

```
ChangeUnit(x.Calculations("MyCalculation"), x.Calculations("MyCalculation").Unit)
```

You can verify whether a calculation is in the list using the Index property.

```
x.Calculations.Index("MyCalculation")
```

Lists

Forms that provide a list with multiple data sets as the result are a special feature. If the formula bundles only data sets into a list, the list elements obtain the header information of the data sets from which they originate:

```
Dim x = [Signal1, Signal2]
```

`x.[0].Name` results in "Signal1"

Note If you return this type of list as a formula result and do not want the header information to be overwritten by that of the formula, you will have to choose the following in the Header Information list box on the Results tab of the formula Properties dialog box: Automatic, Only assign, if not already present or Never assign.

If the formula calculates a result and the result is returned as a list, you will have to set how to name list elements under List Element Names on the Results tab of the formula. The list element names of the calculated source data are often built into the result's list element names and therefore are known to FlexPro. This is done by assigning the source data of the local variable SourceData. FlexPro takes this into account when assigning element names after calculating the formula.

```
Dim SourceData = [Signal1, Signal2]
Integral(SourceData)
```

`x.[0].Name` results in "Signal1Integral", if on formula the Results tab, the option Automatic is set and the formula itself is called "Integral".

`x.[0].CommentsX` If on the Results tab of the formula the option Always assign is set, the expression result provides the X comments of the formula containing the FPScript code. Otherwise, the result provides the X comments of the Signal1 data set, since the Integral function passes this in its result.

If you return a list as the result of a formula, you can assign the header information for each list element individually, e.g. because the individual elements are to be formatted differently:

```
Dim List = [<Date> '1.3.2024', <'Value'> 1.23 ]
List.Date.FormatY = "%(%d.%m.%Y)"
List.'Value'.FormatY = "%.3f"
return List
```

Components

If an FPScript variable contains a component of a composite data set, then the component is used as the template for the properties for which a component can be specified optionally as the argument if you omit the argument:

```
Dim x = DataSet.X
```

`x.Quantity` corresponds to `DataSet.Quantity(fpDataComponentX)`

It is therefore easy to access the attributes of each relevant component.

Legacy Functions in FPScript

The number of available FPScript functions is constantly being expanded and reorganized. Consequently, FPScript functions also become outdated because their functionality ends up being covered by other functions. For the purpose of compatibility, these legacy functions are still supported by FlexPro, but are no longer offered as an option in the Function Wizard and are also no longer mentioned in the documentation. These functions are marked by FlexPro with a '_' sign before the name of the function.

The following is a list of all legacy functions and their successors:

Function	Last Version Used	Replaced By	Remarks
_BartlettWindow	FlexPro 6	DataWindow(s, WIN_BARTLETT)	
_BlackmanWindow	FlexPro 6	DataWindow(s, WIN_COS3BLACKMAN)	
_Bursts	FlexPro 7	Bursts	Algorithm has been changed.
_ConnesWindow	FlexPro 6	DataWindow(s, WIN_BISQUARE)	
_CosineWindow	FlexPro 6	DataWindow(s, WIN_SINE)	
Date	FlexPro 2019	DateTime	The function has been renamed.
Duration	FlexPro 2019	Time	The function has been renamed.
_FFT	FlexPro 6	FFTn(s) / NumberOfRows(s) / 2 FourierSpectrum(s, SPECTRUM_AMPLITUDE, WIN_RECTANGULAR)	FFTn is no longer limited to values based on the power of 2. Please note, however, that FFTn provides a non-normalized complex magnitude spectrum, while FFT calculates complex amplitudes.
_FlatTopWindow	FlexPro 6	DataWindow(s, WIN_FLATTOP)	
_HammingWindow	FlexPro 6	DataWindow(s, WIN_HAMMING)	
_HanningWindow	FlexPro 6	DataWindow(s, WIN_HANNING)	
_IIRBandFilter	FlexPro 7.0.13	IIRFilter and Filter	
_IIRFilter	FlexPro 7.0.13	IIRFilter and Filter	
_IRFFT	FlexPro 6	IRFFTn	IRFFT expects an amplitude spectrum and

Function	Last Version Used	Replaced By	Remarks
			IRFFTn expects a magnitude spectrum as the argument.
_KaiserBesselWindow	FlexPro 6	DataWindow(s, WIN_KAISERBESSEL)	
_LevelCrossing	FlexPro 4	LevelCrossings	
_LevelCrossings	FlexPro 7	LevelCrossings	Algorithm has been changed.
_LocalExtrema	FlexPro 4	Extrema	
_LocalMaxima	FlexPro 4	Extrema	
_LocalMinima	FlexPro 4	Extrema	
_NegativePeaks	FlexPro 4	Extrema	
_NextBurst	FlexPro 7	NextBurst	Algorithm has been changed.
_NextLevelCrossing	FlexPro 7	NextLevelCrossing	Algorithm has been changed.
_NextSlope	FlexPro 7	NextSlope	Algorithm has been changed.
_NumberOfColumns	FlexPro 2019	NumberOfColumns	The list processing method has changed.
_NumberOfRows	FlexPro 2019	NumberOfRows	The list processing method has changed.
_OrderTracking	FlexPro 2019	RevolutionSyncSampling	
_ParSplineX	FlexPro 3	ParametricSpline	
_ParSplineY	FlexPro 3	ParametricSpline	
_PositivePeaks	FlexPro 4	Extrema	
_Random	FlexPro 6	Noise	
_Rectangle	FlexPro 6	Square	
_ReadASCII	FlexPro 3	ReadTextData	
_ReadORP_ORMFileX	FlexPro 5	ReadORP_ORMFile	

Function	Last Version Used	Replaced By	Remarks
_ReadORP_ORMFileY	FlexPro 5	ReadORP_ORMFile	
_SigMax	FlexPro 3	Maximum	
ReadTextFile	FlexPro 10	ReadTextData	ReadTextFile was not renamed to _ReadTextFile, since the function is still used by some import filters.
_SigMin	FlexPro 3	Minimum	
_Slopes	FlexPro 7	Slopes	Algorithm has been changed.
Time	FlexPro 2019	Date(CurrentDate()) + Time(h, m, s)	The functionality of the function has changed. Older FPScript code is accordingly corrected automatically.
TimeFilter	FlexPro 2019	TimePeriods	The function has been renamed.
_Triangle	FlexPro 6	Sawtooth	
_WelchWindow	FlexPro 6	DataWindow(s, WIN_WELCH)	
_WindowFFT	FlexPro 6	STFTSpectrum(s, SPECTRUM_AMPLITUDE, ...)	
XScale	FlexPro 2019	XOffsetScale	The function has been renamed.
_ZeroCrossings	FlexPro 4	LevelCrossings	

Python

Python is an easy-to-learn, open-source interpreter language for which a large number of libraries are available on topics such as artificial intelligence, statistics, numerics and much more. An introduction to the Python programming language is not the subject of this documentation. You can find these at https://www.python.org/doc/, for example.

Python can be extended very flexibly, which enables elegant integration into FlexPro. The array data types required for processing measurement data are provided by the well-known Python library NumPy. These are the basis for data exchange with FlexPro, which is why NumPy is also installed when FlexPro is installed. You can find information on NumPy at https://numpy.org/doc/stable/.

The flexpro Module

When Python is initialized in FlexPro, the Python module flexpro is registered, which is the interface to FlexPro.

The Python classes flexpro.DataObject and flexpro.Object represent FlexPro data objects and other FlexPro objects so that you can access their properties. You create objects of these classes with the flexpro.object() function, to which you pass the path and the name in the object database of the object you want to access as an argument.

`o = flexpro.object(r'\Data\Dataset1')` creates a DataObject that represents the dataset "\Data\Dataset1".

The variable this is already initialized in every Python formula and contains the DataObject that represents the Python formula itself.

`this.name` returns the name of the Python formula as a string.

`this.data = np.linspace(0, 10, 100)` returns a linear data series as the result of the formula.

Note In FPScript, each statement returns a result that is automatically assigned to the formula as the result. This is not the case in Python and you must use an explicit assignment this.data = ... This also applies to intermediate results that you want to see as the result of the formula in the Watch window during debugging with the debugger.

FlexPro data differs from Python or NumPy data in that it can have a unit and can have composite data structures with Y, X and Z components. Lists are also available in Python, but these do not support named list elements. The flexpro module therefore provides the flexpro.Data type, which can represent FlexPro data with all its properties.

You can create a flexpro.Data object with simple or composite FlexPro data using the flexpro.Data() function.

`flexpro.Data(80.3, "dB")` creates a scalar floating point value with the unit "dB".

This example creates a sine signal with unitized Y and X components and returns it as the result of the formula:

```
x = np.linspace(0, 10, 100)
y = np.sin(x)
this.data = flexpro.Data(y, "V", x, "s")
```

If you want to omit the units, e.g. because you have already entered them in the formula's properties dialog box, you must name the arguments:

```
this.data = flexpro.Data(value_y = y, value_x = x)
```

You can create a FlexPro list with two named list elements as follows, for example:

```
list = flexpro.named_list("Signal1", s1, "Signal2", s2)
```

Appending to such a list is also supported:

```
list += flexpro.named_list("Signal3", s3)
```

You can create lists without element names either with the Python operator [...] or with the flexpro.list() function. The latter is somewhat more efficient because no conversion to the FlexPro data format is necessary when transferring to FlexPro.

The data property of the flexpro.DataObject class returns the data of the data object as a flexpro.Data object.

```
data = flexpro.object(r'\Data Set').data
```

If you want to access the data directly, use the call function:

```
data = flexpro.call(r'Data Set')
```

You call an FPScript or Python function by passing the arguments to call:

```
data = flexpro.call(r'\MyFunction', flexpro.call(r'\Data Set'), flexpro.Data(5, 'V'))
```

The flexPro.Data class has the properties value and unit. value provides the numerical data as a Python or NumPy array data type and unit provides the unit as a string.

`flexpro.Data(1.3, "V").value` returns 1.3 as a floating point value.

`flexpro.Data(1.3, "V").unit` returns "V" as a string.

For a composite data structure, first select the component:

```
signal = flexpro.call(r'\Data\Signal1')
```

`signal.y` returns a flexpro.Data object that represents the Y component with unit.

`signal.y.value` returns the data of the Y component as a NumPy array.

`signal.y.unit` returns "V" as a string.

If a list is available, first select the element:

`list["Signal1"]` returns a flexpro.Data object that represents the data of the list element "Signal1".

You must use a numerical index for unnamed lists or non-unique names:

```
list[0]
```

FlexPro attaches the header information of a data object to a flexpro.Data object that originates from a data object. These are essentially the properties that you can specify on the General tab of the data object.

You can have read and write access to these properties:

```
data = flexpro.call(r'\Signal')
```

`data.unit_y` returns the Y-unit of the data object from which the value originates, i.e. from "Signal".

`data.comments_x = "Time"` Assigns an X-comment to the data. However, write access is not to the "Signal" data object from which `data` originates, but to a local copy of this attribute assigned to the Data object.

5.4 FPScript and Python Functions

FlexPro offers you the option of programming your own functions in FPScript or Python. You can create these as you would regular formulas; however, you need to use the Arguments statement in the first code line to declare arguments for the function. This turns the formula into a function which must be supplied with arguments when called. You can also save these types of functions in template databases, thus expanding the number of FPScript functions.

Python is part of FlexPro in the Professional Edition.

An example

a) FPScript

An FPScript function `Sum` would look like this, for example:

```
Arguments a, b
a + b
```

This function can now be called in a different formula:

```
Sum(1 V, 2 mV)
```

or

```
Sum(DataSet1, DataSet2)
```

The arguments `a` and `b` serve as formal parameters to represent the arguments passed. You can use these just like local variables 266.

You can omit any items in the argument list when calling the function. These items then have an empty data type:

`Sum( , 2)` is therefore the equivalent to `Sum(Empty, 2)` and `Sum(1)` is equivalent to `Sum(1, Empty)`.

Please note that in the function `Sum` no Return statement is necessary. The result of the last statement in a formula is automatically used as the result of the formula.

b) Python

The same example in Python looks slightly different. In Python, you must explicitly assign the calculation result to the result value of the function: The arguments are passed as flexpro.Data objects with value and unit:

```
arguments a, b
this.data = a.value + b.value
```

If you want to pass the result with a unit, the code looks like this:

```
arguments a, b
this.data = flexpro.Data(a.value + b.value, a.unit)
```

An FPScript or Python function is called in a Python formula as follows:

```
flexrpro.call('Sum', flexpro.Data(1, 'V'), flexpro.Data(2, 'V'))
```

Please note that Python does not check or adjust the unit. If, for example, `a` has the unit V and `b` has the unit mV, the calculated result is incorrect.

Saving FPScript or Python Functions as a Template

If you save an FPScript function in a template database, you can use it in your project databases as you would a built-in function. To save a function, use a wizard in which you can specify a comment for each argument and define the permitted data types and structures. Functions saved in this way

- are displayed in the wizard for inserting a function into a formula in the Customized category,
- are supported by the FPScript Editor's Assistance feature,
- support optional arguments with default values, and
- when called, automatically check arguments passed for permitted data types and data structures.

Sharing FPScript Functions

FlexPro Professional and FlexPro Developer Suite support shared template databases. You can share FPScript functions that you save in this type of template database with your colleagues. Use the shared template databases to create function libraries for your department or company.

Working with FPScript and Python Functions

Creating an FPScript or Python Function

1. In the Folders window, select the folder in which you want to place the new formula.
2. Click Insert[Data] > Formula > Function.

A new function with one argument `x` is added. You can rename the argument and add additional arguments using commas.

Saving an FPScript Function as a Template

1. Select the FPScript function from the object list and click Home[Selected Objects] > Save As Template.
2. Follow the steps in the Template Wizard and use the dialog box Help if you require additional assistance.

Calling an FPScript Function

- Use an FPScript function as you would a built-in function by entering the name, followed by the argument list, in the FPScript code, as in the following example:
  `Normalize(Signal)`
- If the function to be called was not saved in a template database and is also not to be found in the same folder as the formula from which you want to call the function, you have to specify the path name, such as follows:
  `\MyFunctions\Normalize(Signal)`

Applying FPScript Functions to Data Sets

Often, identical calculations have to be performed for a variety of data sets. One way of automating this process is to save the data sets in different subfolders and to activate 230 the folder from which the data for an analysis is to be taken.

The second option, which is described here, is to pack the calculation rule into an FPScript function and then to use this as a template for generating new formulas.

FPScript offers you the option of programming formulas with arguments. You can then use these formulas like functions. A Normalized formula, which normalizes your argument to a range of values from 0 to 1, looks like this, for example:

```
Arguments Data
Data / Maximum(Data)
```

You can now use this function in other formulas. For instance, you can add a SignalNormalized formula containing the following code:

```
Normalized(Signal)
```

It is very easy to create these types of formulas with FlexPro. You only have to click on the data sets to which you would like to apply the function, use your mouse to select them from the object list and drag them onto the function in the object list. If you drag just one data set to the function, FlexPro then adds a new formula that applies the function to this data set. When you drag multiple data sets to the function, a dialog box appears where you can choose whether to create a formula like this for each data set, or whether to create only one formula that bundles all data sets into a list and with that calls the function.

In the example above, drag the Signal data set to the Normalized function in the object list. FlexPro will then automatically add a new formula named SignalNormalized which contains the code specified above. You can then use this new formula for further calculations or for displaying in documents.

You can also use functions with several arguments. In this case, the first argument used is the data set that you dragged onto the function with the mouse. You can specify additional arguments in a dialog box.

5.5 Analysis Objects

FlexPro offers you special analysis objects for all common analyses. These are FPScript formulas that FlexPro automatically creates. You can set the parameters for the analysis object using its Properties dialog box. The formula will then be adjusted automatically. Analysis objects thus offer automated analyses without any programming.

In addition to simple parameterization, many analysis objects can automatically generate the physical units and comments for their results. You can transform analysis objects into data sets or formulas at any time.

In the Reference section, you will find an alphabetical list of the available analysis objects.

Note In FlexPro View only a small set of analysis objects for basic statistics is available.

External Parameterization of Analysis Objects

In most cases, you enter numerical parameters, such as the smoothing width of a signal smoothing, directly as numerical values in the properties dialog box of the analysis object. Instead, you can also specify a reference to an external data set or a formula that is to provide the parameter. This is useful if, for example, you want to set a parameter for several analysis objects in one place or calculate the parameter. Parameters for which this is possible are displayed in the combination list field typical for data set references instead of in a simple text field.

Analyzing several data sets

For multichannel analyses, the same calculations are used for multiple channels. In this case, FlexPro can create one analysis object for all output data sets or an analysis object for each output dataset.

If you create only one analysis object, it will provide all calculated data sets as a list. The advantage of this is that you only have to parameterize one object and an analysis with a better structure. This option is also particularly useful if all channels need to be processed in the same format, such as when filtering, integrating and displaying 16 channels.

If you decide to create one analysis object per output data set, you can parameterize each analysis individually, and it will be easier to select individual results for further processing.

Working with Analysis Objects

Creating an Analysis Object

To create an analysis for a data set or a range from a data set:

1. In the Folders window, select the folder where you want the new analysis object to be placed.
2. Select the data to which the analysis is to be applied. You can select a data set or formula in the object list or a cell range in the data view or data set window. This step is optional. If you do not select any data, you have to specify the data later in the Properties dialog box 107 of the analysis object.
3. Under Insert[Analyses] choose the analysis from one of the categories provided.
4. In the Properties dialog box that appears, set the parameters for the new analysis object.

Note: Some analysis procedures require more than one argument. In this case, you have to select multiple data sets in the object list or specify additional data sets Properties dialog box of the analysis object.

To create the same analysis for multiple data sets:

1. In the Folders window, select the folder in which you want to place the new analysis objects.
2. In the object list select the data sets or formulas to which the analysis is to be applied.
3. Under Insert[Analyses] choose from one of the categories provided an analysis procedure that calculates only one data set.
4. In the Analyzing several data sets dialog that appears, select whether you want to create one analysis object that calculates all data sets or an analysis object

for each data set. Use the Help in the dialog box to learn more about the advantages of each option.

Note You can also choose the option you want without viewing the dialog box: Hold down the SHIFT key while you select the analysis from one of the menus in Insert[Analyses] to create an analysis object for each data set. Alternatively, press and hold the CTRL key to create a single analysis object.

5. If you created only one analysis object, a Properties window where you can set the parameters for it will appear. Otherwise, the new analysis objects will be selected in the object list. You can set parameters for all objects at once in the FlexPro Properties window.

Note Not all analysis objects are available in FlexPro View.

Editing an Analysis Object

Editing a Single Analysis Object

1. Double-click on the object in the object list to display its Properties dialog box 107.
2. On the Options tab you can adjust all settings for the analysis procedure.
3. Specify the data sets you want to analyze on the Data tab.

Editing Several Similar Analysis Objects

1. Select the analysis object that you would like to edit from the object list.
2. Adjust the desired settings in the FlexPro Properties window.

5.6 Analysis Options

Human Body Vibrations Option

Human body vibrations are mechanical vibrations such as shaking, shocks or jolts. A distinction is made between whole-body vibrations and hand-transmitted vibrations. Human body vibrations occur when a person comes in contact with mechanical vibrations, such as when using vehicles or machines or mechanically operated tools.

Musculoskeletal and circulatory disorders may be caused by vibration. In addition, vibrations negatively impact the well-being and abilities of the person exposed to them.

The Whole-Body Vibration analysis object is used to analyze mechanical vibrations affecting the human body. Whole-body vibrations are caused by vibrations transmitted through the seat or the feet by workplace machines and vehicles.

The Hand-Transmitted Vibration analysis object is used to analyze hand-transmitted vibrations that are transferred from vibrating tools through the palms and fingers to the hands and arms.

The results of the two analysis objects are used to evaluate the risks caused by vibration. The directives and standards include exposure limit values that can be compared to the assessed results. In this way you can determine whether safety measures for eliminating or reducing the effects of vibration are necessary.

Observed Standards

Standard	Description
ISO 2631-1:1997	Mechanical vibration and shock - Evaluation of human exposure to whole-body vibration - Part 1: General requirements
ISO 2631-2:2003	Mechanical vibration and shock - Evaluation of human exposure to whole-body vibration - Part 2: Vibration in buildings (1 Hz to 80 Hz)
ISO 2631-4:2001	Mechanical vibration and shock - Evaluation of human exposure to whole-body vibration - Part 4: Guidelines for the evaluation of the effects of vibration and rotational motion on passenger and crew comfort on fixed-guideway transport systems
ISO 5349-1:2001	Mechanical vibration - Measurement and evaluation of human exposure to hand-transmitted vibration - Part 1:

Standard	Description
Directive 2002/44/EC	General requirements on the minimum health and safety requirements regarding the exposure of workers to the Risks arising from physical agents (vibration)

Analysis Objects

Whole-Body Vibration

Hand-transmitted vibration

FPScript Functions

VibrationFrequencyWeighting

Statistics Option

The Statistics Option offers you a variety of statistical tests and the possibility of calculating theoretical distributions. Unlike descriptive statistics, which permits a description of the data material via statistical quantities, the testing and estimating procedures of inductive statistics offer the possibility, based on samples, of focusing on the population from which the samples originate.

Inductive statistics are increasingly gaining importance, particularly in the fields of statistical process control (SPC).

References

Hartung, Joachim (1993). *Statistik (Statistics), 9th Edition.* Oldenbourg Verlag GmbH, Munich. ISBN 3-486-22055-1.

Analysis Objects

ANOVA

Goodness-of-Fit Test

Outlier Correction

Outlier Test

Variance Test

Distribution

Confidence Interval

FPScript Functions

ANOVA

BartlettTest

ChiSquareTest

ConfidenceInterval

Correlation

Covariance

DavidHartleyPearsonTest

Distribution

GrubbsBeckTest

KolmogorovSmirnovTest

Digital Filters Option

The FlexPro digital filters option offers algorithms for designing IIR and FIR filters as well as filters for smoothing data. The Analysis Wizard makes it possible to design various IIR and FIR filters based on specifications. After creation, the results can be stored directly as objects and documents in FlexPro's Object List.

Filter Design

FlexPro offers three procedures for designing filters: two options for the FIR filter design (FIR: Finite Impulse Response) and one for designing IIR filters (IIR: These procedures are presented in the Digital Filters Tutorial. The filter coefficients can be calculated and the designed filter can be applied directly to the input signal. Phase correction is also possible here.

FIR Filter

The FIR filter design using the window method is probably the most popular FIR filter design option. This design technique is based on direct approximation of the required frequency response of the discrete-time system. In this case, the filter coefficients are determined from the impulse response of the filter to be implemented. In principle, this method should be accurate, but since the impulse response needs to have a finite length, large deviations are more or less the result, which are disruptive particularly in the achievable stopband attenuation and slope steepness.

For the FIR Filter Analysis Object (Window Method) various window functions available (Rectangular, Bartlett, Hamming, Generalized Hamming, Hanning, Blackman, Kaiser, Chebyshev) with which both the stopband attenuation and the slope steepness can be influenced. The Kaiser window and the Chebyshev window have a special role in this case, since for both windows the attenuation behavior and slope steepness can be influenced through adjustable parameters.

The disadvantage of designing FIR filters using the window method is that the approximation error cannot be influenced in different frequency ranges. Therefore it is often better when designing filters to use the minimax strategy (minimize the maximum error) or an error criterion with frequency weighting. This results in the "best" filter that can be achieved for a given specification. The Parks-McClellan method provides the solution for the FIR filter using the Remez Exchange algorithm (Equiripple FIR) in an iteration process. Since filters can be designed using this approximation method with a constant ripple both in the passband and stopband, the filter is also known as the Equiripple filter. This algorithm is an optimization algorithm is based on Chebyshev polynomials and produces a minimum number of filter coefficients. For the required filter, an error function is formed from a linear combination of cosine functions and minimized by an efficient optimization procedure.

Using the FIR Filter Analysis Object (Equiripple Method), it is possible to design freely definable multiband filters in addition to low pass, high pass, bandpass and stopband filters. For this purpose, bands can be specified for which the cut-off frequencies, related gains and approximation error are specified.

IIR Filter

Using the IIR Filter Analysis Object you can design a filter that includes internal feedback (recursive filter). Different filter characteristics are available for this: Bessel filter, Butterworth filter, Chebyshev filter, Inverse Chebyshev filter (Chebyshev II filter) and Elliptic filter (Cauer filter). Low pass, high pass, bandpass and bandstop are also design options.

CFC Filter

CFC is short for Channel Frequency Class. This is a 4-pole phaseless Butterworth filter. The CFC filter is used in particular for crash tests and is described in more detail in the ISO 6487 and SAE J211 standards (CFC Filter Calculation).

SavitzkyGolay smoothing filter

This filter uses polynomial smoothing based on Savitzky and Golay. The data are weighted by applying filter coefficients (SavitzkyGolay Algorithm). Unlike a moving average filter, which reduces the height of the local maximum (peak) while simultaneously spreading the peak, the SavitzkyGolayFilter leaves the local maximum virtually unchanged. Use of this filter is therefore preferred in spectroscopy, for example.

LOESS and LOWESS filters

LOESS and LOWESS are very popular smoothing methods that use a locally weighted regression function. This method uses a weighting function with the effect that the influence of a neighboring value on the smoothed value at a certain position decreases with their distance to that position. Outliers are weighted lower compared to other methods. The important issue is selecting the smoothing width that renders the number of observed values used in calculating a point. In addition, two types of weighting are used: proximity weighting and robust weighting (LOESS/LOWESS Algorithm). LoessFilter uses a quadratic weighting function and LowessFilter uses a linear weighting function.

Counting Procedures Option

As well as harmonic analysis by means of Fourier transform, counting procedures have proved to be an important tool for examining signals, in particular for load-time functions. The counting procedure is based on the search for specific events in the load-time function, e.g. exceeding a certain load level, or a load alternation of a certain amplitude. For this purpose, the range of values of the load-time function is divided into discrete ranges known as classes. Each event found is assigned to a class and counted in this class.

The result shows a frequency for each class. The Counting Procedures Option provides you with a broad spectrum of counting procedures. The bases for the implementation are the DIN 45667 standard and the more recent Rainflow Procedure. The DIN 45667 standard dates back to 1969 and is geared toward the technical counting devices available at the time. The Rainflow procedure has mainly replaced the counting procedures described in DIN 45667. Therefore, only those DIN 45667 procedures were implemented for which the Rainflow procedure does not offer an equivalent. In addition, the Rainflow procedure is better at addressing fatigue strength testing requirements. The Rainflow procedure provides better results for this purpose.

References

[1] DIN 45667 (1970). *Klassierverfahren für das Erfassen regelloser Schwingungen (Counting procedures for determining random oscillations)*. Beuth Verlag

[2] de Jonge, J.B. (1980). *Counting Methods for the Analysis of Load Time Histories.* NLR Memorandum SB-80-106 U

[3] Fraunhofer-Institut für Betriebsfestigkeit LBF (Fraunhofer-Institute for Structural Durability LBF), Darmstadt. *Hinweise zur Anwendung von Zählverfahren im Rahmen von Betriebsfestigkeitsuntersuchungen (Tips on using counting procedures as part of structural durability testing).* www.simtop.fhg.de/cgi-bin/rmcat?1000kl03

[4] ASTM E1049-85(2017)..*Standard Practices for Cycle Counting in Fatigue Analysis.*STM International, West Conshohocken, PA, 2017

Order Tracking Option

Order tracking is the analysis of the vibrations or noise of rotating machines. This is done by normalizing frequencies in the spectrum to their respective speed, i.e. presented as orders. The order is a multiple of the speed.

In FlexPro 2021, order tracking was changed to a completely new procedure. The core methods of the option are the Revolution Synchronous Order Tracking analysis object and the Order Filter analysis object. In both methods, the underlying signal sampled over time is first transformed into the revolution domain. This enables effective order tracking performance, either by ordinary spectral analysis in the revolution domain (the Revolution Synchronous Order Tracking analysis object), or by means of bandpass filtering of individual orders in the revolution domain (the Order Filter analysis object). The transformation into the revolution domain can also be performed independently with the help of the Revolution Synchronous Sampling analysis object. In addition, the option includes the Harmonic Filter analysis object, which can be used to remove or extract harmonic components of a selected order from time signals.

Alternatively, the Order Tracking analysis object (deprecated, removed from Gallery in FlexPro 2021) can be used. With the help of this analysis object, the vibration signals measured at a certain speed are directly subjected to a Fourier transform (FFT). This analysis object provides similar results to the analysis objects Revolution Synchronous Order Tracking and Order Filter newly introduced with FlexPro 2021, but are generally preferred. The Split Orders and Acceleration Hyperbola Family analysis objects (also deprecated) support presentation of the results of the Order Tracking analysis object. The deprecated order tracking analysis objects which were removed from the Gallery with FlexPro 2021 can still be viewed in the Gallery via File > Options > System Settings. Activate this option if you want to continue using the deprecated analysis objects in new projects (not recommended).

Analysis Objects

Revolution Synchronous Order Tracking

Order Filter

Revolution Synchronous Sampling

Harmonic Filter

Order Tracking (deprecated)

Split Orders (deprecated)

Acceleration Hyperbola Family (deprecated)

FPScript Function

RevolutionSyncSampling

OrderCuts

HarmonicRemovalFilter

STFTSpectrum

IIRFilter

IIRPeakFilter

Spectral Analysis Option

FlexPro's spectral analysis option offers state of the art procedures for the analysis of both stationary and non-stationary data in a ground breaking easy-to-use wizard interface. The Analysis Wizard can help you visually study the diverse spectral procedures and algorithms and compare them with one another. Once a given spectral procedure is refined and optimized, the results can be immediately saved as objects and documents in FlexPro's Object List.

Fourier Analyses

FlexPro offers four different options for FFT-based spectra. Three of these procedures are highlighted in the Fourier Spectral Analysis tutorial. This tutorial focuses on high dynamic range Fourier analysis and is particularly recommended if power or amplitude measurements are needed and there are low power signal components present.

The Fourier Spectrum is the main FFT spectral analysis. FlexPro 's best-exact-n FFT makes use of four different fast FFT algorithms. You can thus use any length data sequence without concern for the issues associated with power-of-2 zero padding. FlexPro offers a broad selection of data tapering windows. There are twenty windows with a fixed width windows and nine adjustable tapers, including Chebyshev, VanderMaas and Slepian windows. Zero padding is as simple as specifying the length of the FFT. All Fourier Analyses offer a variety of display formats. In addition to power normalization, you can display dB, normalized dB and amplitude spectra. Peak detection uses a cubic spline-based bin interpolation in order to achieve refinement in peak frequencies. FlexPro also offers critical limits for most Fourier procedures, an asset when you are uncertain if data statistically differ from random noise.

For reduced variance Fourier estimates, the Periodogram spectral procedure averages FFTs from overlapping segments. The Multitaper Spectrum procedure uses the series of orthogonal Slepian data tapers so that information at the edges of the data is utilized, and the variance of the spectral estimate is likewise reduced.

The Peak-Hold Spectrum also segments the data and calculates multiple spectra. However, these are not averaged. Rather, the maximum is formed across all spectra. This procedure is suitable for evaluating non-stationary signals in order to detect resonances during ramp-up, for instance.

Unevenly Spaced Data Fourier Analysis

The Unevenly Spaced Data Fourier Analysis generates a Lomb-Scargle Periodogram. Its primary use in FlexPro is for data with unevenly spaced X values or for data that contains void values. The algorithm was extended to use all continuous data tapering windows. A continuous Chebyshev approximation window was created primarily to service this algorithm.

High Resolution Frequency Estimators

FlexPro offers three options for Spectral estimators. These procedures are often the only alternative for very short data lengths. The Spectral Estimator tutorial is highly recommended for those instances where narrowband harmonics must be estimated to a very high accuracy and when stationary data segments are very short.

The AR Spectral Estimator analysis offers a selection of state-of-the-art autoregressive algorithms. An AR model is fitted to the data and its coefficients are used to generate a continuous spectrum. The best AR spectral methods are excellent frequency estimators, offering this accuracy with quite short data sets. Least-squares methods that offer intrinsic separation of signal and noise through singular value decomposition (SVD) are the most robust of FlexPro’s AR methods. Since AR spectral peaks can be exceedingly sharp, FlexPro offers an Adaptive option which uses a Runge-Kutta procedure to integrate the spectrum adaptively. This offers a frequency set containing frequencies concentrated near the peaks. The peak frequencies are computed from the complex roots of the AR model, and are computed to full machine precision.

The ARMA Spectral Estimator is viewed as a good model for signals with noise, since both peaks and nulls can be described. A pole-zero non-linear model is required. True state of the art non-linear fits are offered by FlexPro, and these can include spectral factorization for stability and SVD for signal-noise thresholding.

The EigenAnalysis Spectral Estimator analysis offers the MUSIC (Multiple Signal Classification) and EV (eigenvector) high-performance frequency estimation algorithms. Since these algorithms can produce exceedingly sharp spectral peaks, FlexPro's Adaptive spectrum is particularly valuable. The frequency of each spectral component is automatically refined to full machine precision.

Time-Frequency Spectrum

FlexPro offers three options for non-stationary data whose frequency domain characteristics are changing across time. For a good exposition of FlexPro's capabilities with non-stationary data, the Time-Frequency Spectral Analysis tutorial is highly recommended. The frequency-time tradeoff that is a key part of optimizing these analyses is covered in significant detail.

The STFT (Short Time Fourier Transform) Spectrum analysis produces a 3D time-frequency plot based upon a segmented overlapped FFT. Windowing is normally used to sharpen the resolution in time and minimize spectral leakage. Since the STFT has a uniform time-frequency resolution, amplitudes can be directly read from the spectrum.

The CWT (Continuous Wavelet Transformation) spectrum is used to split a signal into wavelets. FlexPro offers three adjustable mother wavelets. The number of frequencies is adjustable, as is whether or not the spacing is logarithmic. You can also set whether to use linear or logarithmic frequency division. A high frequency domain resolution Morlet wavelet is available for large data sets

The Peak-Hold STFT spectrum corresponds to the STFT spectrum, but only the global maximum together with its time and frequency data is copied from each single spectrum to the result.

Harmonic Estimation

One of the best ways to characterize signals that consist only of narrowband harmonics and noise is to directly model the oscillations in the time domain. It is also a robust method for harmonic distortion measurements. The Harmonic Estimation tutorial is essential, since the modeling is not a simple one-step procedure.

The Harmonic Estimation analysis uses a powerful composite algorithm that generates a parametric (sinusoids or damped sinusoids) model of the signal. The algorithm has two stages. In the first stage, a procedure is used to estimate the frequencies and component count. The best algorithms use SVD for removing the influence of observation noise. In the second stage a linear fit is made to determine the amplitudes, phases, and damping factors.

Harmonic distortion spectra are available, and are invaluable in optimizing THD measurements.

Two Signal Spectral Procedures

FlexPro offers a variety of two-signal spectral analyses. These are covered in the Cross Spectral Analysis tutorial. The tutorial provides the basics on Fourier Spectral Analysis. It is therefore suggested that you work through the Fourier Spectral Analysis Tutorial first.

The Fourier Cross Spectrum spectral procedure generates spectra that reflect the common power across two distinct signals. The Cross Periodogram procedure finds the cross-spectral components using overlapping segments. The Coherence and SNR Spectra option offers coherence computations and Signal to Noise Ratio (SNR) spectra on overlapping Fourier segments.

The Fourier Transfer Function procedure computes the Fourier transform transfer function relating linear system input and output streams.

Non-Linear Models

The Cepstral Analysis is primarily used in echo detection and speech signal applications.

Shock Response Spectrum (SRS)

The Shock Response Spectrum (SRS) is computed from the signal of an accelerometer. The acceleration signal is used for primary excitation of a series of single degree of freedom (SDOF) systems with customizable natural frequencies. The spectrum is formed by the absolute maxima, maxima or minima of these systems' responses. The shock response spectrum was originally introduced to analyze the damage potential of mechanical impulses, but it can also be used to analyze the damage potential of stationary random vibrations.

Instantaneous Quantities

FlexPro gives you the option to calculate instantaneous quantities (instantaneous amplitude, instantaneous phase and instantaneous frequency) of single-component signals. The instantaneous quantities can also be used to demodulate signals (amplitude demodulation, phase demodulation and frequency modulation).

The algorithm used to determine the instantaneous quantity is based on the Hilbert transform and uses the analytic signal derived from the Hilbert transform. Details and examples of this can be found in the Online Help covering the AnalyticSignal and Hilbert functions.

Analysis Objects

Cepstral Analysis

Cross Spectral Analysis

Fourier Spectral Analysis

Harmonic Estimation

Instantaneous Quantity

Spectral Estimator

Shock Response Spectrum

Time-Frequency Spectral Analysis

Uneven Data Fourier Spectral Analysis Object

FPScript Functions

AnalyticSignal

ApplyWindow

ARMASpectrum

ARSpectrum

CepstralAnalysis

Coherence

CrossPeriodogram

CrossSpectrum

CWTSpectrum

DataWindow

EigenSpectrum

FourierSpectrum

FourierSpectrumUneven

HarmonicEstimation

Hilbert

MultitaperSpectrum

Periodogram

SDOFResponse

SRS

SRSFromSDOFResponse

STFTSpectrum

TransferFunction

VarWindow

Tutorials

Fourier Spectral Analysis

Cross Spectral Analysis

Harmonic Estimation

Spectral Estimator

Time-Frequency Spectral Analysis

Acoustics Option

The Acoustics Option offers you a choice of the main analysis procedures for audio engineers.

To determine the Sound Level and Sound Power Level, you can use the Sound Level and Sound Power analysis objects. The computation of sound power in particular has become increasingly important in recent years, since it is required for the overall reduction of noise as well as for standardized labeling (e.g., for CE labeling) of machines.

The Loudness of a sound is a measurement for how loud a particular sound is perceived by humans as an auditory event. The loudness compares the physically measurable strength or amplitude of the sound (sound pressure level) to the loudness perceived by humans as perceived loudness. The perceived loudness is a psychoacoustic factor and depends on the sound pressure level, frequency spectrum and time response factors.

The sharpness of a sound is an auditory sensation which describes the particular aspect of the tonal perception that is correlated with the frequency distribution of the spectral envelope of sounds. The sharpness is a psychoacoustic variable and is determined by calculating the loudness.

For the frequency analysis, due to the constant relative bandwidth, the Octave and Third Octave Analysis is often used. These are calculated in the Acoustics Option using time domain filters so that accuracy, especially for lower frequencies, is not impaired. Bandwidths from octave via third octave through to 1/24 octave are available.

Observed Standards

Standard	Description
IEC 651, EN 60651	Sound level meter
IEC 804, EN 60804	Integrating-averaging sound level meter
IEC 1260, EN 61260	Band filter for octaves and fractions of octaves
DIN 45651	Octave filter for electroacoustic measurements
(DIN 45652)	Third octave filter for electroacoustic measurements
ISO 3744, 3745, 3746, DIN EN ISO 3744, 3745, 3746	Determines the sound power level of sources of noise. Enveloping surface method in an essentially free field over a reflecting plane. Precision grades 1 (3745), 2 (3744), and 3 (3746).
(DIN 45635)	Noise measurements for machines
ISO 532-1, ISO 532-2, ISO 532 A, ISO 532 B, DIN 45631	Procedure for calculating the loudness level and the loudness from the sound spectrum; E. Zwicker method (ISO 532-1, ISO 532 B - old standard, DIN 45631), method according to Moore-Glasberg (ISO 532-2) and method according to Stevens (ISO 532 A - old standard).
DIN 45692	Measurement technique for the simulation of the auditory sensation of sharpness

Analysis Objects

Loudness

Octave Analysis

Sharpness

Sound Power

Sound Level

FPScript Functions

AcousticCalibration

FrequencyWeightingA

FrequencyWeightingB

FrequencyWeightingC

Loudness

Sharpness

SoundLevel

SoundPower

TimeDomainOctaveAnalysis

See Also

Calibration in Acoustics

5.7 Reference

Analysis Objects and Templates

Signal Analysis

Analysis Object	Available In	Used for
Envelope	Professional, Developer Suite	Calculates the envelope of a data set.
Convolution	Basic, Professional, Developer Suite	Calculates the convolution product of two data sets.
Whole-Body Vibration	Human Body Vibrations Option	Used to analyze mechanical vibrations affecting the human body.
Hand-Transmitted Vibration	Human Body Vibrations Option	Used to analyze hand-transmitted vibrations that are transferred from vibrating tools through the palms and fingers to the hands and arms.
Correlation	Basic, Professional, Developer Suite	Calculates either the autocorrelation function of a data set or the cross-correlation of two data sets.
Rosette Transformation	Professional, Developer Suite	Calculates various quantities such as principal stresses or principal strains from two or three strain signals measured with the aid of a strain gauge rosette.
Signal Sampling	Basic, Professional,	Reduces or increases the number of values of a data set using different methods or samples

	Developer Suite	it for a new X data series.
Signal Analysis	Basic, Professional, Developer Suite	Computes the derivative or integral of a data set or calculates the area under the curve or the signal energy of a data set.
Signal Scaling	Basic, Professional, Developer Suite	Transforms a data set linearly or using a characteristic curve.

Spectral Analysis

Analysis Object	Available In	Used for
Cepstral Analysis	Spectral Analysis Option	Non-linear Fourier method for the deconvolution of two signals.
Fourier Cross Spectral Analysis	Spectral Analysis Option	Selection of four different Fourier cross-spectrum analyses: Cross-spectrum, cross-periodogram, coherence, transfer function
Fourier Spectral Analysis	Basic, Professional, Developer Suite	Provides three different spectral procedures.
Fourier Spectrum for Unevenly Sampled Data	Spectral Analysis Option	Generates a Lomb-Scargle periodogram for data sets with unevenly spaced X values and for data containing void values.
Harmonic Estimation	Spectral Analysis Option	Generates a parametric model of the signal.
Instantaneous Quantity	Spectral Analysis Option	Calculates different instantaneous quantities (instantaneous amplitude, instantaneous phase as well as instantaneous frequency) using the Hilbert transform.
Harmonic Filter	Order Tracking Option	Removes or extracts harmonic components of a selected order from time signals.
Order Filter	Order Tracking Option	Calculates order tracking for speed-dependent oscillations by bandpass filtering individual orders in the revolution domain.
Spectral Estimators	Spectral Analysis Option	Provides three spectral estimators with high frequency resolution.

Shock Response Spectrum	Spectral Analysis Option	Calculates the shock response spectrum from the responses of several SDOF systems that are excited by an accelerometer signal.
Revolution Synchronous Order Tracking	Order Tracking Option	Calculates an order tracking analysis for speed-dependent oscillations by Fourier analysis of the signal converted in the revolution domain.
Time-Frequency Spectral Analysis	Professional, Developer Suite	Provides two different time-frequency spectral procedures for non-stationary data.

Filters

Analysis object	Available In	Used for
CFC Filter	Digital Filters Option	Filters signals with a CFC (Channel Frequency Class) filter.
Event Isolation	Basic, Professional, Developer Suite	Searches data sets for various events.
FIR Filter (Equiripple Method)	Digital Filters Option	Filters signals or calculates the impulse response of a finite impulse response (FIR) filter using the Equiripple method.
FIR-Filter (Window Method)	Digital Filters Option	Filters signals or calculates the impulse response of a finite impulse response (FIR) filter using windowing.
IIR Filter	Digital Filters Option	Filters signals or calculates the numerator and denominator coefficients of an infinite impulse response (IIR) filter.
Harmonic Filter	Order Tracking Option	Removes or extracts harmonic components of a selected order from time signals.
Order Filter	Order Tracking Option	Calculates order tracking for speed-dependent oscillations by bandpass filtering individual orders in the revolution domain.
Signal Smoothing	Basic, Professional, Developer Suite	Smoothens a data set.

Signal Correction	Basic, Professional, Developer Suite	Deletes or interpolates void values from floating-point data sets.

Curve Fitting

Analysis Object	Available In	Used for
2D Approximation	Basic, Professional, Developer Suite	Approximates a selectable model function Y(X, Z) to 2D data based on the least square error method.
Approximation	Basic, Professional, Developer Suite	Approximates a selectable model function Y(X) to data based on the least square error method.
Circle Approximation	Professional, Developer Suite	Carries out circular curve fitting on the underlying data.
Curve Transformation	Basic, Professional, Developer Suite	Converts space curves into signal series and vice versa.
Linear Interpolation	Basic, Professional, Developer Suite	Carries out linear interpolation for data.
Non-Linear Curve Fitting	Basic, Professional, Developer Suite	Fits a model function based on an independent variable and several parameters to a given data set.
Surface Interpolation	Basic, Professional,	Provides four methods for smoothing surfaces.

Analysis Object	Available In	Used for
	Developer Suite	
Parametric Spline Interpolation	Basic, Professional, Developer Suite	Carries out parametric spline interpolation for data.
Peak Fitting	Basic, Professional, Developer Suite	Approximates multiple instances of a peak function to a pre-defined data set.
Regression	Basic, Professional, Developer Suite	Calculates various regressions for a data set.
Signal Sampling	Basic, Professional, Developer Suite	Provides different methods for increasing or reducing the number of values of data sets or samples them for new X values.
Spline Interpolation	Basic, Professional, Developer Suite	Carries out a spline interpolation for data.
Revolution Synchronous Sampling	Order Tracking Option	Transforms a signal sampled over time into the revolution domain.

Statistics

Analysis object	Available In	Used for
ANOVA	Statistics Option	Performs a Fisher analysis of variance for a series of samples from a normally distributed population.

Goodness-of-Fit Test	Statistics Option	Performs goodness-of-fit tests for samples from a normally or exponentially distributed population.
Outlier Correction	Statistics Option	Deletes outliers from a sample.
Outlier Test	Statistics Option	Checks one sample from a normally distributed population for outliers.
Statistical Quantity	View, Basic, Professional, Developer Suite	Calculates various statistical quantities of a data set.
Variance Test	Statistics Option	Tests whether the variances from samples - originating from a normally distributed population - are significantly different or not.
Distribution	Statistics Option	Calculates the distribution or density function of the most important theoretical distributions.
Confidence Interval	Statistics Option	Calculates the confidence intervals for the parameters of the normal distribution of a population based on a sample.

Counting Procedures

Analysis object	Available In	Used for
Histogram	Basic, Professional, Developer Suite	Creates histograms for data.
Counting	Count Option	Counts a signal with one of the procedures described in the DIN45667 standard.
Count Matrix	Count Option	Determines the Rainflow matrix or the Markov (transition) matrix of a signal.
Rainflow Count	Count Option	Derives univariate collectives from an existing Rainflow matrix or Markov matrix.
Compound Count	Count Option	Counts two signals synchronously with some of the procedures described in the DIN45667 standard.

Acoustics

Analysis object	Available In	Used for
Loudness	Acoustics Option	Calculates the loudness from third octave spectra (ISO 532:B) or octave spectra (ISO 532:A).
Octave Analysis	Acoustics Option	Performs an octave analysis with digital filters in the time domain.
Sound Power	Acoustics Option	Determines the sound power according to international standards.
Sound Level	Acoustics Option	Determines sound levels from recorded microphone signals according to international standards.
Sharpness	Acoustics Option	Calculates the sharpness of a stationary or time-varying sound signal.

Other

Analysis object	Available In	Used for
Signal	View, Basic, Professional, Developer Suite	Calculates various synthetic signals.

FPScript Operators

Arithmetic

Operator	Available In	Used for
Addition (+)	View, Basic, Professional, Developer Suite	Adds two expressions.
Division (/)	View, Basic, Professional, Developer Suite	Divides two expressions.
Division Remainder (Mod)	View, Basic, Professional	Determines the remainder of a division.

Operator	Available In	Used for
	, Developer Suite	
Multiplication (*)	View, Basic, Professional, Developer Suite	Multiplies two expressions.
Negation (-)	View, Basic, Professional, Developer Suite	Negates an expression.
Exponentiation (^)	View, Basic, Professional, Developer Suite	Raises an expression by a power.
Subtraction (-)	View, Basic, Professional, Developer Suite	Subtracts two expressions from each other.
Polarity Sign (#)	View, Basic, Professional, Developer Suite	Determines a code number for the polarity sign of an expression.

Bitwise Operations

Operator	Available In	Used for
Bitwise Boolean Exclusive OR	View, Basic, Professional, Developer Suite	Performs a bitwise Boolean Exclusive OR operation for two expressions.
Bitwise Boolean NOT (~)	View, Basic, Professional, Developer Suite	Determines the bitwise complement.
Bitwise Boolean OR (|)	View, Basic, Professional	Performs a bitwise Boolean OR operation for two expressions.

Operator	Available In	Used for
	, Developer Suite	
Bitwise Boolean AND (&)	View, Basic, Professional, Developer Suite	Performs a bitwise Boolean AND operation for two expressions.
Bitwise Shift Right (>>)	View, Basic, Professional, Developer Suite	Shifts the bits in an integer to the right by the specified number of positions.
Bitwise Shift Left (<<)	View, Basic, Professional, Developer Suite	Shifts the bits in an integer to the left by the specified number of positions.

Data Organization

Operator	Available In	Used for
Bundle	View, Basic, Professional, Developer Suite	Bundles scalar values into data series, data series into data matrices or signals into signal series.
Data series	View, Basic, Professional, Developer Suite	Forms a data series with linear ascending or descending values.
List	View, Basic, Professional, Developer Suite	Combines any values into a list.
Concatenation (:)	View, Basic, Professional, Developer Suite	Appends strings, scalar values, data series, data matrices, signals, signal series and space curves to one another.
Multiply (#)	View, Basic, Professional, Developer Suite	Multiplies a value by a given number.

Data Conversion

Operator	Available In	Used for
AbsoluteTime	View, Basic, Professional, Developer Suite	Converts time data to absolute time.
Boolean	View, Basic, Professional, Developer Suite	Converts an expression into the Boolean data type.
CalendarTime	View, Basic, Professional, Developer Suite	Converts an expression into the calendar time data type.
ComplexInteger16	View, Basic, Professional, Developer Suite	Converts an expression into the 16-bit complex integer data type.
ComplexInteger32	View, Basic, Professional, Developer Suite	Converts an expression into the 32-bit complex integer data type.
ComplexInteger64	View, Basic, Professional, Developer Suite	Converts an expression into the 64-bit complex integer data type.
ComplexFloatingPoint32	View, Basic, Professional, Developer Suite	Converts an expression into the 32-bit complex floating point value data type.
ComplexFloatingPoint64	View, Basic, Professional, Developer Suite	Converts an expression into the 64-bit complex floating point value data type.
FloatingPoint32	View, Basic, Professional, Developer Suite	Converts an expression into the 32-bit floating point value data type.

Operator	Available In	Used for
FloatingPoint64	View, Basic, Professional, Developer Suite	Converts an expression into the 64-bit floating point value data type.
Integer16	View, Basic, Professional, Developer Suite	Converts an expression into the 16-bit integer data type.
Integer32	View, Basic, Professional, Developer Suite	Converts an expression into the 32-bit integer data type.
Integer64	View, Basic, Professional, Developer Suite	Converts an expression into the 64-bit integer data type.
RelativeTime	View, Basic, Professional, Developer Suite	Converts time data to relative time.
String	View, Basic, Professional, Developer Suite	Converts an expression into the string data type.
Unit	View, Basic, Professional, Developer Suite	Adds a unit to a value and converts it to a quantity or transforms the unit of a quantity.
Value	View, Basic, Professional, Developer Suite	Removes the unit from a quantity or evaluates an object reference.

Complex Numbers

Operator	Available In	Used for
Imag	View, Basic,	Forms the imaginary part of a complex

Operator	Available In	Used for
	Professional, Developer Suite	number.
Complex Number	View, Basic, Professional, Developer Suite	Combines a real part and an imaginary part into a complex number.
Conjugate Complex	View, Basic, Professional, Developer Suite	Forms the conjugate-complex expression.
Real	View, Basic, Professional, Developer Suite	Forms the real part of a complex number.

Logic

Operator	Available In	Used for
Logical NOT	Basic, Professional, Developer Suite	Carries out a logical Boolean NOT operation for an expression.
Logical OR	Basic, Professional, Developer Suite	Carries out a logical Boolean OR operation for two expressions.
Logical AND	Basic, Professional, Developer Suite	Carries out a logical Boolean AND operation for two expressions.

Data Access

Operator	Available In	Used for
Property	View, Basic, Professional, Developer Suite	Used to access the properties of an object.
Index	View, Basic, Professional, Developer Suite	Extracts an individual value or a section from an expression.

Operator	Available In	Used for
Indirection	View, Basic, Professional, Developer Suite	Used to access data sets, functions and variables by using their names.
Component	View, Basic, Professional, Developer Suite	Extracts the X, Y or Z component from a data set with a composite data structure.
List Element	View, Basic, Professional, Developer Suite	Takes an element from a list or selects a list element to which a new value is to be assigned. The operator can also be used to access a component of a composite data structure.

Comparison

Operator	Available In	Used for
Equal To (==)	View, Basic, Professional, Developer Suite	Compares two numeric expressions and returns TRUE if the left expression is equal to the right expression.
Greater Than Or Equal To (>=)	View, Basic, Professional, Developer Suite	Compares two numeric expressions and returns TRUE if the left expression is greater than or equal to the right expression.
Greater Than (>)	View, Basic, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is greater than the right expression.
Less Than Or Equal To (<=)	View, Basic, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is less than or equal to the right expression.
Less Than (<)	View, Basic, Professional, Developer Suite	Compares two numerical expressions and returns TRUE if the left expression is less than the right expression.
Not Equal To (<>)	View, Basic, Professional	Compares two numerical expressions and returns TRUE if the left expression

Operator	Available In	Used for
	l, Developer Suite	is not equal to the right expression.

FPScript Statements

Conditional Execution

Statement	Available In	Used for
If...Then...Else	View, Basic, Professional, Developer Suite	Evaluates an expression and executes a series of statements, depending on the result.

Loops

Statement	Available In	Used for
Do...While	View, Basic, Professional, Developer Suite	Executes a series of statements until a given condition is FALSE.
For Each Column...End	View, Basic, Professional, Developer Suite	Repeats a series of statements for all data series, i.e. columns, in a data matrix or all signals in a signal series with the option of concurrent execution.
For Each Row...End	View, Basic, Professional, Developer Suite	Repeats a series of statements for all rows of a data set.
For Each Value...End	View, Basic, Professional, Developer Suite	Repeats a series of instructions while traversing all values of a data set.
For Each Element...End	View, Basic, Professional, Developer Suite	Repeats a series of statements for all items in a list with the option of concurrent execution.

Statement	Available In	Used for
For...End	View, Basic, Professional, Developer Suite	Repeats a series of statements several times until a loop counter reaches a certain value.
While...Do...End	View, Basic, Professional, Developer Suite	Executes a series of statements while a given condition is TRUE.

Assignment

Statement	Available In	Used for
Append	View, Basic, Professional, Developer Suite	Appends additional data to a value stored in a variable.
Assignment	View, Basic, Professional, Developer Suite	Assigns a value to an object property, a variable or a component, or a list element therein.
Indexed Assignment	View, Basic, Professional, Developer Suite	Assigns a new value to a section in a data series or a data matrix.
Set	View, Basic, Professional, Developer Suite	Assigns an object reference to a variable or a list element.

Other

Statement	Available In	Used for
Arguments	View, Basic, Professional, Developer Suite	Declares arguments for a formula.

Statement	Available In	Used for
Break	View, Basic, Professional, Developer Suite	Ends a loop prematurely.
Dim	View, Basic, Professional, Developer Suite	Declares local variable for a formula.
Recalculate	View, Basic, Professional, Developer Suite	Forces recalculation of the FPScript formula the next time the result is requested.
Return	View, Basic, Professional, Developer Suite	Ends the calculation of a formula and passes the specified expression as a result.
Throw	View, Basic, Professional, Developer Suite	Throws an exception and passes the specified expression as the value of the exception.
Try...Catch...End	View, Basic, Professional, Developer Suite	Catches exceptions that occur in the included statements and allows their processing.
With Environment...Do...End	View, Basic, Professional, Developer Suite	Sets one or more FPScript environmental variables to a particular value and compiles a series of statements within the context of these set environmental variables or runs these within this context.

FPScript Keywords

Keyword	Available In	Used for
ActivatedFolder	View, Basic, Professional, Developer Suite	Provides access to the activated subfolder within a folder in the FlexPro project database.

Keyword	Available In	Used for
Application	View, Basic, Professional, Developer Suite	Provides access to the "Application" automation object, which represents the FlexPro application.
As Object	View, Basic, Professional, Developer Suite	Interprets a path name as an object reference instead of a value.
python	Basic, Professional, Developer Suite	Calls a Python function or reads a Python constant.
This	View, Basic, Professional, Developer Suite	Provides access to the formula where the FPScript code is located.
ThisFPObject	View, Basic, Professional, Developer Suite	Provides access to the FlexPro object that executes an embedded FPScript code.
ThisObject	View, Basic, Professional, Developer Suite	Provides access to the automation object that executes an embedded FPScript code.

FPScript Properties

Property	Available In	Used for
AssignedX	Basic, Professional, Developer Suite	Read access to the name of the data object assigned as the X component. Read/write access to the name of the data object assigned as the X component of the current formula.
AssignedZ	Basic, Professional, Developer Suite	Read access to the name of the data object assigned as the Z component. Read/write access to the name of the data object assigned as the Z component of the current formula.
AssignHeader	Basic, Professional, Developer Suite	Read access to the header information attribute of a data object. Read/write access to the header information attribute of the current formula.

Property	Available In	Used for
Author	Basic, Professional, Developer Suite	Read access to the name of a data object author. Read/write access to the name of the author of the current formula.
Calculations	Basic, Professional, Developer Suite	Read access to a data object calculation.
Calculations.Unit	Basic, Professional, Developer Suite	Read access to the unit of a data object calculation.
Comments	Basic, Professional, Developer Suite	Read access to the object's comments.
CommentsX	Basic, Professional, Developer Suite	Read access to the comments on a data object's X component. Read/write access to the comments of the X component of the current formula.
CommentsY	Basic, Professional, Developer Suite	Read access to the comments on the data object's Y component. Read/write access to the comments on the current formula's Y component.
CommentsZ	Basic, Professional, Developer Suite	Read access to the comments on the data object's Z component. Read/write access to the comments on the current formula's Z component.
CreationTime	Basic, Professional, Developer Suite	Read access to an object's creation time. Read/write access to the creation time of the current formula.
Cursor.NameX	Basic, Professional, Developer Suite	Read access to a labeling of the X component of the curve under a bound cursor
Cursor.NameY	Basic, Professional,	Read access to a labeling of the Y component of the curve under a bound

Property	Available In	Used for
	Developer Suite	cursor
Cursor.NameZ	Basic, Professional, Developer Suite	Read access to a labeling of the Z component of the curve under a bound cursor
Cursor.PositionIndex	Basic, Professional, Developer Suite	Read access to the row index of a bound cursor.
Cursor.PositionX	Basic, Professional, Developer Suite	Read access to the X value of a cursor.
Cursor.PositionY	Basic, Professional, Developer Suite	Read access to the Y value of a cursor.
Cursor.PositionZ	Basic, Professional, Developer Suite	Read access to the Z value of a cursor.
Cursor.PositionZIndex	Basic, Professional, Developer Suite	Read access to the column index of a bound cursor.
Cursor.UnitX	Basic, Professional, Developer Suite	Read access to the unit of the X component of the curve under a bound cursor.
Cursor.UnitY	Basic, Professional, Developer Suite	Read access to the unit of the Y component of the curve under a bound cursor.
Cursor.UnitZ	Basic, Professional, Developer Suite	Read access to the unit of the Z component of the curve under a bound cursor.

Property	Available In	Used for
FilePath	Basic, Professional, Developer Suite	Read access to the name and the path to the media file of a media object.
Formula	Basic, Professional, Developer Suite	Read access to the code of a formula.
FullName	Basic, Professional, Developer Suite	Read access to the name and the path in the project database of an object.
Hyperlink	Basic, Professional, Developer Suite	Read access to the hyperlink of an object. Read/write access to the hyperlink of the current formula.
LowerRangeLimitX	Basic, Professional, Developer Suite	Read access to the lower range limit of the X component of a data object. Read/write access to the lower range limit of the X-component of the current formula.
LowerRangeLimitY	Basic, Professional, Developer Suite	Read access to the lower range limit of the Y component of a data object. Read/write access to the lower range limit of the Y component of the current formula.
LowerRangeLimitZ	Basic, Professional, Developer Suite	Read access to the lower range limit of the Z component of a data object. Read/write access to the lower range limit of the Z component of the current formula.
Markers.PositionCurve	Basic, Professional, Developer Suite	Read access to the first curve numbers of the bound markers.
Markers.PositionCurve2	Basic, Professional,	Read access to the second curve numbers of the bound range markers.

Property	Available In	Used for
	Developer Suite	
Markers.PositionIndex	Basic, Professional, Developer Suite	Read access to the first row indices of the bound markers.
Markers.PositionIndex2	Basic, Professional, Developer Suite	Read access to the second row indices of the bound range markers.
Markers.PositionX	Basic, Professional, Developer Suite	Read access to the X values of the markers.
Markers.PositionX2	Basic, Professional, Developer Suite	Read access to the second X values of the range markers.
Markers.PositionY	Basic, Professional, Developer Suite	Read access to the Y values of the markers.
Markers.PositionY2	Basic, Professional, Developer Suite	Read access to the second Y values of the range markers.
Markers.PositionZ	Basic, Professional, Developer Suite	Read access to the Z values of the markers.
Markers.PositionZ2	Basic, Professional, Developer Suite	Read access to the second Z values of the range markers.
Markers.PositionZIndex	Basic, Professional, Developer Suite	Read access to the first column indices of the bound markers.

Property	Available In	Used for
Markers.PositionZIndex2	Basic, Professional, Developer Suite	Read access to the second column indices of the bound range markers.
ModificationTime	Basic, Professional, Developer Suite	Read access to the modification time of an object.
Name	Basic, Professional, Developer Suite	Read access to the name of an object. Read/write access to the name of the embedded FPScript.
Objects.Names	Basic, Professional, Developer Suite	Read access to the names of the objects in a folder.
Origin	Basic, Professional, Developer Suite	Read access to the origin of a data object. Read/write access to the origin of the current formula.
Parameters	Basic, Professional, Developer Suite	Read access to a parameter of an object. Read/write access to the value of a parameter of the current formula.
Parameters.Unit	Basic, Professional, Developer Suite	Read access to the unit of an object parameter.
Path	Basic, Professional, Developer Suite	Read access to the path in the project database of an object.
QuantityX	Basic, Professional, Developer Suite	Read access to the name of the physical quantity of a data object's X component. Read/write access to the name of the physical quantity of the current formula's X component.

Property	Available In	Used for
QuantityY	Basic, Professional, Developer Suite	Read access to the name of the physical quantity of a data object's Y component. Read/write access to the name of the physical quantity of the current formula's Y component.
QuantityZ	Basic, Professional, Developer Suite	Read access to the name of the physical quantity of a data object's Z component. Read/write access to the name of the physical quantity of the current formula's Z component.
TimestampX	Basic, Professional, Developer Suite	Read access to the timestamp on a data object's X component. Read/write access to the timestamp on the current formula's X component.
TimestampY	Basic, Professional, Developer Suite	Read access to the timestamp of the Y component of a data object. Read/write access to the timestamp on the current formula's Y component.
TimestampZ	Basic, Professional, Developer Suite	Read access to the timestamp on the data object's Z component. Read/write access to the timestamp on the current formula's Z component.
UnitX	Basic, Professional, Developer Suite	Read access to the unit of a data object's X component. Read/write access to the unit of the current formula's X component.
UnitY	Basic, Professional, Developer Suite	Read access to the unit of a data object's Y component. Read/write access to the unit of the Y component of the current formula.
UnitZ	Basic, Professional, Developer Suite	Read access to the unit of a data object's Z component. Read/write access to the unit of the current formula's Z component.
UpperRangeLimitX	Basic, Professional, Developer Suite	Read access to the upper range limit of the X component of a data object. Read/write access to the upper range

Property	Available In	Used for
		limit of the X-component of the current formula.
UpperRangeLimitY	Basic, Professional, Developer Suite	Read access to the upper range limit of the Y component of a data object. Read/write access to the upper range limit of the Y component of the current formula.
UpperRangeLimitZ	Basic, Professional, Developer Suite	Read access to the upper range limit of the Z component of a data object. Read/write access to the upper range limit of the Z component of the current formula.

FPScript Functions

Acoustics

Function	Availability	Use
AcousticCalibration	Option Acoustics	Calculates the calibration value from a calibration signal.
FrequencyWeightingA	Option Acoustics	Calculates the frequency weighting A according to IEC 651 for an input signal or a signal series.
FrequencyWeightingB	Option Acoustics	Calculates the frequency weighting B according to IEC 651 for an input signal or a signal series.
FrequencyWeightingC	Option Acoustics	Calculates the frequency weighting C according to IEC 651 for an input signal or a signal series.
Loudness	Option Acoustics	Calculates the loudness of a sound signal.
Sharpness	Option Acoustics	Calculates the sharpness of a sound signal.

Function	Availability	Use
SoundLevel	Option Acoustics	Calculates sound levels from an input signal or an input signal series.
SoundPower	Option Acoustics	Calculates the sound power level. The level signals of several microphones are averaged energetically and a number of correction terms for barometric pressure, temperature, background noise, environmental correction and size of the measuring surface are considered.
TimeDomainOctaveAnalysis	Option Acoustics	Calculates an octave analysis of the input signal using time domain filters. The bandwidths octave, third octave, 1/6 octave, 1/12 octave and 1/24 octave can be selected. The frequency range within which the octave analysis is calculated can be selected within the limits of 1 Hz to 100 kHz.

Calendar Time

Function	Availability	Use
CurrentDate	View, Basic, Professional, Developer Suite	Determines the calendar time value of the current date and time with a precision of one second.
Date	View, Basic, Professional, Developer Suite	Determines the date of one or more calendar time values.
DateTime	View, Basic, Professional, Developer Suite	The function forms a calendar time value from individual date and time elements.

Function	Availability	Use
Day	View, Basic, Professional, Developer Suite	Determines the day of the month of one or more calendar time values or the number of days of one or more time span values.
DayOfYear	View, Basic, Professional, Developer Suite	Determines the day in the year of one or more calendar time values.
Hour	View, Basic, Professional, Developer Suite	Determines the hour of the day of one or more calendar time or time span values.
Minute	View, Basic, Professional, Developer Suite	Determines the minute in the hour of one or more calendar time or time span values.
Month	View, Basic, Professional, Developer Suite	Determines the month in the year of one or more calendar time values.
Second	View, Basic, Professional, Developer Suite	Determines the second in the minute of one or more calendar time or time span values.
Time	View, Basic, Professional, Developer Suite	Determines time of one or more calendar time values as a time span value or generates a time span value based on the specified elements.
WeekDay	View, Basic, Professional, Developer Suite	Determines the week day for one or more calendar time values.
Year	View, Basic, Professional, Developer Suite	Determines the value for the year of one or more calendar time values.

Counting Procedures

Function	Availability	Use
CompoundMaximumValueCount	Option Counting Procedures	Performs a maximum value count in accordance with the DIN 45667 standard synchronously for two signals.
CompoundSamplingCount	Option Counting Procedures	Performs a sampling count in accordance with the DIN 45667 standard synchronously for two signals.
CompoundTimeAtLevelCount	Option Counting Procedures	Performs a time-at-level count in accordance with the DIN 45667 standard synchronously for two signals.
Histogram	Basic, Professional, Developer Suite	Creates a histogram for a class division that you can specify.
LevelCrossingCount	Option Counting Procedures	Performs a level crossing count from the Markov or Rainflow Matrix. The function counts the absolute frequencies of positive and negative level crossings.
MarkovMatrix	Option Counting Procedures	Determines the Markov Matrix (transition matrix) for a data set.
MaximumValueCount	Option Counting Procedures	Carries out a maximum value count in accordance with the DIN 45667 standard.
MeanValueCount	Option Counting Procedures	Counts a signal and determines the class mean values.
PeakCount	Option Counting Procedures	Performs a peak value count from the Rainflow Matrix or Markov Matrix. The function counts the absolute or cumulated frequencies of peak or trough values.
RainflowMatrix	Option Counting Procedures	Determines the Rainflow Matrix for a data set.
		Performs a range count from the Markov Matrix or a range pair count

Curve Fitting

Function	Availability	Use
Approximation	Basic, Professional, Developer Suite	Approximates a linear model Y(X) to given data according to the method of the least square error.
Approximation2D	Basic, Professional, Developer Suite	Approximates a linear model Y(X, Z) with two independent variables to given 2D data according to the method of the least square error.
LeastSquaresCircle	Professional, Developer Suite	Calculates the least squares circle (LSCI) of a two-dimensional point set (least squares reference circle). Used to determine roundness.
MaximumInscribedCircle	Professional, Developer Suite	Calculates the maximum inscribed circle (MICI) of a two-dimensional point set. Used to determine roundness.
MinimumCircumscribedCircle	Professional, Developer Suite	Calculates the minimum circumscribed circle (MCCI) of a two-dimensional point set. Used to determine roundness.
MinimumZoneCircle	Professional, Developer Suite	Calculates the reference circles of the minimum zone (MZCI) of a two-dimensional point set. Used to determine roundness.
NonLinCurveFit	Basic, Professional, Developer Suite	Approximates a non-linear model to a data set and returns the model parameters found, the modeled data and a large number of statistical results.

Function	Availability	Use
NonLinModel	Basic, Professional, Developer Suite	Calculates a non-linear model function for model parameters that you specify.
ParameterEstimation	Basic, Professional, Developer Suite	Approximates a non-linear model to a data set. Here, non-iterative methods are used. The Grid Search and the Random Search methods are the options available.

Curve Interpolation

Function	Availability	Use
ConvexHull	Basic, Professional, Developer Suite	Calculates the convex hull of a two-dimensional point set.
ParametricSpline	Basic, Professional, Developer Suite	Interpolates a data set through a parametric spline curve and samples this at definable points.
PeriodicSpline	Basic, Professional, Developer Suite	Interpolates a data set using a periodic spline curve and samples the spline curve at definable points.
PolynomialInterpolation	Basic, Professional, Developer Suite	Carries out polynomial interpolation at specifiable sampling points. Polynomial interpolation is understood to be the search for a polynomial that runs exactly through the predefined points.
SmoothingSpline	Basic, Professional, Developer Suite	Interpolates a data set through a compensating spline curve and samples this at definable points.
Spline	Basic, Professional, Developer Suite	Interpolates a data set with a spline curve and samples this curve at definable points.

Data Import

Function	Availability	Use
HistoryBaseFirstTime	View, Basic, Professional, Developer Suite	Provides the date and time of the first (oldest) available measurement value from a HistoryBase ring buffer.

Function	Availability	Use
HistoryBaseLastTime	View, Basic, Professional, Developer Suite	Provides the date and time of the last (most recent) available measurement value from a HistoryBase ring buffer.
HistoryBaseRead	View, Basic, Professional, Developer Suite	Reads a section of a trace from a HistoryBase ring buffer.
HistoryBaseReadTime	View, Basic, Professional, Developer Suite	Reads a section of a time trace from a HistoryBase ring buffer.
IsTimeInHistoryBase	View, Basic, Professional, Developer Suite	Returns **TRUE** if present. Otherwise, it returns **FALSE**.
IsTimeInRingBuffer	View, Basic, Professional, Developer Suite	Returns **TRUE** if present. Otherwise, it returns **FALSE**.
ReadASAMODS	Option ASAM ODS Import	Reads ASAM ODS (Open Data Services) data.
ReadAstroMedDCRFile	View, Basic, Professional, Developer Suite	Reads a data set from a file in Astro-Med Dash 18/8X Data Capture Record format.
ReadBinaryFile	View, Basic, Professional, Developer Suite	Reads a data series from a binary file.
ReadBSME3File	View, Basic, Professional, Developer Suite	Reads a data set from a file in B+S ME 3 format V 3.00.
ReadCDFFile	View, Basic, Professional, Developer Suite	Reads a z or r variable from a file in CDF format.

Function	Availability	Use
ReadDataFile	View, Basic, Professional, Developer Suite	Reads an element, such as a channel, from a data file.
ReadDPOFile	View, Basic, Professional, Developer Suite	Reads a signal from a file in Tektronix DPO 3D image file format.
ReadEDASFileX	View, Basic, Professional, Developer Suite	Reads the X component of a channel from a file in EDAS format.
ReadEDASFileY	View, Basic, Professional, Developer Suite	Reads the Y component of a channel from a file in EDAS format.
ReadExcelFile	View, Basic, Professional, Developer Suite	Reads one or more data sets from an Excel file.
ReadFlexProFile	View, Basic, Professional, Developer Suite	Reads a data set from a FlexPro text file.
ReadGouldScopeFile	View, Basic, Professional, Developer Suite	Reads a data set from a file in Gould Oscilloscope format.
ReadINSIGHTFile	View, Basic, Professional, Developer Suite	Reads a data set from a file in INSIGHT's SDB or MDF format.
ReadIOtechFile	View, Basic, Professional, Developer Suite	Reads a data set from a file in IOtech's WaveView, DaqView, PersonalDaqView or ChartView format.
ReadLeCroyWaveformFile	View, Basic, Professional, Developer Suite	Reads a LeCroy Waveform file as a signal.

Function	Availability	Use
ReadMATLABFile	View, Basic, Professional, Developer Suite	Reads a field from a file in MATLAB format.
ReadNextViewFile	View, Basic, Professional, Developer Suite	Reads an analog or digital channel from a file in BMC NextView format.
ReadODBC	View, Basic, Professional, Developer Suite	Reads one or more columns from a table or data source using ODBC.
ReadORFile	View, Basic, Professional, Developer Suite	Reads a channel from a file in Yokogawa ORP/ORM format.
ReadOROSWaveFile	View, Basic, Professional, Developer Suite	Reads a channel from an OROS Wave file.
ReadPCScopeFileX	View, Basic, Professional, Developer Suite	Reads the X component of a channel from a file in IMTEC's PC-Scope format.
ReadPCScopeFileY	View, Basic, Professional, Developer Suite	Reads the Y component of a channel from a file in IMTEC's PC-Scope format.
ReadSignalysFileX	View, Basic, Professional, Developer Suite	Reads the X component of a channel from a file in Signalys format.
ReadSignalysFileY	View, Basic, Professional, Developer Suite	Reads the Y component of a channel from a file in Signalys format.
ReadSPSSFile	View, Basic, Professional, Developer Suite	Reads a variable from an SPSS file.

Function	Availability	Use
ReadSYSTATFile	View, Basic, Professional, Developer Suite	Reads a variable from a SYSTAT file.
ReadTAFFmatFile	View, Basic, Professional, Developer Suite	Reads a data set from a file in TEAC TAFFmat format.
ReadTDSFile	View, Basic, Professional, Developer Suite	Reads a signal from a file in Tektronix TDS Waveform format.
ReadTEAMFile	View, Basic, Professional, Developer Suite	Reads a data set from a file in Nicolet Odyssey TEAM format.
ReadTextData	View, Basic, Professional, Developer Suite	Reads one or more data sets from a text file.
ReadTextFile	View, Basic, Professional, Developer Suite	Reads a data set from a text file. **This is a deprecated function! Use the** ReadTextData **function instead.**
ReadTurboLabFileX	View, Basic, Professional, Developer Suite	Reads the X component of a channel from a file in TurboLab format.
ReadTurboLabFileY	View, Basic, Professional, Developer Suite	Reads the Y component of a channel from a file in TurboLab format.
ReadWaveFile	View, Basic, Professional, Developer Suite	Reads an audio file as a signal or signal series.
ReadWFTFile	View, Basic, Professional, Developer Suite	Reads a data set from a file in Nicolet Waveform format.

Function	Availability	Use
RingBufferFirstTime	View, Basic, Professional, Developer Suite	Provides the date and time of the first (oldest) available measurement value from a HistoryBase ring buffer.
RingBufferLastTime	View, Basic, Professional, Developer Suite	Provides the date and time of the last (most recent) available measurement value from a HistoryBase ring buffer.
RingBufferRead	View, Basic, Professional, Developer Suite	Reads a section of a trace from a HistoryBase ring buffer.
RingBufferReadTime	View, Basic, Professional, Developer Suite	Reads a section of a time trace from a HistoryBase ring buffer.

Data Information

Function	Availability	Use
DataOrder	View, Basic, Professional, Developer Suite	Determines the data order of a data set or of its X component.
DataQuery	Professional, Developer Suite	This function specifies whether the data query is to be carried out on external data (files) or internal data (project database).
DataStructure	View, Basic, Professional, Developer Suite	Determines the data structure of a data set.
DataType	View, Basic, Professional, Developer Suite	Determines the data type of a data set.

Function	Availability	Use
HasVoidValues	View, Basic, Professional, Developer Suite	Returns TRUE if a data set contains void values.
Increment	View, Basic, Professional, Developer Suite	Determines the increment of a data set or of its X component with linearly increasing or decreasing values.
ListElementNames	View, Basic, Professional, Developer Suite	Determines the name of the list element with the specified index.
NumberOfColumns	View, Basic, Professional, Developer Suite	Determines the number of columns in a data set.
NumberOfElements	View, Basic, Professional, Developer Suite	Determines the number of elements in a data set.
NumberOfRows	View, Basic, Professional, Developer Suite	Determines the number of rows in a data set.
ParameterList	View, Basic, Professional, Developer Suite	Returns the parameters of a data set as a list.
Rank	Basic, Professional, Developer Suite	Determines the number of dimensions of a data set.
SamplingRate	View, Basic, Professional, Developer Suite	Determines the sampling rate of a data set or of its X component with linearly increasing or decreasing values.
Shape	Basic, Professional, Developer Suite	Determines the dimension lengths of a data set.

Function	Availability	Use
SIUnits	Basic, Professional, Developer Suite	Returns the presentation of the unit of a data set in SI units.
Trend	Basic, Professional, Developer Suite	Determines the constant, linear or adaptive trend of a data set.
Unit	Basic, Professional, Developer Suite	Returns the unit of a data set as a quantity with the value 1.
UnitSymbol	Basic, Professional, Developer Suite	Returns the symbol of the data set unit.
UnitType	Basic, Professional, Developer Suite	Determines the type of unit for a data set.

Data Manipulation

Function	Availability	Use
Absolute	View, Basic, Professional, Developer Suite	Forms the absolute value of real numbers, complex numbers or time spans.
AdjustUnit	Basic, Professional, Developer Suite	Adjusts the unit(s) of a data set to those of a different data set.
AssignHeader	View, Basic, Professional, Developer Suite	Assigns header information to a value.

Function	Availability	Use
AssignListElementNames	View, Basic, Professional, Developer Suite	Assigns element names to one or more elements in a list.
Bit	View, Basic, Professional, Developer Suite	Extracts a bit trace from a data set with digital data.
ChangeDataType	View, Basic, Professional, Developer Suite	Converts the data type of a data set.
ChangeUnit	View, Basic, Professional, Developer Suite	Converts or sets the unit(s) of a data set.
ChangeUnitSymbol	Basic, Professional, Developer Suite	Replaces one or more unit symbols of a data set.
Clip	Basic, Professional, Developer Suite	Clips the amplitude of a data set.
Clip2D	Basic, Professional, Developer Suite	Cuts surfaces at an edge or along a closed curve.
Concatenate	Basic, Professional, Developer Suite	Appends strings, scalar values, data series, data matrices, signals, signal series and space curves.
ConcatenateList	Basic, Professional, Developer Suite	Concatenates all elements of a list to a data series or signal, or the elements of several lists into a single list with data series or signals.
CurveToSurface	Basic, Professional, Developer Suite	Transforms a space curve into a surface (signal series with Z component).

Function	Availability	Use
Detrend	Basic, Professional, Developer Suite	Removes a constant, linear or adaptive trend from a data set.
ExtractSegments	Basic, Professional, Developer Suite	Extracts segments of different lengths from a data set and returns them as a list.
ImaginaryPart	Basic, Professional, Developer Suite	Forms the imaginary part of real or complex numbers.
List	View, Basic, Professional, Developer Suite	Generates a list.
ListToSeries	Basic, Professional, Developer Suite	Converts a list with items of the same structure into a data series, data matrix or signal series.
MeshGrid	Basic, Professional, Developer Suite	Produces a two-dimensional grid. Helpful for three-dimensional visualization of functions with two variables or functions with complex-valued arguments.
NextHighestInteger	Basic, Professional, Developer Suite	Determines the next highest integer for a floating point value.
NextLowestInteger	Basic, Professional, Developer Suite	Determines the next-lowest integer for a floating point value.
OrderCuts	Option Order Tracking	Cuts order curves (if necessary by RMS or maximum calculation in a line band) from an order spectrum.
Phase	Basic, Professional,	Forms the phase of complex numbers.

Function	Availability	Use
	Developer Suite	
PhaseUnwrap	Basic, Professional, Developer Suite	Unwraps phase angles for producing smoother phase responses.
PolarTransform	View, Basic, Professional, Developer Suite	Performs a polar transformation.
PrimaryListElement	View, Basic, Professional, Developer Suite	Takes the primary element from a list.
RealPart	Basic, Professional, Developer Suite	Forms the real part of real or complex numbers.
RemoveHeader	View, Basic, Professional, Developer Suite	Removes header information from a value.
Reshape	View, Basic, Professional, Developer Suite	Changes the dimension lengths and/or number of dimensions of a data series or a data matrix.
Round	Basic, Professional, Developer Suite	Rounds up to a predefined number of valid decimal places or to a multiple of a specified rounding interval.
RoundDown	Basic, Professional, Developer Suite	Rounds down to a number of decimal places that you can specify.
RoundUp	Basic, Professional, Developer Suite	Rounds up to a number of decimal places that you can specify.
SearchListElements	Basic, Professional,	Searches for one or more list elements in a named list.

Function	Availability	Use
	Developer Suite	
SeriesToList	View, Basic, Professional, Developer Suite	Converts a one or two dimensional data set into a list whose number of elements corresponds to the number of values or columns in the data set.
SignalToSeries	View, Basic, Professional, Developer Suite	Transforms a signal into a signal series or a data series in a data matrix whose number of columns corresponds to the number of values in the data set.
Sort	Basic, Professional, Developer Suite	Sorts a data set or provides a sort index to a data set.
SplitIntoSegments	Basic, Professional, Developer Suite	Splits a data set into segments of different lengths and returns them as a list.
SurfaceToCurve	Basic, Professional, Developer Suite	Transforms a surface (signal series with Z component) into a space curve.
XOffsetScale	View, Basic, Professional, Developer Suite	Scales the X component of a signal or signal series.
XShift	View, Basic, Professional, Developer Suite	Shifts the X component of a signal or signal series so that it starts with a given initial value.

Dialog Boxes

Function	Availability	Use
ChooseFile	Basic, Professional, Developer Suite	Opens a dialog box for choosing a file.
Input	Basic, Professional, Developer Suite	Opens a dialog box for entering an expression.
MessageBox	Basic, Professional, Developer Suite	Displays a message box.
TextInput	Basic, Professional, Developer Suite	Opens a dialog box for entering text.
TimeInput	Basic, Professional, Developer Suite	Opens a dialog box for entering a date and time.
TimeSpanInput	Basic, Professional, Developer Suite	Opens a dialog box for entering a time span.

Envelopes

Function	Availability	Use
LowerEnvelope	Basic, Professional, Developer Suite	Determines a lower envelope for a signal or the indices of points in a signal that belong to its lower envelope.
UpperEnvelope	Basic, Professional, Developer Suite	Determines an upper envelope for a signal or the indices of points in a signal that belong to its upper envelope curve.

Event Isolation

Function	Availability	Use
Bursts	Basic, Professional, Developer Suite	Searches for beginnings and/or ends of bursts in a data set.
Extrema	Basic, Professional, Developer Suite	Searches for local minima and/or maxima in a data set.
GlobalExtrema	Basic, Professional, Developer Suite	Searches for global minima and/or maxima in a data set.
GlobalMaximum	Basic, Professional, Developer Suite	Searches for the global maximum in a data set.
GlobalMinimum	Basic, Professional, Developer Suite	Searches for the global minimum in a data set.
IndexAfter	Basic, Professional,	Performs a sequential operation for two index data sets.

Function	Availability	Use
	Developer Suite	
IndexAnd	Basic, Professional, Developer Suite	Performs a Boolean AND operation for two index data sets.
IndexBefore	Basic, Professional, Developer Suite	Performs a sequential operation for two index data sets.
IndexNot	Basic, Professional, Developer Suite	Forms the complement of an index data set.
IndexOr	Basic, Professional, Developer Suite	Performs a logical Boolean OR operation for two index data sets.
IndexSegments	Basic, Professional, Developer Suite	Searches for segment beginnings and/or ends in an index data series.
IndexSort	Basic, Professional, Developer Suite	Sorts an index data set so that it becomes ascending.
Intersections	Basic, Professional, Developer Suite	Determines the exact intersecting points of two data sets or the level crossings of one data set.
LevelCrossings	Basic, Professional, Developer Suite	Searches a data set for level crossings.
NegativePeaks	Basic, Professional, Developer Suite	Searches in a data set for negative peaks.
NextBurst	Basic, Professional,	Searches for the next beginning and/or end of a burst from a given

Function	Availability	Use
	Developer Suite	position onwards.
NextExtremum	Basic, Professional, Developer Suite	Searches for the next extreme from a given position onwards.
NextGlobalExtremum	Basic, Professional, Developer Suite	Searches for the next global extreme from a given position onwards.
NextLevelCrossing	Basic, Professional, Developer Suite	Searches for the next level crossing from a given position onwards.
NextNegativePeak	Basic, Professional, Developer Suite	Searches in a data set for the next negative peak starting from a predefined position.
NextPositivePeak	Basic, Professional, Developer Suite	Searches in a data set for the next positive peak starting from a predefined position.
NextSlope	Basic, Professional, Developer Suite	Searches for the next slope of a specified minimum steepness from a certain position onwards.
NextSlopeAtLevel	Basic, Professional, Developer Suite	Searches for the next slope of a specified minimum steepness through a level from a certain position onwards.
NextTimePeriod	Basic, Professional, Developer Suite	Searches a data set with calendar time values for the next value that lies in a periodic time segment.
NextValueAboveLevel	Basic, Professional, Developer Suite	Searches for the next value above a certain level.

Function	Availability	Use
NextValueBelowLevel	Basic, Professional, Developer Suite	Searches for the next value below a certain level.
NextValueInBurst	Basic, Professional, Developer Suite	Searches for the next value in a burst starting from a given position.
NextValueInInterval	Basic, Professional, Developer Suite	Searches for the next value in a particular interval or for the next void value.
NextValueInSpike	Basic, Professional, Developer Suite	Searches for the next value in a spike starting from a given position.
PositivePeaks	Basic, Professional, Developer Suite	Searches in a data set for positive peaks.
SearchValue	Basic, Professional, Developer Suite	Searches for one or more values in a data set.
Slopes	Basic, Professional, Developer Suite	Searches for slopes of a specified minimum steepness in a data set.
SlopesAtLevel	Basic, Professional, Developer Suite	Searches within a data set for slopes of a specified minimum steepness that have to pass through a particular level.
TimePeriods	Basic, Professional, Developer Suite	Searches a data set with calendar time values for periodic time segments.
Trigger	Basic, Professional, Developer Suite	Represents a Schmitt trigger comparator, which performs a threshold observation for a data set.

Function	Availability	Use
ValuesAboveLevel	Basic, Professional, Developer Suite	Searches a data set for values above a certain level.
ValuesBelowLevel	Basic, Professional, Developer Suite	Searches a data set for values below a certain level.
ValuesInBursts	Basic, Professional, Developer Suite	Searches within a data set for values that belong to a burst.
ValuesInInterval	View, Basic, Professional, Developer Suite	Searches a data set for values that lie in a particular interval, or searches for void values.
ValuesInSpikes	Basic, Professional, Developer Suite	Searches a data set for values that belong to a spike.

Filtering And Smoothing

Function	Availability	Use
AmplitudeResponse	Option Digital Filters	Calculates the amplitude response from the filter coefficients.
CFCFilter	Option Digital Filters	Filters a data set using a CFC filter. CFC is short for Channel Frequency Class.
DCRemovalFilter	Basic, Professional, Developer Suite	Removes the DC offset (DC bias) with the help of a digital (recursive) high-pass filter.
Filter	Basic, Professional,	Filters a data set with a Finite Impulse Response (FIR) or an Infinite Impulse

Function	Availability	Use
	Developer Suite	Response (IIR) filter.
FIRFilterEquiripple	Option Digital Filters	Calculates the impulse response of an FIR band filter using the Equiripple algorithm.
FIRFilterWindow	Option Digital Filters	Calculates an FIR filter's impulse response using windowing.
GaussianFilter	Option Digital Filters	Filters a signal with a Gaussian filter. Typically used for smoothing and as a filter for measuring and analyzing roundness.
GroupDelay	Option Digital Filters	Calculates the group delay from the filter coefficients.
HarmonicRemovalFilte r	Option Order Tracking	Removes harmonic components from time signals (harmonic removal). To do this, the data is converted from the time domain to the revolution domain, where cycle averaging and subtraction provides the desired harmonic removal.
IIRFilter	Basic, Professional, Developer Suite	Calculates the numerator and denominator coefficients or the poles and zeros of an infinite impulse response filter (IIR filter).
IIRNotchFilter	Option Digital Filters	Calculates the numerator and denominator coefficients of a second-order IIR notch filter that can be used to filter out frequencies within a narrow frequency range (narrow-band notch filter).
IIRPeakFilter	Option	Calculates the numerator and denominator coefficients of a second-

Function	Availability	Use
		order IIR peak filter (also called a resonant filter) that can be used to block out frequencies outside a narrow frequency range (narrow-band bandpass filter).
LoessFilter	Option Digital Filters	Filters a data set with a Loess smoothing filter (locally weighted regression scatter plot smoothing).
LowessFilter	Option Digital Filters	Filters a data set with a Lowess smoothing filter (locally weighted regression scatter plot smoothing).
PhaseResponse	Option Digital Filters	Calculates the phase response from the filter coefficients.
SavitzkyGolayDerivative	Basic, Professional, Developer Suite	Calculates the derivative of a data set with a Savitzky-Golay smoothing filter (least squares derivative). Effective method for determining the smoothed derivative of a noisy data set.
SavitzkyGolayFilter	Option Digital Filters	Filters the data set with a Savitzky-Golay smoothing filter (also known as Least-Squares Smoothing).
Smooth	Basic, Professional, Developer Suite	Smoothens a data set by calculating a floating mean value.
VibrationFrequencyWeighting	Option Human Body Vibrations	Filters an acceleration signal with a band-limited weighting filter for analyzing whole-body and hand-transmitted vibrations.

FPScript

Function	Availability	Use
Conditional	View, Basic, Professional, Developer Suite	Data analysis. Functional variant of the If...Then...Else statement.
Execute	Basic, Professional, Developer Suite	Interprets a string as FPScript code and executes it.

General Mathematics

Function	Availability	Use
Erf	Basic, Professional, Developer Suite	Calculates the Gaussian error function.
Factorial	Basic, Professional, Developer Suite	Calculates the factorial N! of a natural number.
Gamma	Basic, Professional, Developer Suite	Calculates the gamma function for real-valued arguments.
J0	Basic, Professional, Developer Suite	Calculates the Bessel function of the first kind of the order 0.
J1	Basic, Professional, Developer Suite	Calculates the Bessel function of the first kind of the order 1.
Jn	Basic, Professional, Developer Suite	Calculates the Bessel function of the first kind with the order specified.

Function	Availability	Use
LogGamma	Basic, Professional, Developer Suite	Calculates the natural logarithm of the absolute value of the gamma function for real-valued arguments.
Product	Basic, Professional, Developer Suite	Calculates the product of all of the values in a data series or the products of all rows in a data matrix.
Sinc	Basic, Professional, Developer Suite	Calculates the sinc function Sin(*Angle*)/*Angle*.
Sqrt	View, Basic, Professional, Developer Suite	Calculates the square root.
TrackDistance	View, Basic, Professional, Developer Suite	Calculates the distance between locations on Earth.
Y0	Basic, Professional, Developer Suite	Calculates the Bessel function of the second kind of the order 0.
Y1	Basic, Professional, Developer Suite	Calculates the Bessel function of the second kind of the order 1.
Yn	Basic, Professional, Developer Suite	Calculates the Bessel function of the second kind with the order specified.

Logarithm And Exponent

Function	Availability	Use
dB	View, Basic, Professional, Developer Suite	Converts an amplitude or power signal into decibels.
Exp	Basic, Professional, Developer Suite	Raises e to the power of the specified exponent.
Log	Basic, Professional, Developer Suite	Calculates the logarithm to the base e (natural logarithm).
Log10	View, Basic, Professional, Developer Suite	Calculates the logarithm to base 10.

Matrices

Function	Availability	Use
InverseMatrix	Basic, Professional, Developer Suite	Inverts a matrix.
MatrixMultiplication	Basic, Professional, Developer Suite	Multiplies matrices with vectors and matrices in all combinations.
TransposeMatrix	Basic, Professional, Developer Suite	Transposes a matrix.

Signal Analysis

Function	Availability	Use
ACF	Basic, Professional, Developer Suite	Calculates the autocorrelation function of a signal. Describes the similarity of a signal to itself.
CCF	Basic, Professional, Developer Suite	Calculates the cross-correlation function of two signals. Used to describe the similarity between two signals.
Convolution	Basic, Professional, Developer Suite	Calculates the convolution product of two signals.
CumulativeSum	Basic, Professional, Developer Suite	Calculates the cumulative sum of a data set.
Derivative	View, Basic, Professional, Developer Suite	Calculates the first derivative.
Diff	View, Basic, Professional, Developer Suite	Calculates differences of neighboring Y values as well as right and left-sided difference quotients.
Frequency	Basic, Professional, Developer Suite	Determines the frequency of a data set.
ImpulseToFrequency	Basic, Professional, Developer Suite	Transforms an impulse signal or step signal into a frequency signal.
Integral	View, Basic, Professional, Developer Suite	Calculates the integral.
Period	Basic, Professional, Developer Suite	Determines the period of a data set.
		Calculates various quantities, such as principal stresses or principal strains, from two or three strain signals measured using a strain gauge rosette. The calculation is

Signal Generation

Function	Availability	Use
Chirp	Basic, Professional, Developer Suite	Calculates a swept-frequency cosine, i.e. a cosine signal with a variable frequency.
Dirichlet	Basic, Professional, Developer Suite	Calculates the Dirichlet function, i.e. the periodic sinc function.
Noise	Basic, Professional, Developer Suite	Generates equally, normally or exponentially distributed random numbers.
Sawtooth	Basic, Professional, Developer Suite	Calculates the sawtooth function with a period of 2π, amplitude of 1 and given width.
Series	Basic, Professional, Developer Suite	Forms a data series with linear ascending or descending values.
Signal	View, Basic, Professional, Developer Suite	Composes a signal, signal series or a space curve from individual components.
Square	Basic, Professional, Developer Suite	Calculates the square function with a period of 2π, amplitude of 1 and given duty cycle.
StraightLine	Basic, Professional, Developer Suite	Sets up a straight line equation based on the Y intercept and slope or based on two XY value pairs and evaluates this at the specified X positions.

Signal Sampling

Function	Availability	Use
BlockCompress	Basic, Professional, Developer Suite	Reduces the number of values in a data set using block operations or splits it into a list of segments.
DeltaCompress	Basic, Professional, Developer Suite	Reduces the number of values in a data set through delta compression.
Expand	Basic, Professional, Developer Suite	Increases the number of values in a data set.
LinearInterpolation	Basic, Professional, Developer Suite	Carries out a linear interpolation of a data set.
Reduce	View, Basic, Professional, Developer Suite	Reduces the number of values in a data set.
Resample	Basic, Professional, Developer Suite	Increases or reduces a data set's sampling rate.
RevolutionSyncSampling	Option Order Tracking	Transforms a signal sampled over time into the revolution domain, i.e. the time signal is transformed into an equidistantly sampled rotational speed range by resampling. Effective method for performing order tracking, since the frequency spectrum of the signal converted to the revolution domain directly provides the order spectrum. In the same way, ordinary bandpass filtering in the revolution domain can be used to calculate the (temporal) order curves directly.
Sample	Basic, Professional, Developer Suite	Samples a signal using linear interpolation.

Spectral Analysis

Function	Availability	Use
AnalyticSignal	Option Spectral Analysis	Transforms a real signal into an analytic signal whose imaginary part results in the Hilbert transform. Frequently used to calculate instantaneous amplitude or instantaneous frequency as well as for the demodulation of signals.
ApplyWindow	Basic, Professional, Developer Suite	Applies a window to the argument.
ARMASpectrum	Option Spectral Analysis	Computes the Autoregressive-Moving-Average (ARMA) high-resolution frequency estimation spectrum.
ARSpectrum	Option Spectral Analysis	Computes the Autoregressive or AR Spectral estimators spectrum.
CepstralAnalysis	Option Spectral Analysis	Computes the cepstrum or its minimum phase reconstruction.
Coherence	Option Spectral Analysis	Computes the coherence spectrum between two data sets.
CrossPeriodogram	Option Spectral Analysis	Computes the windowed Fourier cross-periodogram between two data sets.
CrossSpectrum	Option Spectral Analysis	Computes the windowed Fourier cross-spectrum between two data sets.
CWTSpectrum	Option Spectral Analysis	Computes the Continuous Wavelet Transform (CWT) time-frequency

Function	Availability	Use
		spectrum.
DataWindow	Option Spectral Analysis	Creates a uniformly spaced data taper of a given size.
EigenSpectrum	Option Spectral Analysis	Computes the EigenAnalysis Spectral estimators spectrum.
FFTn	Basic, Professional, Developer Suite	Computes the complex Fourier transform of the argument.
FourierSpectrum	Basic, Professional, Developer Suite	Computes the windowed Fourier spectrum.
FourierSpectrumUneven	Option Spectral Analysis	Computes the windowed Fourier spectrum for unevenly sampled data.
HarmonicEstimation	Option Spectral Analysis	Estimates the least-squares harmonic components of a signal.
Hilbert	Option Spectral Analysis	Calculates the Hilbert transform. Frequently used to calculate instantaneous amplitude or instantaneous frequency as well as for the demodulation of signals.
IFFTn	Basic, Professional, Developer Suite	Computes the complex inverse Fourier transform of the argument.
IRFFTn	Basic, Professional, Developer Suite	Computes the real inverse Fourier transform of the argument.
MultitaperSpectrum	Option Spectral Analysis	Computes the Slepian or DPSS Multitaper Fourier spectrum.

Function	Availability	Use
OctaveAnalysis	Basic, Professional, Developer Suite	Performs an octave analysis for an amplitude spectrum.
Periodogram	Basic, Professional, Developer Suite	Computes the segmented-overlapped Fourier spectrum.
SDOFResponse	Option Spectral Analysis	Calculates the responses of individual single-degree-of-freedom (SDOF) systems of the SRS model from the signal of an acceleration sensor.
SRS	Option Spectral Analysis	Calculates one or more shock response spectra (SRS) from the signal of an acceleration sensor.
SRSFromSDOFResponse	Option Spectral Analysis	Calculates the shock response spectrum (SRS) from the responses of the SRS model's individual SDOF systems, which can be calculated using the SDOFResponsefunction.
STFTSpectrum	Professional, Developer Suite	Computes the Short-Time Fourier Transform (STFT) time-frequency spectrum.
ThirdOctaveAnalysis	Basic, Professional, Developer Suite	Carries out a third octave analysis for an amplitude spectrum.
TransferFunction	Option Spectral Analysis	Computes the transfer function between an input and an output data set.
VarWindow	Option Spectral Analysis	Creates a data taper compatible with a data set.

Statistics

Function	Availability	Use
AbsoluteDeviationFromMean	Basic, Professional, Developer Suite	Determines the mean absolute deviation from the mean value of a data set. The displacements from the mean are determined for all values of the data set and the average is calculated from these.
AbsoluteDeviationFromMedian	Basic, Professional, Developer Suite	Determines the mean absolute deviation from the median of a data set. This is the average of the deviations from the median. The median is the value that lies exactly in the center of a data series after sorting. If the data set contains an even number of values, the average of the two center values is formed.
ANOVA	Option Statistics	Carries out a Fisher analysis of variance. You can either calculate an ANOVA table or perform an F-test. The F-test specifies whether the mean values of several samples are significantly different or not. The ANOVA table provides characteristic quantities for the analysis of variance.
BartlettTest	Option Statistics	Carries out a Bartlett variance test.
BoxPlot	Basic, Professional, Developer Suite	Determines the statistical parameters for displaying a box plot.
ChiSquareTest	Option Statistics	Carries out a Chi-square goodness-of-fit test. The test checks whether the sample passed corresponds to the normal or exponential distribution specified.
CoefficientOfVariation	Basic, Professional, Developer Suite	Calculates the absolute or relative coefficient of variation for a data set.
	Option	Calculates a confidence interval for the mean or the variance of the

Strings

Function	Availability	Use
Format	Basic, Professional, Developer Suite	Formats a value and returns the result as a string.
RemoveDuplicates	Basic, Professional, Developer Suite	Removes duplicates from a data series of strings.
SearchStrings	Basic, Professional, Developer Suite	Searches in a data series for a string and returns the indices of the matches as a data series.
StringConcat	Basic, Professional, Developer Suite	Appends multiple strings to each other.
StringFind	Basic, Professional, Developer Suite	Searches for a substring and returns its position.
StringLeft	Basic, Professional, Developer Suite	Takes a substring out of a string starting from the left.
StringLength	Basic, Professional, Developer Suite	Determines the number of characters in a string.
StringLowerCase	Basic, Professional, Developer Suite	Converts a string into lower case.
StringMid	Basic, Professional, Developer Suite	Takes a substring out of a string starting from a particular position.

Function	Availability	Use
StringReplace	Basic, Professional, Developer Suite	Searches for a substring and replaces all occurrences with a different string.
StringReverseFind	Basic, Professional, Developer Suite	Searches for the last match in a substring and returns its position.
StringRight	Basic, Professional, Developer Suite	Pulls a substring out of a string starting from the right.
StringSet	Basic, Professional, Developer Suite	Replaces a substring starting from a predefined position in a string.
StringUpperCase	Basic, Professional, Developer Suite	Converts a string into upper case.

Surface Interpolation

Function	Availability	Use
BicubicSpline	Basic, Professional, Developer Suite	Interpolates a two-dimensional data set through a bicubic spline surface and samples this spline surface at definable points.
GriddedSurface	Basic, Professional, Developer Suite	Interpolates a two-dimensional data set using Natural Neighbor interpolation and samples this at definable points.
Isoline	Basic, Professional, Developer Suite	Determines one or more isolines from a data matrix or signal series.
ScatteredSurface	Basic, Professional, Developer Suite	Models a surface using the Natural Neighbor method for the sampling points specified as a space curve and evaluates this at definable grid points.
SmoothingSpline2D	Basic, Professional, Developer Suite	Interpolates a two-dimensional data set through a grid of compensating spline curves and samples these curves at definable points.
Spline2D	Basic, Professional, Developer Suite	Interpolates a two-dimensional data set through a grid of spline curves and samples these curves at definable points.

Trigonometry

Function	Availability	Use
ArcCos	Basic, Professional, Developer Suite	Calculates the inverse cosine of the argument. The result is output in radians.
ArcSin	Basic, Professional, Developer Suite	Calculates the inverse sine of the argument. The result is output in radians.
ArcTan	Basic, Professional, Developer Suite	Calculates the inverse tangent of the argument.
ArcTan2	Basic, Professional, Developer Suite	Calculates the arctangent with two arguments.
Cos	Basic, Professional, Developer Suite	Calculates the cosine of an angle.
CosHyp	Basic, Professional, Developer Suite	Calculates the hyperbolic cosine of a number.
Sin	Basic, Professional, Developer Suite	Calculates the sine of an angle.
SinHyp	Basic, Professional, Developer Suite	Calculates the hyperbolic sine of a number.
Tan	Basic, Professional, Developer Suite	Calculates the tangent of an angle.
TanHyp	Basic, Professional, Developer Suite	Calculates the hyperbolic tangent of a number.

Void Values

Function	Availability	Use
InterpolateVoidValues	Basic, Professional, Developer Suite	Interpolates or extrapolates void floating point values in a data set.
RemoveVoidValues	Basic, Professional, Developer Suite	Removes void values from a data set.
SearchVoidValues	Basic, Professional, Developer Suite	Searches for void floating point values in a data set and passes their positions.

Python Properties

Property	Available In	Used For
assign_header	Professional, Developer Suite	Read access to the header information attribute of a data object. Read/write access to the header information attribute of the current formula.
assigned_x	Professional, Developer Suite	Read access to the name of the data object assigned as the X component. Read/write access to the name of the data object assigned as the X component of the current formula or FlexPro data.
assigned_z	Professional, Developer Suite	Read access to the name of the data object assigned as the Z component. Read/write access to the name of the data object assigned as the Z component of the current formula or FlexPro data.
author	Professional, Developer Suite	Read access to the name of a data object author. Read/write access to the name of the author of the current formula or FlexPro data.

Property	Available In	Used For
calculations	Professional, Developer Suite	Read access to a data object calculation.
comments	Professional, Developer Suite	Read access to the object's comments.
comments_x	Professional, Developer Suite	Read access to the comments on a data object's X component. Read/write access to the comments of the X component of the current formula or FlexPro data.
comments_y	Professional, Developer Suite	Read access to the comments on the data object's Y component. Read/write access to the comments of the Y component of the current formula or FlexPro data.
comments_z	Professional, Developer Suite	Read access to the comments on the data object's Z component. Read/write access to the comments of the Z component of the current formula or FlexPro data.
creation_time	Professional, Developer Suite	Read access to an object's creation time. Read/write access to the creation time of the current formula or FlexPro data.
data	Professional, Developer Suite	Read access to the data of a data object. Read/write access to the data of the current formula.
formula	Professional, Developer Suite	Read access to the code of a formula.
full_name	Professional, Developer Suite	Read access to the name and the path in the project database of an object.
hyperlink	Professional, Developer Suite	Read access to the hyperlink of an object. Read/write access to the hyperlink of the current formula or FlexPro data.
lower_range_limit_x	Professional, Developer Suite	Read access to the lower range limit of the X component of a data object. Read/write access to the lower range limit of the X component of the current formula or FlexPro data.
lower_range_limit_y	Professional, Developer Suite	Read access to the lower range limit of the Y component of a data object. Read/write access to the lower range limit of the Y component of the current formula or FlexPro data.

Property	Available In	Used For
lower_range_limit_z	Professional, Developer Suite	Read access to the lower range limit of the Z component of a data object. Read/write access to the lower range limit of the Z component of the current formula or FlexPro data.
modification_time	Professional, Developer Suite	Read access to the modification time of an object or FlexPro data.
name	Professional, Developer Suite	Read access to the name of an object. Read/write access to the name of FlexPro data.
origin	Professional, Developer Suite	Read access to the origin of a data object. Read/write access to the origin of the current formula or FlexPro data.
parameters	Professional, Developer Suite	Read access to a parameter of an object. Read/write access to the value of a parameter of the current formula.
path	Professional, Developer Suite	Read access to the path in the project database of an object or FlexPro data.
quantity_x	Professional, Developer Suite	Read access to the name of the physical quantity of a data object's X component. Read/write access to the name of the physical quantity of the X component of the current formula or FlexPro data.
quantity_y	Professional, Developer Suite	Read access to the name of the physical quantity of a data object's Y component. Read/write access to the name of the physical quantity of the Y component of the current formula or FlexPro data.
quantity_z	Professional, Developer Suite	Read access to the name of the physical quantity of a data object's Z component. Read/write access to the name of the physical quantity of the Z component of the current formula or FlexPro data.
timestamp_x	Professional, Developer Suite	Read access to the timestamp on a data object's X component. Read/write access to the timestamp of the X component of the current formula or FlexPro data.
timestamp_y	Professional, Developer Suite	Read access to the timestamp of the Y component of a data object. Read/write access

Property	Available In	Used For
		to the timestamp of the Y component of the current formula or FlexPro data.
timestamp_z	Professional, Developer Suite	Read access to the timestamp on the data object's Z component. Read/write access to the timestamp of the Z component of the current formula or FlexPro data.
unit	Professional, Developer Suite	Read access to the unit of FlexPro data.
unit_x	Professional, Developer Suite	Read access to the unit of a data object's X component. Read/write access to the unit of the X component of the current formula or FlexPro data.
unit_y	Professional, Developer Suite	Read access to the unit of a data object's Y component. Read/write access to the unit of the Y component of the current formula or FlexPro data.
unit_z	Professional, Developer Suite	Read access to the unit of a data object's Z component. Read/write access to the unit of the Z component of the current formula or FlexPro data.
upper_range_limit_x	Professional, Developer Suite	Read access to the upper range limit of the X component of a data object. Read/write access to the upper range limit of the X component of the current formula or FlexPro data.
upper_range_limit_y	Professional, Developer Suite	Read access to the upper range limit of the Y component of a data object. Read/write access to the upper range limit of the Y component of the current formula or FlexPro data.
upper_range_limit_z	Professional, Developer Suite	Read access to the upper range limit of the Z component of a data object. Read/write access to the upper range limit of the Z component of the current formula or FlexPro data.
value	Professional, Developer Suite	Read access to the value of a data object or FlexPro data. Read/write access to the hyperlink of the current formula or FlexPro data.
x	Professional, Developer Suite	Read access to the X component of FlexPro data with a composite data structure.

Property	Available In	Used For
y	Professional, Developer Suite	Read access to the Y component of FlexPro data with a composite data structure.
z	Professional, Developer Suite	Read access to the Z component of FlexPro data with a composite data structure.

Python Functions

Function	Available In	Used For
flexpro.assign_header	Professional, Developer Suite	Assigns header information to a flexpro.Data object.
flexpro.call	Professional, Developer Suite	Returns the data of a formula or a data set or calls an FPScript or Python function with arguments and returns the result.
flexpro.Data	Professional, Developer Suite	Creates a flexpro.Data object that represents a simple or composite FlexPro data structure with value and unit.
flexpro.list	Professional, Developer Suite	Creates a flexpro.Data object that represents a FlexPro data structure "List".
flexpro.named_list	Professional, Developer Suite	Creates a flexpro.Data object that represents a FlexPro data structure "List" with named list elements.
flexpro.object	Professional, Developer Suite	Creates a reference to an object in the project database.
flexpro.remove_header	Professional, Developer Suite	Removes header information from a flexpro.Data object.
flexpro.run	Professional, Developer Suite	Executes an FPScript statement and returns the result as a flexpro.Data object.

6 Presenting Data

To present data and calculated results, FlexPro offers you the following presentation objects: 2D and 3D Diagrams 381, Column, Row and Cell Tables 437, Text 459 and Media 464. You can either export these objects directly or insert them in documents 470 and worksheets 493.

Presentation objects have a definable fixed size corresponding to a rectangular area that the object adopts when it is placed into a document, for instance. However, if you open a presentation object in its own window, then its size is automatically adapted to fit the size of the window when you enable the appropriate option.

Presentation and Document Templates

You can create templates for diagrams, tables, texts and documents and store these in the current project database or in a template database. You can then choose a previously saved template in the various wizards for creating objects.

You can embed or link diagrams, tables, text and media in document templates. Linked objects are replaced by the selected objects when the template is used at a later date. The links in the template serve as placeholders for the diagrams, tables, text and media to be inserted. For embedded presentation objects, however, the data links are replaced. If a document template contains an embedded diagram with two curves, for instance, then the selected data objects are entered in the curves when the template is used.

FlexPro makes a distinction between static and dynamic table and diagram templates. If you use a dynamic template, then a curve or a table column or table cell is created for each selected data set. In a static template, only the existing curves, columns or cells are linked to the selected data objects. Excess data objects, if present, are ignored. You define the type of template when saving it.

Linking and Embedding

You can embed presentation objects into documents and worksheets or insert a link to a project database object. An embedded object is a component of the document or worksheet and can only be used in the location where it has been inserted. Links to embedded objects cannot be created.

If you would like to use the same object in several places, such as in a document and in a worksheet, then you have to set it up in the project database and insert a link to it at the points where it is to appear. When selected, linked objects are displayed with a dashed border and embedded objects are displayed with a solid border.

If you edit an object with links to it, then the changes affect all links. Updating various links does not take place automatically, but instead must be activated using the Update 105 command. The exception here is the size of the object. You can set the dimensions for each link individually. A diagram can therefore be displayed, for example, in a different size in the worksheet than the size in the document.

Not only can you insert links into FlexPro documents and worksheets, but you can also insert them into documents you created using a different application. For example, you can insert a link to a diagram into a text document. Your word processing program just needs an OLE interface. You can also update and edit these links.

You cannot embed presentation objects in external applications, because the FlexPro objects can only be used practically within the environment of the data and analyses on which they are based. This then only consists of the image information and cannot be edited subsequently with FlexPro.

Drawing in Presentation Objects

You can insert any number of drawings into all presentation objects except the text object in order to highlight or label interesting parts of a curve, for instance. You can use the same set of shapes as in a document. All drawings that you create within a presentation object appear in the foreground. If you enlarge a diagram, the drawing will automatically be enlarged as well.

You can display drawing grids and use alignment guides to help you with your drawing.

Updates multipage diagrams, tables and texts selected.

FlexPro can split 2D diagrams, column tables, row tables and text over several pages in a document. You can use this option, for instance, to display longer continuous curves in a diagram, to display larger data sets in table, or to display longer text.

In a 2D diagram, page splitting and numbering depends on the options selected on the Page Splitting tab of the Properties dialog box 107 for the diagram. For a column table, the space available on the first page is used and then wraps automatically onto the additional pages if not all of the data can be displayed.

You can place the diagram, table or text on any page in the document. The additional pages automatically appear at the same position on the subsequent pages of the document. The number of pages in the document is automatically adjusted by FlexPro so that all pages of the presentation object are displayed. You can adjust the position of the object for each page. In the case of a multipage table or multipage text, you can also set the height of each page individually.

Note: You can only display entire multipage tables, diagrams and text in FlexPro documents and in the object window. In the worksheet or external applications, only the first page is displayed.

Embedded FPScript

On the Data tab of the Properties dialog box for presentation objects, specify the path name of the formula or data set providing the data to be displayed, for instance, as a curve in a diagram. The easiest thing to do is to specify only the name of the data set. However, if you prefer, you can enter any FPScript expression. This embedded FPScript is then interpreted as a formula located in the same folder where the presentation object is also located.

The following FPScript expression can be used, for instance, to display the last signal of a signal series as a curve:

```
SignalSeries[-1]
```

To work with the FPScript expression, FlexPro dynamically creates a formula embedded in the presentation object, which is then compiled and executed. UnitY, UnitX, UnitZ, CommentsY, CommentsX, and CommentsZ. FlexPro attempts to create relevant entries for these properties automatically by taking these from the last data object used in the formula. In the example above, this would be the `Signal Series` data set. For the most frequent situation where the embedded FPScript only accesses a data set, this provides the correct entries, such as for axis labeling. If this is not the case, you will have to correct the relevant property by write-accessing it. The following FPScript, for instance, normalizes the data of a signal to 100 % and corrects the Y comments accordingly:

```
This.CommentsY = "Normalized Signal"; Signal / Maximum(Signal) * 100.
```

An embedded FPScript, however, is not used only for accessing data for presentation, but is also used to access attributes that are to be displayed as an axis label or in a table cell, for instance. To do this, use a formatter that has an embedded FPScript expression appended to it in the relevant text. This type of formatter always starts with a % sign and determines how the data is to be presented as well as from where the data originates. A table cell that displays a maximum value could, for instance, contain the following:

Maximum of %{`Signal.CommentsY`}: %&{`Maximum(Signal)`}

Here, two fields 511 are embedded in the text, the first contains an FPScript expression that provides the comment of the data set, and the second calculates the maximum. The part of the field before the curly bracket '{' with a % sign is also called the Formatter 503, which controls the output format. The & character in the second formatter specifies, for instance, that the data should be output with the unit. Depending on the format selected, additional codes appear after the % sign.

By default, the following text is entered on the Axis Labeling tab for the axis of a 2D diagram:

```
%{.Data.YValueObject(%<ListElement>).NameOrQuantityOrComments(.Data.YComponent)}
```

Before the FPScript code is compiled, the `%<ListElement>` placeholder is replaced by the number of the curve that is currently being processed if the curve displays a list with several data sets. The FPScript expression uses the curve's Automation object model to access the attributes of the data object that is used for the curve's Y component. To be more precise, the embedded FPScript formula specified for the Y component of the curve is being accessed. This in turn usually takes the attributes from the data object to which it refers.

6.1 2D and 3D Diagrams

You can use diagrams to make attractive and expressive presentations of data and results of calculations. FlexPro offers a wide variety of diagram types. These are the result of the flexible structure of just two objects: the 2D diagram and the 3D diagram.

The 2D diagram presents data in a rectangular two-dimensional plane, while the 3D diagram presents the data in a cube projected onto the plane. The 2D diagram is especially ideal for presenting signals and data series, and the 3D diagram is ideal for presenting signal series, data matrices and space curves. The diagram plane, i.e. the rectangular area in which the curves are displayed, is surrounded by a border in which the axis labels and the legend are displayed. The size of the border area can be adjusted.

For each data set that you want to display in a diagram, you have to add a curve to the diagram. This refers to the data and determines how the data is to be displayed. The variety of optional diagram types is the result of the large range of optional display formats coupled with their ability to be combined. For the curve of a 2D diagram, you can, for instance, combine any of the following display formats: connection line, symbols, error indicators, labeling, columns/bars, and fill.

Diagrams consist of the following elements, which are explained in the subsequent sections: axis, grid, curve, legend and color legend.

You can set the style of diagrams using the Diagram style and Color palette on the first page of the wizard. These attributes then control all style elements in the diagram that are set to Automatic. If you set a color on a data set Format tab, then this color is used instead of the corresponding color scheme when the data set is displayed in a diagram. However, to do this, the option Use colors of data sets must be selected on the Style page of the diagram.

Axis

The diagram plane or space is spanned by axes, which determine the scaling of the data to be displayed. The 2D diagram uses at a minimum one vertical Y axis and one horizontal X axis, and the 3D diagram also uses a Z axis, which is pointed toward the viewer. Please note that, contrary to the typical labeling in mechanics, the vertical axis in FlexPro is also called the Y axis in a 3D diagram. While 3D diagrams use a fixed number of three axes, one each for the X, Y, and Z directions, several Y and X axes can be used for a 2D diagram. This makes it possible to display curves with different physical units in the same diagram. If you use more than one Y axis, then you can alternatively stack them on top of each other.

Dynamic Number of Axes

If you have a list containing several data sets with different units above an axis and display the list in a 2D Diagram, FlexPro automatically displays several scales: one per unit. You can also refer to the Dynamic Axes and Curves option in the Diagram Wizard.

Axis Type

To scale the axes of 2D and 3D diagrams, you can use the linear, logarithmic with base 10, 2 or e, offset reciprocal, third octave, octave, probability, probit, logit and Weibull scale types.

With linear scaling you can choose whether you would like to specify a fixed division interval or whether the division interval should be determined automatically so that there are a fixed number of divisions.

The linear scaling with metric division option lets you establish a fixed relationship between the length of an axis and the physical unit of the curve displayed against this axis. You can set the scale freely. For example, if you set 0.1 V/mm as the scale and 1 V as the division interval, the divisions of the axis will be exactly 1 cm apart. If you enlarge the diagram and the length of the axis changes accordingly, the number of divisions will increase automatically. The distance between the divisions will remain constant.

By selecting Linear, division by data set or Logarithmic, division by data set you can determine at which positions on the axis divisions are to be drawn by using a data set or a formula. The procedure used depends on the data set or formula that you specify for labeling:

- Data series: The values are taken from the data series in sequential order, and a division is drawn at the corresponding position. All numerical data types are allowed. The process ends when all values have been read.
- Formula with one argument (function): The function is called with an ascending index as the argument. The first index is a null value. The result of the function determines the position of the axis division. All numerical data types are allowed for the result. The process ends when a value lies outside of the starting or end value of the axis, or when a void floating point value or the value `Empty` is passed.

The reciprocal scaling option uses the transformation X' = 1 / X.

Use Offset reciprocal scaling to convert from Kelvin to °C. The transformation is X' = 1 / (X + 273.15).

The probability scaling option is used to linearize normal distribution functions. If the Y axis of a diagram has been probability-scaled, these types of distribution functions appear as straight lines and can easily be compared.

Probit scaling option corresponds to probability scaling, although here standard deviations are considered, where 50 % corresponds to the value of 5. The value range of a probability or probit-scaled axis lies in the interval [0.0001 %, 99.9999 %].

Logit scaling is similar to probability scaling. As a transformation, however, the Logit function X' = X / (100 - X) is used. The value range of a logit-scaled axis is within the interval [0.1 %, 99.9 %].

The Weibull scaling linearizes the Weibull distribution. The transformation used here is Y' = ln(-ln(1 - Y/100)). You must use logarithmic scaling for the X-axis. The value range of a logit-scaled axis lies in the interval [0.1 %, 99.9 %].

The scaling types Third octave and Octave are logarithmic scaling options with the divisions corresponding to the third octave or octave series.

Unit

You can define an output unit for the axis. All curves are then converted to this unit before output. If you do not define an output unit, the axis is assigned the unit of the first curve whose data has a unit; all additional curves are transformed to this unit.

When the axis is displayed at both borders of a 2D diagram, you can assign a different unit to the second scaling, such as m/s for scaling on the left border and km/for scaling on the right border.

Scaling

You can set the scaling of the axis manually by specifying the physical values that correspond to both ends of the axis. Alternatively, you can specify data sets or FPScript formulas that provide end values. With Autoscaling these end values are automatically determined from the curves to be displayed. This means that the curves are displayed with optimal spread. If you select Data minimum/maximum the extreme values of the curves displayed on the axis are used. By selecting Data range lower limit/upper limit the data range values entered on the General tab of the data sets are used. You can exclude individual curves 420 from autoscaling. Autoscaling is often used with the Extend end values attribute. This means that the extreme values found are not used directly as axis end values; instead, a correction takes place first, so that the end values of the axis come to lie on a division. The options Symmetric scaling and Align zero point with that of the previous axis extend the start or end value of the axis so that the zero point is exactly in the middle of the axis or is aligned with that of the previous axis. The axis end values can be calculated dynamically using an FPScript formula. In 2D diagrams with several axes, you can also link the end values of neighboring axes.

If all of the diagram axes have the same physical unit, you can achieve an undistorted picture using isometric axis adjustment. This corrects the axis end values corrected so that they correspond to the axis length ratios. The isometric axis adjustment means for example, that with a 2D diagram, circular curves are also displayed as circles and not as ellipses.

Orientation

Usually the X axis is aligned from left to right, the Y axis is aligned from bottom to top and the Z axis is aligned from back to front. However, the orientation can be switched around for each axis. You can also add orientation arrows to the axes, which point in the direction of the increasing scalar values.

Axis Origin

You can define an axis origin for all 2D diagram axes and for the Y axis of a 3D diagram. For 2D diagrams, the axis origin determines the point of intersection of the corresponding axis of the axis system unless this axis is positioned at the edge of the diagram. If you have activated floor lines for the X axis, then a floor line parallel to the X axis is displayed at the height of the axis origin of the corresponding Y axis. You can choose whether the floor line should be drawn only for columns and surfaces or for any curves.

For a 3D diagram, the axis origin determines the location of the Floor, unless it was positioned at one of the end values of the axis. This floor determines, for instance, the plane on which the 3D columns are placed and is displayed as a frame with divisions.

Division

You can provide an axis with divisions and subdivisions that can be labeled with the values corresponding to their position on the axis.

The divisions are shown as small dashes on the axis, and you can change their type, length and color. The position of the divisions on the axis result from the axis scaling. You can either specify the amount that should correspond to one division interval or you specify the number of division intervals. It is also possible to specify the positions of the divisions as data series or, to calculate them using an FPScript function, which obtains the index of the division to be calculated as the argument.

With the division origin, you determine the value from which the division is to start at the top and bottom. This does not have to lie between the starting value and end value of the axis. When the subdivisions are applied, the division intervals are split up into a certain number of subdivisions. You can specify this number. Logarithmic axes are treated separately: With a division of 1 and either a number of subdivisions of 2 or a number of subdivisions of 8 and the Avoid overlaps option switched on, only positions 2 and 5 are labeled, e.g. "10^2 2 5 10^3".

The axes divisions and subdivisions, for which you have enabled the option Account for this axis when drawing the grid on the Appearance tab of the Properties dialog box, also determine the location of the grid lines, if a grid is to be displayed.

You can label the divisions and subdivisions of an axis with any values taken from a data set or calculated using a function. For labeling you can use a data series, a signal or a formula with an argument (function). FlexPro then labels the divisions and subdivisions as follows:

Data structure	Labeling	Typical use
Data Series	The values are taken from the data series one by one and written onto the axis. All data types, including strings, are permitted. The values have to be stored in the same order in the data set in which the labeling of the axis occurs. The division labeling then proceeds from the division origin to the end value and then to the starting value of the axis. This is why you should set the division origin to the starting value of the axis.	Labeling of the divisions with text originating from a data set.
Signal	The value resulting from the axis division is searched for in the X component of the signal and the corresponding Y value is then written onto the axis. If necessary, interpolation is performed.	Display signals against any measured quantity, which does not have to be monotone, e.g., time signals over a space coordinate.
Function	The value resulting from the axis division is passed to the function as an argument and the result of the function is written onto the axis. All data types are allowed for the result, including strings.	Linearization of any characteristic line.

The axis will continue to be scaled according to the scaling configured and, if applicable, the values established through automatic scaling.

Axis Labeling

You can add an axis label to each axis of a diagram. You can specify the axis label text directly or use the text from the curves displayed on the axis, which also have an Axis Labeling tab in their Properties dialog box. You can enter any text in which you can embed Fields 511 for accessing header information of the displayed data sets. FlexPro offers a variety of pre-defined fields for labeling axes so that you do not have to program these yourself.

If you display multiple curves and use their axis labels, then these form text stacked vertically written on the axis. However, you can turn this option off for individual curves to show the time information only once on the X axis, for instance. The axis labels of the individual curves can be colored in with the respective color of the curve to make label assignment easy.

The data that you display above an axis should all have the same physical unit. You can add this unit to the axis label, display it between the last two divisions or behind each division.

Position

The axes of 2D diagrams can be displayed at either edge of the diagram, at both edges or in the axis origin of an assigned axis pointing in the other direction. Several Y axes in a 2D diagram can be displayed as stacked instead of side-by-side. Here, the diagram is divided into several areas, and you can adjust the size of these areas. It is also possible to limit the area across which an axis is stretched. You can, for instance, create a second Y axis that only stretches over an area of 0 % to 10 % of the height of the diagram to overlay a digital channel in a narrow band as a step line. You can also display several X axes side by side 410.

Curve

A diagram can have any number of curves. Each curve refers to a data set to be displayed and describes how its data is to be presented visually.

A curve in a 2D diagram can show all data structures except space curves. Each curve is assigned to exactly one X axis and one Y axis. If you specify a data series, its data is used for the Y coordinates of the points of the curve and the X coordinates are generated automatically by numbering, beginning with zero. When the curve is drawn, the positions of the points in the diagram plane are determined by the axis scaling. Parts of the curve that lie outside the area of the diagram are clipped. Scalar values can be displayed as horizontal or vertical lines. Some display formats use additional data, e.g., a data set with error values to display error indicators.

A 3D diagram curve can show data matrix, signal series and space curve data structures. Analogous to the 2D diagram, X and/or Z components that are not present are also generated automatically by numbering, beginning with zero. Instead of a signal series or a data matrix, it is also possible to specify several signals or data series that are then managed as a list in the curve and together form a signal series and/or a data matrix. Unlike 2D diagrams, there is only one axis system in 3D diagrams, so all curves are applied to the same axes and therefore have to have the same unit. For 3D curve types that require gridded data, a space curve (scatter data) is automatically converted into a signal series before display. You can choose one of two algorithms on the Data tab to display the surface.

On the Data tab of the curve, not only can you enter a reference to a data set or a formula, but you can also enter any FPScript expression. For instance, use an index operation to present a signal segment or a particular signal from a signal series. The FlexPro Diagram Wizard uses FPScript expressions within curves, e.g., for displaying percentages or decibels.

Both diagram types support display of lists that contain several data sets. Only place one curve in the diagram and specify the list on the Data tab. Several curves are shown, one per list element. You can also refer to the Dynamic Axes and Curves option in the Diagram Wizard. When displaying lists in 3D diagrams, the units of all of the list's data sets must be compatible with each other.

2D Curve Display Formats

The 2D diagram curves let you use the following display formats, which you can combine in any way you like:

- Connection Line

Use this display format to display signals. You can choose from several types of connection lines:

Type	Function
Straight line	connects the points of the curve with a straight line.
Step line	connects neighboring points with horizontal and vertical lines instead of diagonal lines.
Segment	draws a line between every other point of the curve.
Triple segment	connects two points of the curve with a line and then leaves a gap.
Spline	connects the points of the curve with a spline curve. You can specify the number of sampling points to be calculated.
Compensating spline	connects the points using a compensating spline curve. The spline curve does not necessarily run through the points. You can control this using a weighting factor.

You can specify the line style, color and arrow type. Another option is to use a color palette. You then have a multi-colored curve for displaying tolerance bands, for example.

If the data set contains void values, you can specify whether these are to be bridged with a line or whether a gap should be shown in the curve.

- Symbols

This display format draws a small symbol, e.g. a rhombus or a circle, at each point of the curve. You can freely set symbol shape, size and color. Alternatively, the size of the symbols can be determined using an additional data set. Particularly for the frequent occurrence of the "Symbols" plus "Connection Line" combination, you can specify that a small gap must be maintained between the symbol and the connection line. The connection line then stops just before the symbol.

The "Horizontal Line" and "Vertical Line" symbols are especially intended for adding range limits into the diagram. You can specify a scalar value that then determines the location of the line.

The "Vector" symbol is available for displaying vectoral measures. Here you can pre-define the angle and length of vectors or specify additional data sets.

- Error Indicators

The Error Indicators display format draws an error indicator at every point of the curve. The indicators can be drawn in Y or X direction and only in one or in both directions. The length of the error indicator from the point of the curve to the limiting line can be specified as an absolute value, as a distance to the point of the curve, or as a percentage value in relation to the point of the curve. To do this, you can enter fixed values or additional data sets.

- Data Labeling

  The "Data Labeling" display format outputs a text with the value and unit of X and/or Y coordinate for the point at the location of each point of the curve. Alternatively, you can output values from an additional data set. The data set may contain numerical data or strings. You can freely set the writing direction and placement relative to the point.

- Columns/Bars

  The "Columns" display format draws a column at each X value of the curve. This column runs from the X axis to the Y value. The "Bars" display format draws a bar at each Y value of the curve. This bar runs from the Y axis to the X value. These display formats are also available in stacked and differential variations as well as in a variant in which the base point is not the axis, but rather the axis origin. For stacked columns, the columns for several curves are displayed one on top of another instead of grouped side by side. Differential columns do not start at the X axis, but instead start with Y values taken from another data set. Analog applies to stacked bars and differential bars, but in the X direction in this case. You can set column or bar width, fill color and style.

  With bar charts and column charts, you can either specify a fixed width as a percentage of the spacing between neighboring columns, or select the Dynamic column width option on the curve's Columns/Bars tab. In this case, the columns are drawn in a way that allows them to be seamlessly tagged onto each other. This is the case even if the X intervals between neighboring columns vary. The limit of neighboring columns always lies in the middle between two neighboring points of the curve.

- Fill

This display format fills the surface between the curve and the X or Y axis with a particular color, a hatching or a linear color gradient shading. A second data set can also be specified for differential filling.

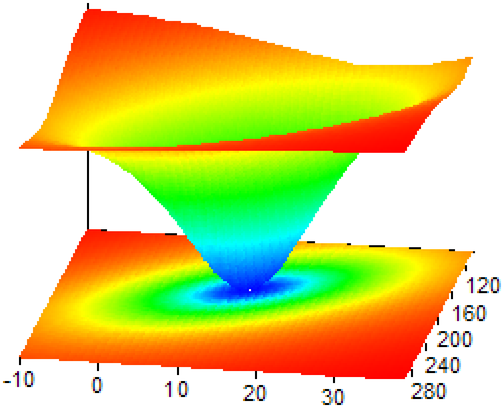
Display formats for a single curve can be combined in any way you want; for instance, a connection line can be combined with symbols or error indicators. Alternatively, you can also set up several curves with different display formats; for instance, the first curve would have a connection line to represent a signal, and a second curve would have symbols to represent a data set that contains only local maxima in the signal as points.

3D Curve Types and Display Formats

As opposed to a 2D diagram, the following curve types differ for a 3D diagram:

- Surface

This type of curve displays the data as a surface defined by data points arranged in a grid. Space curves (scatter data or XYZ data) are automatically converted to a signal series (data grid) before display. The conversion takes place either through resorting or through "natural neighbor" interpolation. The surface can be smoothed, if necessary. For the "resort" algorithm, smoothing takes place later using bicubic splines. You can adjust all of the curve settings on the Data tab. You obtain a contour diagram by setting the diagram's angles of rotation and inclination so that the result is a plan view.

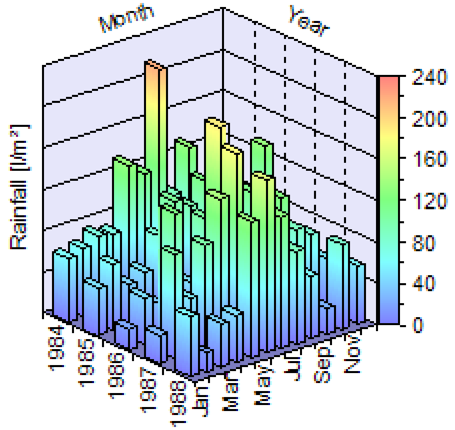


- 3D Columns

  You can display signal series (with a Z component, if applicable) or data matrices in a 3D column chart. A column chart can also be a combination of several data series or signals with identical X components.

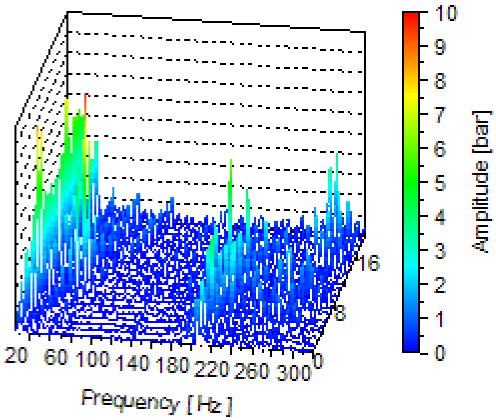
  You can specify a fixed width of 1 to 100 percent of the spacing between neighboring columns for the column width, defining the minimum distance in the X and Z direction to 100 percent. The spike chart means that columns are drawn as "needles", using the attributes from the Fill tab as the base. The border lines are omitted in this case.

  When selecting Color field matrix in the Diagram Wizard, the result is a 3D column in plan view with a color shading or color palette in the Y direction.



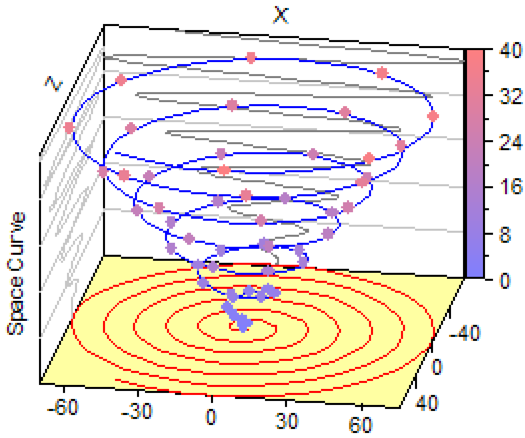
- Waterfall

  In a waterfall chart you can display signal series (possibly with a Z component and two-dimensional X component), data matrices or space curves. A waterfall diagram can also be put together from several data series or signals. Space curves are converted automatically to a signal series before being displayed by resorting using the CurveToSurface function. You can change the tolerance for the conversion on the Data tab of the curve in the diagram.



- Space Curve

In a space curve chart, you can display data sets of the space curve data structure or three data series, which combine to form a space curve.



Usually, a 3D diagram only uses one curve from one of the four types. In some cases, however, it makes sense to use several curves, for instance, to label individual points in a contour chart.

Display Formats

The following display formats can be used within a single type of curve:

- Border Lines (3D Columns)

  To reinforce the impression of space, the borders of the three-dimensional columns can be highlighted with lines. These border lines are particularly important when no lighting source is used. Typical line attributes can also be applied to border lines.

- Grid Lines (Surface)

  You can use grid lines to display the surface as a grid. Neighboring points are joined up by lines whose style and color you can set. It is possible to use a color palette in the Y direction. In addition to this, you can project the grid onto the coordinate system planes.

  If you do not use any filling for the surface, you can also make the grid lines visible, although these are not visible from the viewing angle set.

If the data set contains void values, then all of the connection lines to these values are omitted, causing gaps in the grid network.

- Connection Line (Space Curve, Waterfall)

  This display format connects a curve's adjacent points with a line for which you can adjust the style and color. For the waterfall chart, you can use a color palette in the Y or Z direction or linear or spectral color gradient in the Y direction. The space curve can be colored using a color palette in any direction.

  If the data set contains invalid values, you can specify whether these are to be bridged with a line or whether a gap in the curve is to be shown.

  The connection line for the waterfall chart can also be drawn as a step line, especially for displaying digital signals. You can freely choose the location of the vertical edge between neighboring points.

- Symbols (Surface, Space Curve, Waterfall)

  This display format draws a small symbol, e.g. a rhombus or a circle, at each point of the curve. You can freely set symbol shape, size and color.

- Data Labeling (Surface, Space Curve, Waterfall)

  The "Data Labeling" display format shows text with the value of the X, Y or Z coordinate of the point for each point on the curve. You can also use data from additional data sets for data labeling.

- Fill (Surface, Waterfall, 3D Columns)

  This display format fills the surface, the area under each curve of a waterfall chart, or 3D column walls with a solid color, linear color gradient or a color spectrum. Alternatively, you can specify a palette with fill colors that can be used for the surface in the Y direction, for the waterfall chart in the Z direction, and for the columns in any direction. If desired, you can also project the filling of a surface onto the planes of the coordinates system.

  If you use a color palette in the Y direction or one of the color shadings, you can switch on a color legend and thus visualize the color change regardless of the viewing angle of the 3D diagram. The coloring also depends on the lighting settings on the 3D diagram's 3D View.

- Contour Lines (Surface)

  These are intersection lines between the surface and planes, which are parallel to one of the three planes of the axis system. Contour lines emphasize the contour of a surface and can be drawn for any direction. The intersection planes can be specified by the divisions of the corresponding axis or by a freely selectable data set. The contour lines can also be optionally projected onto the corresponding plane of the coordinates system. You can also have a label with the value assigned to the contour line.

- Projections (Space Curve)

  This display format projects the space curve onto the planes of the coordinates system. You can freely set the line attributes of the projection.

- Perpendiculars (Space Curve)

  This display format allows you to draw perpendicular lines from the points of the curve onto the planes of the coordinates system. You can freely set the line attributes for the perpendicular lines.

Legend

A diagram can have a legend that provides clear assignment of curves to the data on which the curves are based.

For each diagram curve selected to have a legend, a short section of the curve is drawn in the legend and labeled with text. For each curve, you can enter the text and insert fields that are replaced by header information from the curve's data set.

You can specify a title for the legend. You can color in the background, and a frame can be added to the legend. You can specify how many columns the legend is to have and in which order the curves are to be displayed. You can freely position the legend. It is, however, assigned to a corner of the diagram, so that when the diagram size is changed, the legend is shifted accordingly.

Axis and Curve Labeling

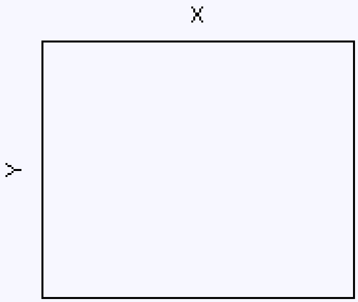
In addition to the legend, you can also use axis and curve labels to display particular data on the curve. The curve label is written directly on the course of the curve. You can set its position. You can specify the axis label for individual curves separately for each axis of the diagram. The lines of text are then written on the axis, with each line placed below the last line. You can add fields to the text to display attributes such as the name or unit of the data set upon which the curve is based.

3D View and Lighting

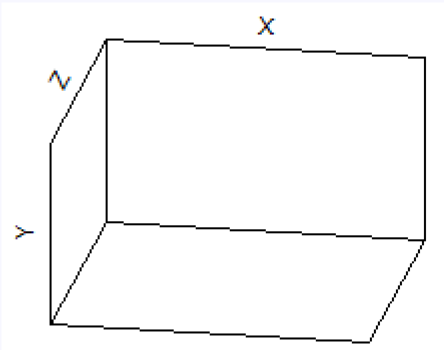
The 3D view is determined by specifying an angle of inclination and an angle of rotation. Here, the angle of rotation specifies by how many degrees the coordinates system should be rotated clockwise around the Y axis. The angle of inclination determines by how many degrees the coordinates system should be tilted counter-clockwise, relative to the projection plane.

An angle of rotation and an angle of inclination of 0 specify that the viewing angle is located exactly opposite the XY plane of the coordinates system.

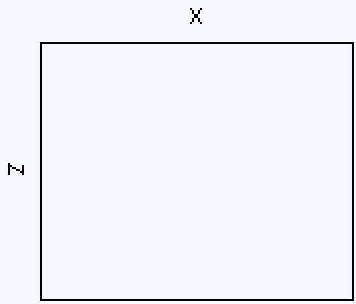
The following diagrams are a few examples:



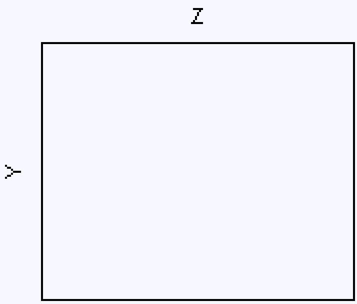
Angle of rotation = Angle of inclination = 0°
(opposite the XY plane, Z points to the viewer)



Angle of rotation = 10°, Angle of inclination = 30°
(Standard)



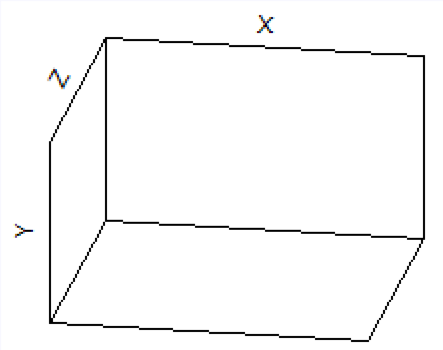
Angle of rotation = 0°, Angle of inclination = 90°



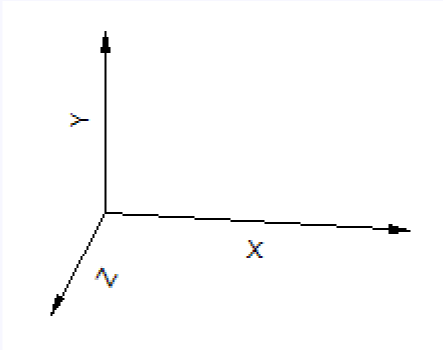
Angle of rotation = 90°, Angle of inclination = 0°

(Plan view, Y points to the viewer)	(opposite the YZ plane, X points to the viewer)

The axis system of 3D diagrams can be presented as an axis intercept with planes or as an axis intercept only:



Axes and planes



Only axes

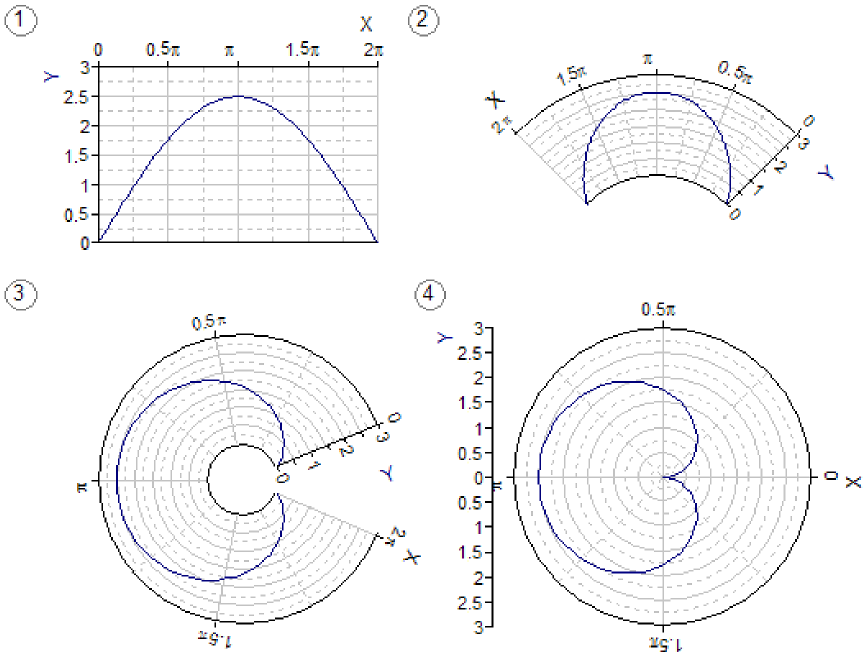
While the planes are always displayed behind the curve, the axis intercept is always displayed in the coordinates origin, even if it ends up in front of the curve.

For curves types with surfaces (surface, columns), you can also add a lighting source. The position of the lighting source is determined similarly to the viewing angle, but here, all of the angle specifications are relative to the viewing angle.

Polar Graph

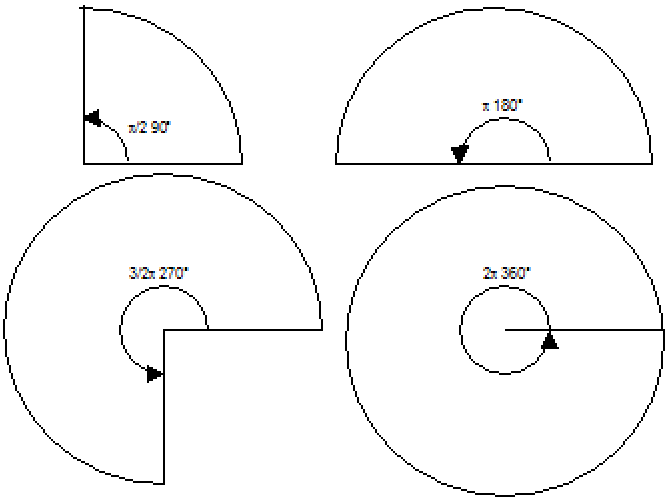
All 2D diagram display formats are also available for the polar graph. Here, the complete diagram is transformed into a full circle, a segment of a circle or a ring by using polar transformation. The X axis of the 2D diagram is transformed into an angle axis, and the Y axis is transformed into a radial axis.

The following illustration shows the transformation from Cartesian graph to polar graph:

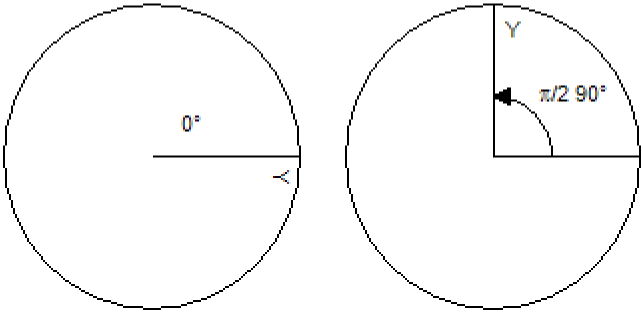


Note that the orientation of the X axis is reversed by the transformation.

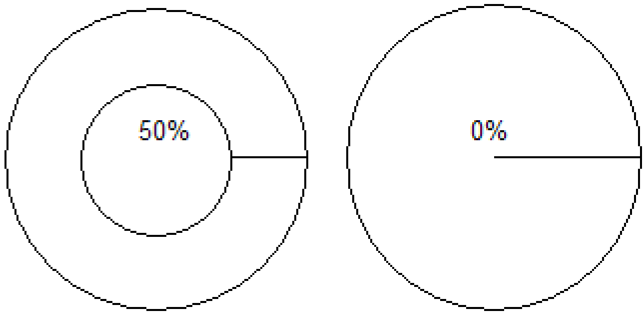
On the Polar Transformation tab of the 2D diagram's Properties dialog box, you can activate polar transformation and set parameters for it. The transformation is controlled by the opening angle, angle of rotation and inner radius attributes. The opening angle determines the size of the circle segment upon which the diagram is formed. An opening angle of 2π or 360° in this case is the equivalent of a complete circle.



The angle of rotation determines the location of the diagram's original left border. The polar diagram can be rotated counterclockwise.



The inner radius is specified in percent and determines whether the diagram is to be in the shape of a circle or a ring.



The X values of the data to be plotted can include the angle in radians (0 - 2π) or in degrees (0 - 360°). The diagram's angle axis is automatically labeled with the respective angles if this is set for metric axis division. It is also possible to label the angle axis in degrees, even though the measured values are in radians, and vice versa. Here, only the respective angle formatter for the axis must be specified.

A polar graph with an opening angle of 2π or 360° and metrically scaled X axis with a start value of 0 is considered closed, i.e. no longer has a start or end value for the angle axis. You can thus also plot data that stretches across a larger angular range than [0, 2π] or [0, 360°], i.e. over several "rotations". Also, with regard to positioning the Y axis or radial axis, the closed polar diagram deviates from the open polar diagram. While the Y axis, depending on the selected position, is plotted for an open diagram at one or both sides or within the circle segment, this is only displayed for the closed diagram within the circle if it was positioned at the axis origin of the X axis. Otherwise, the axis is positioned next to the full circle.

The diagram is usually an exact circle and fits into the center of the diagram's rectangular area. However, if you enable the option Allow elliptical distortion, then the diagram is distorted elliptically so that the diagram's plane is sized for best fit.

Color Palette and Color Gradient

Using a color palette or color gradient can considerably improve legibility, especially of 3D graphics. In 2D diagrams, color palettes are often used for presenting ranges of values, such as tolerance bands and warning ranges. Color gradients, especially spectral color gradients, which simulate the colors of a rainbow, form the basis for what is called a 2 1/2D graphics contour and color field matrix. Here, the information for the third dimension is marked by the color.

The assignment of the colors for color gradient or for the color palette to the scaling the Y axis in a 3D diagram is represented by its color legend.

Grid

You can provide the surface of the 2D diagram or the coordinates planes of the 3D diagram with a grid to increase the legibility of the diagram. The location of the grid lines is oriented toward the divisions and subdivisions of the axes spanning the plane. Since 2D diagrams can have any number of X and Y axes, each axis can be set separately, whether or not the grid lines are to be drawn for their divisions.

You can separately style the horizontal and vertical lines as well as the frame and the grid background. The frame and background are part of the grid in 2D diagrams only, since here the axes do not have to be positioned on the edges. In 3D diagrams, the framing of the planes as well as their background color are part of the axis system.

Color Legend

In 3D diagrams, the color legend creates a relationship between the colors of the color shading used in a curve and/or a color palette and the scaling of the diagram's Y axis. This is another Y axis with a band of color, which is positioned next to the actual diagram.

A color legend uses the scaling of the diagram's Y axis. To do this, all settings must be applied at the Y axis. Otherwise, you can freely format the color legend as you would an axis.

Working with Diagrams

Creating a Diagram

1. In the Folders window, select the folder where the new diagram is to be placed.
2. Select the data you want to display as curves in a diagram. You can select data sets, formulas, analysis objects or a cell range in a folder's data view or in the window of an open data set. You can also use the range between the cursors of a diagram as data selection.

Note If you want to display measurement data located in a subfolder, first activate it to show the measurement data sets.

---

3. If you want to embed the new diagram into a worksheet or document, at this time click on the open worksheet or document window.
4. Choose an appropriate diagram from one of the galleries under Insert[Diagram] or click Insert[Diagram] -> Diagram Wizard if you want to create a special view, such as an XY diagram or a diagram with logarithmic axes. You can obtain additional information using Help in the Diagram Wizard.

Note You can easily create an embedded line diagram for a series of data sets by selecting the data sets in the object list and using drag-and-drop to drag them into a document or worksheet.

Changing the Diagram Style, Colors and Background

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. From Design[Style], select Diagram Style, Colors and Background.

Notes

- All diagram attributes that are controlled by the diagram style, color and background, such as the color of curves, are automatically reset. If you do not want this to happen, change the settings for Diagram Style, Colors and Background on the Style tab in the Properties dialog box of the diagram.
- The first palette in the Colors list box can be customized. You can change these palette colors as well as additional attributes assigned to the selected diagram style by opening the Design[Style] > Diagram Style list box and clicking Edit Style Sheet.

Resizing the Diagram Borders and Plane

To resize the borders using your mouse:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on a point on the plane of the diagram where there are no grid lines or curves.
3. Handles appear at the four borders of the diagram's plane. You can use these handles to resize the diagram's frame by dragging the borders with the mouse.

You can resize the entire diagram by dragging the hatched frame surrounding it, provided the Fit To Window option in the View menu is switched off.

To resize the borders and diagram plane, do the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a part of the diagram's plane where there are no grid lines or curves.
3. Click on the Size tab.
4. Enter the desired measurements.

Adding or Deleting a Diagram Title

To add a diagram title, do the following:

1. Click Design[Diagram Layout] > Add Diagram Element and select Diagram Title.
2. Double-click on the new title.
3. Enter text for the title or from the Fields menu select an appropriate field, such as Comments.

Note If the title is not completely displayed, you may have to increase the diagram margins or enable the Correct margins automatically option on the Size tab.

To delete a diagram title, do the following:

1. Double-click on the diagram title.
2. In the dialog box that appears, deselect the Diagram title on option.

Aligning the Diagram Title

1. Click on the diagram title to select it.
2. Click on the desired alignment under Format[Align].

Enabling and Disabling Automatic Margin Correction

FlexPro can automatically resize the borders of 2D and 3D diagrams if the diagram's elements extend beyond the edge of the window. You can enable or disable automatic margin correction by doing the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on a point on the diagram's surface where there is no curve.
3. Click on the Size tab.
4. Check the option Correct margins automatically or remove the checkmark.

Changing the Background Color of the Diagram's Plane

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

To select a design color for the current color scheme:

2. Select one of the design colors from the list box under Diagram Design[Style] > Background.

To choose a background color:

2. Select the diagram background by clicking on a position where there is no curve.
3. Choose a color from the list box under Format[Fill] > Color.

Note If you choose No Fill for the color, the design color selected under Diagram Design[Style] > Background will be used.

Toggling Grid Elements On and Off

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

To display a horizontal or vertical grid line:

2. Click Diagram Design[Diagram Layout] > Add Diagram Element and select Horizontal Grid Lines or Vertical Grid Lines.

To toggle individual grid elements on or off:

2. Double-click on the diagram plane to open the Properties dialog box 107.
3. Click on the Grid & Background tab.

3D Diagram:

4. Select the tab for the coordinates plane for which you want to display a grid.
5. Select the grid elements to be displayed.

Styling Grid Elements

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the grid elements for which you would like to change the style.
3. Select the desired line width in the list box under Format[Lines] > Width, the desired line style in the list box under Format[Lines] > Dashes and the color in the list box under Format[Lines] > Color.

Positioning the Color Legend of a 3D Diagram

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Use your mouse to select and drag the color legend to the desired position.
3. To specify which corner of the diagram you want to assign the legend, double-click on the surface of the color legend.
4. In the Color Legend Properties dialog box, select the Position tab.
5. Select the desired position in the field To corner.

Splitting a 2D Diagram into Multiple Pages

1. If the diagram is in a document, double-click on the diagram to open it.
2. Use your right mouse button to click on the diagram and select Properties from the context menu.
3. Click on the Page Splitting tab.
4. Select the desired Mode for page splitting. Here, call up Help in the dialog box if necessary.

5. In the field that now appears, specify the desired Number of pages or Number of divisions per page.

Navigating Multi-Page 2D Diagrams

- Use the commands Diagram Design[Page] > Previous or Diagram Design[Page] > Next to go to the previous or next page of the diagram.
- To show the first or last page of the diagram, use the commands Diagram Design[Page] > Previous > First or Diagram Design[Page] > Next > Last.
- In the input field Diagram Design[Page] > Go To enter any page number to view it.
- You can use the mouse wheel to scroll through several pages of the diagram.

Changing the 3D Diagram Viewing Angle

To change the viewing angle, do the following:

1. Click 3D Diagram Design[3D View] > Rotate Right to rotate the diagram clockwise by 10°. You can rotate the diagram by 10° in the opposite direction by clicking 3D Diagram Design[3D View] > Rotate Left.
2. Click 3D Diagram Design[3D View] > Higher Inclination to increase the angle of inclination by 10°, or 3D Diagram Design[3D View] > Lower Inclination to decrease it by 10°.
3. Open the menu under 3D Diagram Design[3D View] > Standard View and select Plan View to view the diagram from above. The Y axis is now pointing toward you and is no longer visible. The same applies to the Z axis if you click on Front View and for the X axis if you click on Side View.
4. By clicking on the icon 3D Diagram Design[3D View] > Standard View you can reset the view to the default setting.

If you want to set the angles more precisely, do the following:

1. Click on Diagram Design [Object] > Properties.
2. In the dialog box that appears, select the 3D View tab.
3. In the Perspective box, select one of the viewing angles provided or select Free Perspective and enter the angle.

Changing the 3D Diagram Lighting Parameters

To change the lighting parameters for a column or surface diagram, do the following:

1. Click on Diagram Design [Object] > Properties.
2. In the dialog box that appears, select the 3D View tab.
3. Turn on the lighting, select the desired lighting position and set the contrast.

Embedding Pictures

1. Click Diagram Design[Illustrations] > Pictures.
2. Select a file in the Insert Image dialog box.
3. Click OK to close the dialog box. The newly inserted image is now displayed.
4. If necessary, drag the image with your mouse to position it.
5. Set the absolute height and width on Format[Measures]. You can change the relative size in relation to the original image size in the following Properties window fields: Scaling Width and Scaling Height.

Working with Axes

Adding or Removing an Axis

To remove an axis:

1. Select the axis that you would like to delete.
2. Press the DEL key.

Note: You cannot delete the last Y axis or X axis.

To add an axis:

1. Select an existing axis of the same type (X or Y axis).
2. Use your mouse to drag a copy of the axis to the desired position while holding down the CTRL key.

Specifying the Output Unit

1. Double-click on the axis.
2. In the Properties dialog box 107 of the axis, click on the Scaling tab.
3. Under Unit(s) enter the desired output unit or choose the unit from the list.
4. Close the dialog box with OK.

Note: If you specify an output unit, the unit of the data of all curves displayed across this axis must be compatible.

Changing the Axis Origin

1. Double-click on the axis.
2. In the Properties dialog box 107 of the axis, click on the Scaling tab.
3. Enter the axis origin in the field of the same name.

Note: Only the Y axis in 3D diagrams has an axis origin.

Arranging Axes in a 2D Diagram

To change the order of axes at the same location (edge, axis origin):

Using the mouse:

- Drag the axis to a different axis whose position it is to take, or drag it to the outside edge of the diagram in order to place it there.

Using the diagram's Properties dialog box:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on Diagram Design [Object] > Properties.
3. In the dialog box that appears, click on the Axes tab.
4. In the Axes list, select an X or Y axis.
5. Use the Up or Down buttons to move the axis.

Using the Properties window:

1. Select the axis that you want to move.
2. Enter a different value in the Plot order field of the Properties window. If you increase the value shown, an X axis will appear further down and a Y axis will appear further to the left or, in diagrams with stacked Y axes, further down.

To display axes at a different location:

Using the mouse:

- Drag the axis to the desired edge of the diagram or to the diagram area in order to place it at that edge or in the axis origin.

Using the diagram's Properties dialog box:

1. Double-click on the axis whose position you want to change.
2. In the Properties dialog box 107 of the axis, click on the Position tab.
3. Now in the Position field, choose whether the axis should appear on the left or right edge of the diagram or in the axis origin of an axis for the other direction. If it is a different axis at the same location, you can also specify the Spacing to the previous axis. If you select In axis origin, then in the Assigned axis field specify the axis origin of the axis to be used.

Using the Properties window:

1. Select the axis whose position you want to change.
2. Select the desired position in the Position field of the Properties window.

To set the spacing of adjacent axes:

Using the mouse:

1. Use your mouse to click on the line of the axis that you want to move.
2. Drag the axis to the desired position.

Using the diagram's Properties dialog box:

1. Double-click on the axis that you want to move.
2. In the Properties dialog box 107 of the axis, click on the Position tab.
3. Enter the desired value in the field Spacing to the previous axis.

Using the Properties window:

3. Select the axis that you want to move.

4. Enter the desired value in the Spacing field of the Properties window.

Note If on the Axes tab of the 2D diagram's Properties dialog box you selected the option Automatically enlarge axis spacing when axes overlap, FlexPro automatically selects a minimum spacing to ensure that no overlapping occurs when displayed.

See Also

Axis 382

Stacking the Y Axes of a 2D Diagram

To stack the Y axes of a 2D diagram, do the following:

1. Double-click on the area of the diagram, but not on the curve.
2. In the diagram Properties dialog box 107, click on the Axes tab.
3. When arranging axes, select Y axes stacked or Y axes side-by-side, grouped by unit.

Adjusting the Height of Stacked Y Axes

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

Using the mouse:

2. Drag the dividing line between the two diagram planes to the desired position.

Using the Properties window:

2. Select the Y axis whose height you want to adjust.
3. In the Size percentage field of the Properties window, specify the desired height of the axis relative to the overall height of the diagram.

Displaying the X Axes of a 2D Diagram Side By Side

FlexPro cannot stack X axes of a diagram like it can Y axes, but you can obtain the same effect by doing the following:

1. If the diagram does not have two X axes, first add an axis 407.

2. Double-click on the diagram's first X axis.
3. In the Properties dialog box 107 of the axis, click on the Position tab.
4. In the Start of axis field, enter 0% and in the End of axis field, enter 50%.
5. Close the dialog box with OK.
6. Double-click on the second axis and switch again to the Position tab.
7. In the Position field, select the same position that you used for the first axis.
8. In the Spacing to the previous axis field, enter 0.
9. In the Start of axis field, enter 50% and in the End of axis field, enter 100%.
10. Switch to the Appearance tab and make sure that the option Account for this axis when drawing the grid is checked.

Adding or Deleting an Axis Division

To delete an axis division:

1. Double-click on the axis division.
2. On the Appearance tab of the Properties dialog box 107 of the axis or color legend, uncheck the Divisions and/or Subdivisions option.

Note You can also delete an axis division by selecting from the Format[Lines] > Color menu the entry No Line.

To add an axis division:

1. Double-click on the axis to which you would like to add a division.
2. In the axis Properties dialog box, click on the Appearance tab.
3. Select Divisions and/or Subdivisions.

Changing Division Length and Type

1. Select the division that you would like to edit. You can simultaneously select the divisions for several axes and the color legend by holding down the CTRL key. Select all divisions of the diagram by holding down the ALT key.
2. You change the length of the division dashes by specifying a different value in the field under Format[Measures] > Width.

3. You can change the division type by clicking on the corresponding icon under Format[Alignment]. For instance, click on to set divisions, depending on the axis orientation, to Left (Y Axis), Bottom (X axis) or Outside (3D Axis).

Toggling Isometric Axis Adjustment On and Off

1. Click on Diagram Design [Object] > Properties.

2D Diagram:

2. In the diagram Properties dialog box 107, click on the Axes tab.
3. Check the option Adjust primary system of axes isometrically or remove the checkmark.

3D Diagram:

2. In the diagram Properties dialog box 107, click on the System of Axes tab.
3. Isometric Adjustment.

Adding or Deleting a Division Label

To delete a division label, do the following:

1. Double-click on the division label.
2. On the Division Labeling tab of the Properties dialog box 107 of the axis or color legend, uncheck the Division labeling on checkbox. If you would only like to switch off the subdivision labels, remove the checkmark from Also label the subdivisions.

To add a division label, do the following:

1. Double-click on the axis or color legend to which you would like to add a division label.
2. In the Properties dialog box, click on the Division Labeling tab.
3. Check the Division Labeling checkbox and, if applicable, Also label the subdivisions.

Changing the Division Label Format

To change the formatting of all values of an axis division label, do the following:

1. Double-click on any division label with a format that you would like to change.
2. Click on the button on the right side of the Format input field to adjust the formatter.
3. Select the desired formatting.

Using the Properties window, you can set formatting for several axes at once:

1. If you do not already see the Properties window, select View[Task Windows] > Show > Properties.
2. Select the division labels for several axes whose formatting you would like to change by clicking on these while holding down the CTRL key. Select all diagram division labels by holding down the ALT key.
3. Now click in the Format input field of the Properties window and then click on the button that appears to the right of the field.
4. Select the desired formatting.

Note: The formatting applies to all of the division labels for the selected axes.

Aligning the Division Label

1. Click on the division label to select it.
2. Click on Format[Alignment] and select the appropriate symbol to choose the alignment. For instance, click on to position the label at the top left of the division marker.

Labeling the Axis Divisions with Text

1. Double-click on the axis with the division that you would like to label.
2. In the Properties dialog box, click on the Division Labeling tab.
3. Check the option Division labeling using a data series, signal or function.
4. Specify a data series with strings that contains the desired text, or specify directly an FPScript expression that generates the data series, such as `{"One",`

`"Two", "Three"}`. The values must be specified in the same order as the axis labeling. The division labeling then occurs from the division origin to the end value and then to the starting value of the axis.

Displaying the Unit in the Axis Division

1. Double-click on the axis division label in which you want to add the unit.
2. On the Division Labeling tab, select the option Add unit to division labeling.
3. In the field under this option, choose whether you want to display the unit at every division or between the last and second to last division.

Adding or Deleting an Orientation Arrow

To delete an orientation arrow, do the following:

1. Double-click on the orientation arrow.
2. On the Appearance tab of the axis or color legend Properties dialog box 107, uncheck the Orientation arrow checkbox.

To add an orientation arrow, do the following:

1. Double-click on the axis or color legend to which you would like to add an orientation arrow.
2. In the Properties dialog box, click on the Appearance tab.
3. Check the Orientation arrow box and specify the desired Position.

Positioning an Orientation Arrow

1. Double-click on the orientation arrow.
2. On the Appearance tab of the axis or color legend Properties dialog box 107, select the desired position under Orientation arrow.
3. Under Length, enter the desired length of the arrow.
4. Under Spacing, specify, if applicable, the distance from the division label or from the axis label.

Positioning the orientation arrow on the axis label of 3D diagrams is only an option when Parallel to the axis is set as the Angle on the Axis Labeling tab.

Displaying an Axis with Customized Scaling

FlexPro can label the divisions of an axis with values that can be calculated using a function. This allows you to develop your own scaling. The following example shows how you can display a square characteristic curve as linearized, for instance. The characteristic curve is linearized using its inverse function, in this example the square root function, and displayed in a linearly scaled diagram. The Y axis of the diagram is labeled using the square function.

Do the following:

1. Create an FPScript function called Linearized that allows you to linearize the data set with the characteristic curve. The formula must contain the following:

   ```
   Arguments x
   Sqrt(x)
   ```

2. Use the mouse to drag the CharacteristicCurve data set onto this formula in order to obtain the linearized characteristic curve as new formula called CharacteristicCurveLinearized.
3. Create an FPScript function called Parabola used to square the division labels of the Y axis:

   ```
   Arguments x
   x^2
   ```

4. Select the formula CharacteristicCurveLinearized and call up the Diagram Wizard to create a linearly scaled 2D diagram. The characteristic curve is to be displayed as a straight line. The divisions of the Y axis, however, still correspond to the linearized values.
5. Double-click on the division label of the Y axis to open the Division Labeling tab.
6. Check the option Division labeling using a data series, signal or function.
7. Check the option Also for data labels, markers and in the coordinates window to make sure that the transformation will also be used there.
8. Enter the function Parabola into the field for the data set.
9. After you have exited the dialog box by clicking on OK, the squared values will be displayed on the Y axis.

Changing the Axis Scaling

1. Double-click on the axis.
2. In the Properties dialog box 107 of the axis, click on the Scaling tab.
3. Select the desired scaling options.

Labeling Only the Axis End Values

1. Double-click on the axis.
2. In the Properties dialog box 107 of the axis, click on the Scaling tab.
3. As the Axis type, select Linear, number of divisions.
4. Select the options Extend start value to previous full division or by 10 % and Extend end value to next full division or by 10 %.
5. In the Division origin list box, select equal to start value.
6. As the Number of division intervals enter 1.

Note: If you want the axis to be labeled with the exact minimum and maximum of the curve, under Scaling you will need to select Data minimum/maximum as the Start value/End value and enter the value 0 in the Significant digits field.

Changing the Number of Subdivisions

1. Double-click on the axis line.
2. On the Scaling tab of the axis Properties dialog box 107, enter the number of subdivisions per division in the field with the same name.

Adding or Deleting an Axis Label

To add or delete an axis label to a curve, do the following:

1. Click on the curve to select it.
2. Select or deselect the option Diagram Design[Curves] > Axis Label.

To write an axis label directly on an axis, do the following:

1. Double-click on the line or division of the axis to open its properties dialog box.

2. Switch to the Axis Labeling tab.
3. If applicable, check the option Axis labeling on.
4. Uncheck the option Use axis labels of curves.
5. Now enter the desired label in the Text field.

Changing Axis Label Alignment and Orientation

1. Click on the axis label to select it.
2. Click on Format[Alignment] and select the appropriate icon to choose the position. For instance, click on to position the label at the top left of the axis.
3. Under Format[Measures] > Angle choose the writing angle at which the text is to appear.
4. If necessary, use the mouse to move the axis label in order to correct the label's position.

Displaying Data Set Information in the Axis Label

1. Double-click on the line or division of the axis to open its properties dialog box.
1. Switch to the Axis Labeling tab.
2. Make sure that Axis labeling on is selected.
3. Uncheck the option Use axis labels of curves.
4. In the Text field, place the cursor at the position where you would like to insert a field.
2. Select the desired item from the Fields list box. A field is then inserted into the text. This field is later replaced by the updated data set information.

Displaying the 3D Axis System as an Axis Intercept

1. Double-click a location outside the diagram area to open the diagram's Properties dialog box.
2. In the dialog box that appears, select the System of Axes tab.
3. Under Visualization, select Only Axes and click OK to close the dialog box.

If you want to add orientation arrows to the axis cross, perform the following additional steps:

1. Select all axes by clicking on the division label of an axis while holding down the ALT key.
2. In the Properties window, open the Orientation arrow category and set Visible to True.
3. In the Position field, select At axis.

Working with Curves

Switching On a Display Format

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve for which you would like to toggle the display formats or on the curve's axis label.
3. In the dialog box that appears, click on the tab corresponding to the display format.
4. Select the option corresponding to the display format.

Note: You can combine several display formats with each other, such as Connection line with Fill.

Arranging Curves

Using the mouse:

- Drag the curve to any other axis or, in the case of a diagram with stacked Y axes, to the diagram area spanned by the desired axis.

Using the diagram's Properties dialog box:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

2D Diagram:

2. Use your right mouse button to click on the diagram. Make sure that nothing in the diagram has been selected.
3. Click on Properties in the context menu.

4. In the dialog box that appears, click on the Curves tab.
5. Select a curve from the Curves list.
6. Move the curve using the In Foreground or In Background buttons.

3D diagram displaying several data series or signals:

2. Double-click on the curve.
3. In the curve Properties dialog box that appears, click on the Data tab.
4. Select a data set in the list.
5. Move the data set to the background using the Up button or use the Down button to move it to the foreground.

In the 2D diagram using the Properties window:

1. Select the curve that you want to send farther back or bring forward, or select its axis label.
2. Enter a different value in the Plot order field of the Properties window. Decreasing the value shown, sends the curve farther back. Increasing the value brings it forward.

Adding a Curve to a 2D Diagram

The easiest way to add a curve to a 2D diagram is to drag the data set to be displayed as the curve with the mouse from the object list into the diagram. If you use several Y axes, you have to drop the data set onto the Y axis that is to be assigned to the curve.

However, you can also insert an empty curve and then enter the data set manually:

1. If necessary, select the curve in the diagram that you want to use as a template for the new curve. If you do not select a curve, the curve selected as a template on the Curves tab of the Properties dialog box is used.
2. Open the list box under Diagram Design[Diagram Layout] > Add Diagram Element and select Legend.
3. In the dialog box that then appears, enter the data set for the curve. You can also specify separate data sets for the X and Y coordinates of the points of the curve.

Changing the Axis Assignment for a 2D Curve

Using the mouse:

- Drag the curve to any other axis or, in the case of a diagram with stacked Y axes, to the diagram area spanned by the desired axis.

Using the diagram's Properties dialog box:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

1. Double-click on the curve with the axis assignment that you would like to change, or click on its axis label.
2. In the dialog box that appears, click on the Axes tab.
3. Select the desired Y axis and/or X axis from the list boxes.

Excluding a Curve from Axis Autoscaling

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on the curve that you want included when cursoring in order to select it.
3. Uncheck the option Diagram Design[Curves] > Autoscale.

Hiding Curves

Using the Properties dialog box:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the diagram, but not on the curve.
3. In the diagram properties dialog box that appears, click on the Curves tab.
4. In the Visible column of the Curves list, remove the checkmark from all curves that you want to hide.

Note: You should use the Delete Curve button to remove curves from the list that you want to hide permanently.

Using the Properties window:

1. Select a curve by clicking on the curve itself or on its axis label. You can select several curves at the same time by holding down the CTRL key and clicking on them. Select all diagram curves by holding down the ALT key.
2. In the Properties window under Visible, select False.

Opening the Data Set of a Curve

1. Select the curve whose data set you want to open by clicking on the curve or on its axis label.
2. Click Diagram Design[Curve] > Open Data Set.

Adding a Curve Label

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or the axis label of the curve to which you want to add a curve label.
3. In the dialog box that appears, click on the Curve Labeling tab.
4. Check the option Label curve.
5. Enter the text with which you want to label the curve. You can insert fields into the text to include attributes of the data set upon which the curve is based.
6. Specify under Location the relative location of the label to the course of the curve.

Note: You can adjust the location of the cursor label later using your mouse.

Aligning a Curve Label

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the curve label that you would like to align.
3. Select a suitable alignment on Format[Align] to place the label above or below the curve.

4. Drag the text to the desired position on the curve.

Note You can move the curve label only along the curve. For 3D diagrams, it is best to choose the front view or plan view.

Adding a Data Series to a 3D Curve

To add a data series to a 3D curve that displays several data series or signals, do the following:

1. Drag a signal or a data series from the object list onto a 3D diagram.
2. A question appears, asking whether you would like to add the data set to the last curve. Answer this question by clicking on Yes.
3. To add more signals or data series to the graphic, repeat this step.

Displaying Digital Data as a Step Line

You can set the appearance for a 2D line diagram and 3D waterfall chart as a step line:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or on the axis label of the curve for which you would like to add step lines.
3. In the dialog box that appears, click on the Connection Line tab.
4. Check the option Connection line on and select Step Line from the Type field.
5. In the Location of the vertical line field, specify where the step is to lie between neighboring points.

Displaying the Hidden Grid Lines of a Surface

1. If the 3D diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surfaces for which you would like to turn the hidden lines on or off.
3. In the dialog box that appears, click on the Grid Lines tab.
4. Check the option Hidden lines.

Note: If you have surface filling turned on, the Hidden lines option will have no effect.

Smoothing a Surface

You can smooth the surface using bicubic spline interpolation.

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surface you would like to smooth.
3. In the dialog box that appears, click on the Data tab.
4. Specify the smoothing factor for Points per X and Z interval. This applies to both space directions.

Note: The display time increases quadratically with the smoothing factor.

Styling the Columns/Bars

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the columns that you would like to style.
3. Make the desired settings on Format[Lines] und Format[Fill].

Changing the Column/Bar Width

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any column or bar in the curve.

2D Diagram:

3. In the Column width field, choose either Fixed and specify the value in percentage of distance from the neighboring columns, or choose Dynamic to set the columns seamlessly side-by-side for a different X interval.

3D Diagram:

3. Switch to the Display tab and choose either Column with width of and specify the width, or choose Spike graphic.

Changing the Bottom Line of Columns/Bars

The bases of columns and bars of 2D curves can either be on the other axis or on the axis origin of the axis in whose direction they are drawn. This is the Y axis for columns and the X axis for bars.

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any column or bar in the curve.
3. Under Type, select a column or bar type with the desired bottom line.

Adding Contour Lines in One Direction

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surface for which you would like to add a contour line.
3. In the dialog box that appears, click on the Contour Lines tab.
4. Select the tab that corresponds to the direction for which you would like to add contour lines.
5. Check the option Contour lines on.
6. If you want to label the contour lines, select the option Labeling on.

Editing Contour Lines

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surface with the contour lines that you would like to edit.
3. In the dialog box that appears, click on the Contour Lines tab.
4. Select the tab that corresponds to the direction for which you would like to edit the contour lines.
5. Make the desired changes in the corresponding fields.

Adding Contour Line Labels

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the surface with the contour lines that you would like to edit.
3. In the dialog box that appears, click on the Contour Lines tab.
4. Select the tab that corresponds to the direction for which you would like to add the label.
5. Check the option Labeling on.

Adding a Color Gradient

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click the fill that you would like to set for the color gradient.

2D Columns: On the Columns/Bars tab, under Fill, select Color Gradient in the Type field.

2D Surface: On the Fill tab, select Color Gradient from the Fill style field.

3D Column, 3D Surface, 3D Waterfall: On the Fill tab, select Color Gradient or Color Spectrum from the Fill style field.

3D Waterfall: To fill the area under the curves with a color gradient, click on the Fill tab and select Color Gradient or Color Spectrum from the Fill Style field. To add a color ramp to the lines of a curve, select Color gradient or Color spectrum from the Coloring field on the Connection Line tab.

3D Space Curve: To add a color gradient to the symbols, click on the Symbols tab, select the option Fill and select Fill style Color Gradient or Color Spectrum.

Changing the Fill Direction for a 2D Curve

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the fill.
3. Set the desired fill direction in the Fill Area field.

Note: If you use the Area between curve and Y axis or Area between curve and origin of X axis fill direction, on the Data tab of the Properties dialog box 107 of the curve you should select the option Swap X and Y components. The curve is then no longer drawn from left to right, but from bottom to top instead, which matches this filling direction.

Changing a Curve's Fill Color and Style

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the fill whose style you would like to change. You can select several fills at the same time by holding down the CTRL key and clicking on them. You can select all fills by pressing the ALT key.
3. Set the color for the fill with Format[Fill] > Color. To remove the fill, select No Fill.
4. Change the fill style for a 2D diagram under Format[Fill] > Style.

Changing the Indicator Size

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on any error indicator of the curve. You can select several curves at the same time by holding down the CTRL key and clicking on them. You can select all curves by pressing the ALT key.
3. Set the size of the transverse line terminating the indicator under Format[Measures] > Width.

Changing the Indicator Type

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label.
3. In the Properties dialog box 107 of the curve you can set the indicator type in the Type drop down list on the Error Indicators tab.

Changing the Number of Indicators

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any error indicator of the curve to open its Properties dialog box 107.
3. On the Error Indicators tab, select for the Placement mode Relative and enter the desired value in the Number field. In the Start at field, enter the position of the first indicator as a percentage of the distance resulting from the total number.

Selecting the Data to be Labeled

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label.
3. Click on the Data Labeling tab of the Properties dialog box 107.
4. In the Fields list, select the items of the data points to be used for labeling.
5. If you choose a field which refers to an additional data set, specify the name of the data set in the Data Set field from which the data to be labeled should originate.
6. In the Text field, arrange the fields as desired and, if necessary, enter additional text that you want displayed in each data label.
7. Select Absolute as the Placement mode and enter the index of the first data point to be labeled in the Start at field and the number of data points to be labeled in the Spacing field. If you choose Relative as Placement mode, in the Number field, enter the desired number of labels. In the Start at field, enter the position of the first label as a percentage of the distance resulting from the total number.

Changing Data Label Formatting

To change the formatting of all values of a curve labeling, do the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any value of the labeling whose format you would like to change.

3. On the Data Labeling tab of the Properties dialog box 107 of the curve, set the insertion point in the Text text box on the field whose formatting you want to change. An example would be "%<YValue>".
4. Next, click on the Edit Formatter button.

With the FlexPro Properties window, you can set formatting for several curves simultaneously:

1. Select all curves with formatting that you would like to change by clicking on the curves while holding down the CTRL key. You can select all curves by pressing the ALT key.
2. Now in the Properties window, click in the Text text box under Label and then click on the button that appears to the right of the box.
3. In the dialog box that now appears, click on the Edit Formatter button.

Note: The formatting applies to all of the values in the curve selected.

Changing Data Label Alignment and Orientation

1. Click on the curve data label to select it. You can select several curves at the same time by holding down the CTRL key and clicking on them. You can select all curves by pressing the ALT key.
2. Under Format[Align] click on the appropriate symbol to choose the alignment. For instance, click Align Top Left to position the label at the top left corner of the curve point.
3. Choose the desired writing angle from Format[Measures] > Angle.

Creating Lines

1. If the diagram is in a document or a worksheet, double-click on the diagram to activate it.
2. Select the lines in the curve that you want to change; for instance, this might be the connection line of a curve, the grid line of a surface chart, or the border of columns. Press the CTRL key to select several items. Press the ALT key to select all items of a particular type.
3. You can set the color of the border lines by selecting a line color with Format[Lines] > Color. To remove a connection or grid line, select No Color. You

can change the line style and width with Format[Lines] > Style and Format[Lines] > Width. Change the transparency with Format[Lines] > Transparency.

Notes

A line style cannot be selected for column borders.

The border around 2D columns cannot be turned off.

The border lines of individual columns cannot be changed because only the entire curve can be selected.

Changing Display of Void Values

Points in the curve, where either the Y component or the X component is void, cannot be drawn. If you use the Connection Line display format, you can specify whether void points are to be bridged or whether they should appear as gaps in the curve.

Do the following:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or on the axis label of the curve for which you would like to adjust the settings.
3. In the dialog box that appears, click on the Connection Line tab.
4. Check or uncheck the Bridge void values option.

Turning On a Color Palette

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve for which you would like to use a color palette.

2D Line Diagram:

3. In the dialog box that appears, click on the Connection Line tab.
4. In the Style list box, select Palette in Y direction or Palette in X direction.

3D Waterfall Diagram:

3. In the dialog box that appears, click on the Connection Line tab.

4. In the Style list box, select Palette in Y direction or Palette in Z direction.

3D Surface with Grid:

3. In the dialog box that appears, click on the Grid Lines tab.
4. In the Style list box, select Palette in Y direction.

3D Columns:

3. In the dialog box that appears, click on the Fill tab.
4. In the Style list box, select Palette in X direction, Palette in Y direction or Palette in Z direction.

Connection Line for a 3D Space Curve:

3. In the dialog box that appears, click on the Connection Line tab.
4. In the Style list box, select Palette in X direction, Palette in Y direction or Palette in Z direction.

Symbols for a 3D Space Curve:

3. In the dialog box that appears, click on the Symbols tab.
4. In the Fill list box, select Palette in Y direction.

Determining the Interval Limits for a Color Palette

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label with the color palette interval limits that you would like to change.
3. In the dialog box that appears, click on the Connection Line, Grid Lines or Fill tab.
4. In the Form palette intervals with field, specify whether the palette intervals are to be predefined by the divisions of the corresponding axis or by a data set. Linking the palette to the axis division is only possible with linear or logarithmic scaling.
5. If you would like to specify a data set, enter its name in the Data Set field. You can also enter an FPScript expression that calculates a data series directly in the field.

Note: Since the two outer intervals are open intervals, the data set must contain only n-1 interval limits for n intervals.

Editing a Color Palette

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label with the color palette that you would like to edit.
3. In the dialog box that appears, click on the Connection Line or Grid Lines tab.

Adding a Palette Entry

1. Click Paste.
2. Click on Color, Line style or Line Width of the new entry to set the attribute.

Moving a Palette Entry

1. In the list, select the entry which you would like to move.
2. Click Up or Down.

Changing a Palette Entry

1. Click on Color, Line style or Line Width of an entry to change its attribute.

Deleting a Palette Entry

1. In the list, you select the entry which you would like to delete.
2. Click Delete.

Transferring a Palette from One Diagram to Another

1. Open the tab of the diagram with the palette that you want to transfer, as described above.
2. Click Save.
3. Select the saving location, enter a file name, and click OK.
4. Close the Properties dialog box by clicking Cancel.

5. Open the tab of the diagram to which you want to assign the palette, as described above.
6. Click Load.
7. Select the previously saved database and then click OK.

Displaying Perpendicular Lines

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.

2D Diagram:

2. Double-click on the curve from whose points you would like to drop perpendicular lines onto the X or Y axis or onto its axis label.
3. In the dialog box that appears, click on the Symbols tab.
4. Select the option Symbols on.
5. In the Type field, select Perpendicular on Y axis or Perpendicular on X axis.

3D Diagram with Space Curve:

2. Double-click on the curve from whose points you would like to drop perpendicular lines onto a plane.
3. In the dialog box that appears, click on the Perpendiculars tab.
4. Select the option that corresponds to the plane onto which you would like to drop the perpendicular line.
5. Set the appearance of the perpendicular lines in the corresponding line attribute fields.

Note You can add separate perpendicular lines to a curve by activating cursors and place at the desired position and then in the menu Cursors[Marker] > Place select Horizontal perpendicular or Vertical perpendicular.

Editing Perpendicular Lines

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on the perpendicular line that you would like to edit.
3. Under Format[Lines] select a line color or width.

Adding a Projection

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve that you would like to project.

3D Diagram with Space Curve:

3. In the dialog box that appears, click on the Projections tab.
4. Select the planes on which you would like to project the space curve.
5. Set the appearance of the projection lines in the associated line attribute fields.

3D Diagram with Surface:

3. In the dialog box that appears, click on the Fill tab.
4. Under Projections select the planes on which you would like to project the surface.

3D Diagram with Grid:

3. In the dialog box that appears, click on the Grid Lines tab.
4. In the Projection on field, select the planes on which you would like to project the grid.

Changing the Symbol Size

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Click on any symbol on the curve. You can select several curves at the same time by holding down the CTRL key and clicking on them. You can select all curves by pressing the ALT key.

3. Set the symbol size in the edit box Format[Measures] > Width.

Changing the Symbol Shape

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the curve or its axis label.
3. In the Properties dialog box 107 of the curve you can set the symbol type in the Type drop down box on the Symbols tab. If you choose Automatic, FlexPro automatically selects a symbol using the number of the curve or the data series in a 3D curve.

Changing the Number of Symbols

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on any curve symbol to open its Properties dialog box 107.
3. On the Symbols tab, for the Placement mode select Relative and enter the desired value in the Number field. In the Start at field, enter the position of the first symbol as a percentage of the distance resulting from the total count.

Working with Legends

Adding or Deleting a Legend

To add a legend:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Open the list box under Diagram Design[Diagram Layout] > Add Diagram Element and select Legend.

To remove the legend:

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the diagram plane to open the Properties dialog box 107.
3. Uncheck the option Use legend.

Positioning the Legend

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Select the legend by clicking on its border. If there is no border, click on the background. The legend will now appear with a hatched border.
3. Choose a Position in Format[Align] to specify on which side of the diagram the legend is to appear.
4. If necessary, use the mouse to move the legend in order to adjust its position.

Adding or Deleting a Curve in a Legend

If the diagram is in a document or a worksheet, double-click on the diagram to open it.

1. Click on the curve that should be added to or removed from the legend in order to select it.
2. Check the option Diagram Design[Curve] > Legend or remove the checkmark.

Displaying Data Set Information in the Legend

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click in the legend on the curve text that you want to change.
3. In the Text input field, place the cursor at the position where you would like to insert a field.
4. Select the desired item from the Fields list box. A field is then inserted into the text. This field is later replaced by the updated data set information.

Note: If you leave the text input field empty, the selected diagram style will determine the text.

Adding a Title to the Legend

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the legend title, edge of the legend or diagram surface, but not on a curve in the legend.
3. In the Properties dialog box 107 click on the Legend tab, if necessary.
4. Select the option Title on.
5. Enter the desired text in the Text input field.

Changing the Number of Columns in a Legend

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the legend title, edge of the legend or diagram surface, but not on a curve in the legend.
3. In the Properties dialog box 107 click on the Legend tab, if necessary.
4. Under Layout enter the desired value in the Number of columns field.
5. You can adjust the arrangement of the entries using the Arrange entries line by line option.

Adding or Deleting a Legend Frame

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Double-click on the legend title, edge of the legend or diagram surface, but not on a curve in the legend.
3. In the Properties dialog box 107 click on the Legend tab, if necessary.
4. Under Style select the option Border or remove the checkmark.

6.2 Column, Row and Cell Tables

You can use these presentation objects 378 to display data in table form. You can use the Table Wizard to create column, row and cell tables, which offers an assortment of templates.

Column Table

The column table consists of one or more columns. One data set can be assigned to each column, supplying all of the data for this column. Therefore, the column table is particularly suited for displaying data sets containing more than one value, i.e. data series, data matrices, signals or signal series. A typical example would be a data series with the local maxima of a signal. Every column has one column title where fields for the name and unit of the data set can be inserted. Due to this structure, the behavior of the column table is particularly dynamic. The number of values supplied by a data set for a column may vary. The number of rows in the table then changes automatically. You can even display a list containing several data sets. In this case the number of table columns automatically adjusts to the number of data sets in the list.

The column table supports a variety of design options, such as any writing direction, alignment of numbers to the decimal point, and wrapping of longer strings.

The following figure shows a column table with an index column, a column with a data series and a column with a data matrix. A column index is also added to the last one.

Column Table

Index Column	Column With Data Series	Column With Data Matrix		
		1	2	3
1	17.11	0	15.95	5.01
2	2.49	3.66	19.07	14.64
3	6.15	10.71	16.98	15.76
4	11.47	18.04	17.81	3.42
5	9.46	15.6	12.57	14.28
6	15.24	19.3	3.71	9.62
7	3.84	3.95	12.34	1.63
8	14.56	18.03	5.3	16.13

Data Columns, Index Column and Column Index

In the column table you can display all data structures supported by FlexPro except for signal series with two-dimensional X components. Composite data structures 122 are displayed in several columns. For example, if you add a signal, two columns are displayed side by side in the column table: one column with the X component of the signal and one column with the Y component of the signal. All data types 118 are permitted, particularly including complex numbers and strings.

If a column represents a data matrix or signal series, then this column is divided into several sub-columns, one for each data series in the (Y) data matrix. The Z component of a signal series is displayed as a column index under the column title.

In addition to the data columns, an index column can be added to the column table, and you can use this column to number the rows in the table. You can give the complete table a text label that you can position above or below it.

Unit

You can specify an output unit for each data column. The corresponding data is then converted to this unit before output. If you do not specify an output unit, the data is output in the unit in which it occurs.

Multi-Page Column Tables

If you display column tables in the object window or insert them into FlexPro documents, these column tables can be wrapped across several pages in the document. Wrapping occurs automatically if not all columns of the table fit next to each other or if not all data rows of the columns to be displayed fit on one page.

When wrapping due to too many columns, you can specify that all columns should be displayed on the same page. The subsequent columns will then be displayed further down on the same page, but this means that fewer rows can be displayed.

When wrapping due to too many data rows, FlexPro first tries to wrap the column on the same page, and the number of columns in the table is multiplied. The column table is not continued on the next page until all of the space on the first page is fully occupied. The titles appear on every page.

Note This pagination option is only available for display in the object window and in FlexPro documents. If you export the column table as an OLE object, only the first page is displayed.

Row Table

The row table is particularly suitable for displaying scalar values, such as measurement parameters or statistical results. Using the Orientation attribute, you can convert a column table into a row table and vice versa. The table is then essentially mirrored on the diagonal, i.e. columns become rows and vice versa. The same design options are available for the row table as for the column table.

The following row table corresponds to the column table above. In addition to the orientation, only the writing direction for the title of the data matrix has been changed.

Row Table

Row Index		1	2	3	4	5	6	7	8
Row with Data Series		17.11	2.49	6.15	11.47	9.46	15.24	3.84	14.56
Row with Data Matrix	1	0	3.66	10.71	18.04	15.6	19.3	3.95	18.03
	2	15.95	19.07	16.98	17.81	12.57	3.71	12.34	5.3
	3	5.01	14.64	15.76	3.42	14.28	9.62	1.63	16.13

Multi-Page Row Tables

If you display row tables in the object window or insert them into FlexPro documents, these tables can be wrapped across several pages in the document. Wrapping is done automatically if not all rows of the table fit under each other or if not all data columns of the rows to be displayed fit on one page.

When wrapping due to too many rows, you can specify that all rows should be displayed on the same page. The subsequent rows will then be displayed further down on the same page, but this means that fewer columns can be displayed.

When wrapping due to too many data columns, FlexPro first tries to wrap the row on the same page, and the number of rows in the table is multiplied. The row table is not continued on the next page until all of the space on the first page is fully occupied. The titles appear on every page.

Note This pagination option is only available for display in the object window and in FlexPro documents. If you export the row table as an OLE object, only the first page is displayed.

Sorting the Data

You can now simply output the data of a column or row table sorted by any column or row. This column then specifies the order in which all the data in the table is displayed.

Note If the columns or rows contain different numbers of values, the shorter ones are filled in. This is done for floating point data with void values, for integer data with the value 0 and for string data with empty strings.

Conditional Formatting

Usually the color of the table cells comes from the color scheme selected for the table. However, you can also use conditional formatting to visualize the relative quantity of the values displayed. For instance, display a small bar after the numerical value or vary the color of the cell background and/or cell text.

Bar Graphics	Background Color	Text Color
0,0000	0,0000	0,0000
0,6428	0,6428	0,6428
0,9848	0,9848	0,9848
0,8660	0,8660	0,8660
0,3420	0,3420	0,3420
-0,3420	-0,3420	-0,3420
-0,8660	-0,8660	-0,8660
-0,9848	-0,9848	-0,9848
-0,6428	-0,6428	-0,6428

Cell Table

The cell table is made up of several cells arranged as columns and rows. Each cell may contain an individual text where you can embed fields for calculation results. The cell table corresponds to the tables you might already be familiar with from your word processor.

The following illustration shows a typical cell table:

	Quantity	Value
Statistics	Maximum	2.50 V
	Minimum	0.50 V
Mean Values	Arithmetic Mean	1.49 V
	Square Mean (RMS)	1.59 V

Working with Tables

Creating a Table

1. In the Folders window, select the folder where you want to place the new column. row or cell table.
2. Select the data you want to display as columns, rows or cells in the table. You can select data sets, formulas, analysis objects or a cell range in a folder's data view or in the window of an open data set. You can also use the range between the cursors of a diagram as the data selection.
3. Choose an appropriate table from one of the galleries under Insert[Tables] or click Insert[Tables] > Table Wizard if you want to create a special view. You can obtain additional information using Help in the Table Wizard.

Changing Table Style and Table Color

1. If the diagram is in a document or a worksheet, double-click on the diagram to open it.
2. Under Design[Style] select Tables, Colors and Background.

Notes

- All table attributes that are controlled by the table style, color and background are reset automatically. If you do not want this to happen, change the settings for Table Style, Colors and Table Color on the Style tab of the table Properties dialog box.

- The table color is a color from the palette selected in the Colors list box.
- The first palette in the Colors list box can be customized. You can change these palette colors as well as additional attributes assigned to the selected table style by opening the list box under Design[Style] > Table Style and clicking on Edit Style Sheet.

Aligning the Table

1. Make sure that no element of the table is selected by pressing the ESC key.
2. Click on the desired alignment under Table Design[Alignment].

Conditional Formatting of a Table

1. Select one or more columns or cells of the table.
2. Select Columns/Cell Table Design[Style Sheets] > Conditional Formatting and select the desired formatting.
3. Depending on the formatting you choose, a dialog box appears where you can specify the formatting attributes.

See Also

Column, Row and Cell Tables 441

Adding or Deleting Dividing Lines and Frames

To delete a dividing line or the frame, do the following:

1. Click on the line or frame while holding down the SHIFT key to select the item.
2. Now choose No Line from Format[Lines] > Color.

To add a dividing line or the frame, do the following:

1. Make sure that no item in the table is selected by clicking on a point outside the table.
2. Now use your right mouse button to click on the table and select Properties.
3. In the dialog box, select the Grid Lines & Background tab.
4. Select the option corresponding to the item to be displayed.

Changing the Color of Dividing Lines, Frames, Text and the Background

1. Select the objects that have a color you would like to change. You can select several objects by holding down the CTRL key and clicking on them.
2. You can change the font color of the selected elements by selecting a color in the Format[Font] > Text Color menu. You can change the color of lines and fills accordingly using the Format[Lines] > Color or Format[Fill] > Color menus.

Changing the Text Font

1. Select all of the text in the table that has a font you would like to change. Press the CTRL key to select several lines of text. Press the ALT key to select all text.
2. Select the desired font. Format[Font] > Font.

Changing the Text Alignment

1. Click on the column title or cell text that has the alignment you would like to change.
2. You can use calculations in Format[Align] click on the icon showing the desired text alignment.

Changing the Text Direction

1. Click the title or cell text whose direction you want to change.
2. Under Format[Measures] > Angle specify the desired angle.

Changing the Padding Between Text and Cell Borders

1. Double-click on the cell text for which you want to change the distance from the cell border.
2. In the Properties dialog box that appears, enter the desired values in the fields Padding, Left, Right, Top and Bottom.

Text Wrapping for the Column Title or Cell Text

1. Double-click on the column title text or cell text that is to wrap to the next line.
2. In the Properties dialog box that appears, under Style select the option Wrap Text.

Adding or Deleting a Table Title

To add a table title, do the following:

1. Click Design[Table Layout] > Add Table Element and select Table Title.
1. Double-click on the new title.
2. Enter text for the title or from the Fields menu select an appropriate field, such as Comments.

To delete a table title, do the following:

1. Double-click on the table title.
2. In the dialog box that appears, uncheck the option Table title on.

Aligning the Table Title

1. Click on the table title to select it.
2. Click on the desired alignment under Format[Align].

Embedding Pictures

1. Click Table Design[Illustrations] > Pictures.
1. Choose a file in the Insert Picture dialog box.
2. Click OK to close the dialog box. The newly inserted image is now displayed.
3. If necessary, drag the image with your mouse to position it.
4. To define the absolute height and width, click Format[Measures]. You can change the relative size in relation to the original image size in the following Properties window fields: Scaling Width and Scaling Height.

Sorting the Data

1. Click on the title of the column or row according to whose data you want to sort the table.
1. Right-click to open the context menu and select the desired orientation under Sort.

Cancel Sorting

1. Click on the title of any column or row in the table.
1. Right-click to open the context menu and select the sub-item None under Sort to cancel the sorting and display the data in its original order again.

Working with Column Tables

Adding a Data Column

The easiest way to add a data column to a column table is to use the mouse to drag the data set to be displayed in the column from the object list into the column table. The new column is then inserted at the location selected and adopts the style attributes of its left neighbor.

However, you can also insert an empty column and then manually enter the data set:

1. Open the list box under Column Table Design[Table Layout] > Add Table Element and select Data Column.
2. In the dialog box that appears, enter the data set for the column.
1. Click on OK to add the column.

Adding an Index Column

- Open the list box under Column Table Design[Table Layout] > Add Table Element and select Index Column.

Specifying the Output Unit

1. Double-click on the data of the column table or column index for which you want to specify an output unit.
2. In the Properties dialog box that appears, enter the desired output unit in the Unit field or choose the unit from the list.
3. Close the dialog box with OK.

Note: If you specify an output unit, the data must be present in a compatible unit.

Rearranging Columns

- To move a column in a column table to a different location, use your mouse to drag the column to the desired location and drop it there.

Note: You can also rearrange the columns on the Columns / Rows tab in the Properties dialog box 107.

Changing the Column Width

To change the width of a column:

- To change the width of a column, move the right border of a column by dragging it with the mouse.

To adjust the width of all columns:

- Use the command Column Table Design[Columns] > Distribute Columns.

Note: The column width that you set is a minimum width. FlexPro automatically increases the column width if the data cannot be displayed using the width specified.

Changing the Height of the Rows and Column Titles

- To change the height of a row, use your mouse to drag and move the bottom edge of the row.

Note: You can set the row height for the title and the data rows separately; however, the setting for rows with data applies for all data rows. The row height is a minimum height. FlexPro automatically increases the size of the row if the content cannot be displayed with the height specified.

Changing Page Splitting in a Column Table

In a column table with many columns, not all rows may fit next to each other.

To display the remaining columns on the same page further down, proceed as follows:

- Click Row Table Design[Style] > Pagination and select the option Multiple Tables per Page.

To display the remaining columns on the next page, proceed as follows:

- Click Column Table Design[Style] > Pagination and select the option One Table per Page.

Navigating Multi-Page Column Tables

- Use the commands Row Table Design[Page] > Previous or Row Table Design[Page] > Next to go to the previous or next page of the table.
- To display the first or last page of the table, use the commands Row Table Design[Page] > Previous > First or Row Table Design[Page] > Next > Last.
- In the Row Table Design[Page] > Go To input field you can enter any page number to display it.
- You can use the mouse wheel to scroll through several pages of the table.

Enabling/Disabling the List View with Alternating Colors

To turn off the list view, do the following:

1. Double-click on a data column in the table to view the Data tab of the table's Properties dialog box.
1. Uncheck the option Color every 2nd column the following color.

To enable the list view, do the following:

1. Double-click on a data column in the table to open the Data tab of the table's Properties dialog box.
2. Under Style select the option Color every 2nd row the following color.
2. Click on the color field to set the color.

Changing the Data Format in a Column Table

To change the formatting of all values of a column, do the following:

1. Double-click on any value in the column with the format that you would like to change.
1. Under Style click on the button in the Format input field to adjust the formatter.
2. Select the desired formatting.

Using the Properties window, you can set formatting for several columns at once:

1. If the Properties window is not already open, select View[Task Windows] > Show > Properties.
2. Select the columns that have formatting that you would like to change by clicking on these while holding down the CTRL key. You can select all columns by pressing the ALT key.
3. Now click in the Format input field of the Properties window and then click on the button that appears to the right of the box.
4. Select the desired formatting.

Note: The format applies to all values in the column(s) selected.

Displaying the Unit Symbol after the Data Values

1. Double-click on the data of the column in the table that you want to display with unit symbol.
2. In the Properties dialog box that appears, click on the button in the Format field.
3. In the Formatter Properties dialog box on the Numerical Value tab, select the option Append unit.
4. Close both dialog boxes by clicking OK.

Adding or Deleting a Column Title

To delete a column title, do the following:

1. Double-click on the title that you want to remove.

2. In the dialog box that appears, uncheck the option Title on.

To add a column title, do the following:

1. Double-click on the column to be given a title.
2. In the dialog box, select the Title, X Title or Z Title tab, depending on the data component for which you want to display a title.
3. Select the option Title on.
4. Enter text for the title. You can embed fields for displaying header information from the data set.

Displaying Data Set Information in the Column Title

1. Double-click on the title into which you would like to insert header information from the data set. The tab Title, X Title or Z Title is opened in the Properties dialog box, depending on the title data component that you want to edit.
2. Check whether the option Title on is selected. If not, then select this option now.
3. In the Text field, place the cursor at the position where you would like to insert a field.
4. Choose the desired item from the Fields list box. A field is then inserted into the text. This field is later replaced by the updated data set information.

Adding or Deleting a Column or Row Index

To delete an index, do the following:

1. Double-click on the index that you would like to delete.
2. In the dialog box that appears, uncheck the option X data on or Z data on.

To add a column index, do the following:

1. Double-click on the column to be given an index.
2. In the dialog box, select the X Data tab for the row index or Z Data for the column index.
3. Check the option X data on or Z data on.
4. Select Index and enter the Starting value and Increment. Alternatively, you can take the data for the index from a data set.

Note A column index can only be used for columns with a data set that supplies a data matrix.

Working with Row Tables

Adding a Data Row

The easiest way to add a data row to a row table is to use the mouse to drag the data set to be displayed in the row from the object list into the row table. The new row is then inserted at the selected position and receives the style attributes of the row above it.

However, you can also insert an empty row and then enter the data set manually:

1. Open the list box under Row Table Design[Table Layout] > Add Table Element and select Data Row.
2. In the dialog box that then appears, enter the data set for the row.
1. Click OK to add the row.

Adding an Index Row

- Open the list box under Row Table Design[Table Layout] > Add Table Element and select Index Row.

Specifying the Output Unit

1. Double-click on the data of the data row for which you would like to specify an output unit.
2. In the Properties dialog box that appears, enter the desired output unit in the Unit field or choose the unit from the list.
3. Close the dialog box with OK.

Note: If you specify an output unit, the data must be present in a compatible unit.

Rearranging Rows

- To move a row in a row table to another position, drag it with the mouse to the desired position and drop it there.

Note: You can rearrange the rows on the Columns / Rows tab in the Properties dialog box 107.

Changing the Row Height

To change the height of a row:

- To change the height of a row, move the lower border of a row by dragging it with your mouse.

To make the height of all rows equal:

- Use the command Row Table Design[Rows] > Distribute Rows.

Note: The row height is a minimum height. FlexPro automatically increases the row size when the data cannot be displayed with the predefined height.

Changing the Width of Columns in Row Titles

- To change the width of a column, move the right border of a column by dragging it with the mouse.

Note: You can set the column width for the title and the data columns individually, but the setting for the data columns applies to all data columns. The column width that you set is a minimum width. FlexPro automatically increases the column size when the content cannot be displayed with the predefined width.

Changing Page Splitting in a Row Table

In a row table with many rows, not all rows may fit under each other.

To display the remaining rows on the same page further to the right, proceed as follows:

- Click Row Table Design[Style] > Pagination and select the option Multiple Tables per Page.

To display the remaining rows on the next page, proceed as follows:

- Click Row Table Design[Style] > Pagination and select the option One Table per Page.

Navigating Multi-Page Row Tables

- Use the commands Row Table Design[Page] > Previous or Row Table Design[Page] > Next to go to the previous or next page of the table.
- To display the first or last page of the table, use the commands Row Table Design[Page] > Previous > First or Row Table Design[Page] > Next > Last.
- In the Row Table Design[Page] > Go To input field you can enter any page number to display it.
- You can use the mouse wheel to scroll through several pages of the table.

Enabling/Disabling the List View with Alternating Colors

To turn off the list view, do the following:

1. Double-click on a data row in the table to open the Data tab of the table Properties dialog box.
2. Uncheck the option Color every 2nd column the following color.

To enable the list view, do the following:

1. Double-click on a data row in the table to open the Data tab of the table Properties dialog box.
3. Under Style select the option Color every 2nd row the following color.
2. Click on the color field to set the color.

Changing the Data Format in a Row Table

To change the formatting of all values in a row, do the following:

1. Double-click on any value with a format that you would like to change.
1. Under Style click on the button in the Format input field to adjust the formatter.
2. Select the desired formatting.

Using the Properties window, you can set the formatting for several rows at once:

1. If you do not already see the Properties window, select View[Window] > Task Window > Properties.
2. Select the rows whose formatting you would like to change by clicking on them while holding down the CTRL key. You can select all rows by holding down the ALT key.
3. Now click in the Format input field of the Properties window and then click on the button that appears to the right of the field.
4. Select the desired formatting.

Note: The formatting applies to all of the values for the selected row(s).

Displaying the Unit Symbol after the Data Values

1. Double-click on the data of the row in the table that you want to display with unit symbol.
2. In the Properties dialog box that appears, click on the button in the Format field.
3. In the Formatter Properties dialog box on the Numerical Value tab, select the option Append unit.
4. Close both dialog boxes by clicking OK.

Adding or Deleting a Row Title

To delete a row title, do the following:

1. Double-click on the title that you want to remove.
2. In the dialog box that appears, uncheck the option Title on.

To add a row title, do the following:

1. Double-click on the row that should have a title.
2. In the dialog box, select the Title, X Title or Z Title tab, depending on the data component for which you want to display a title.
3. Select the option Title on.
4. Enter text for the title. You can embed fields for displaying header information from the data set.

Displaying Data Set Information in the Row Title

1. Double-click on the title into which you would like to insert header information from the data set. The tab Title, X Title or Z Title is opened in the Properties dialog box, depending on the title data component that you want to edit.
2. Check whether the option Title on is selected. If not, then select this option now.
3. In the Text field, place the cursor at the position where you would like to insert a field.
4. Choose the desired item from the Fields list box. A field is then inserted into the text. This field is later replaced by the updated data set information.

Adding or Deleting a Column or Row Index

To delete an index, do the following:

1. Double-click on the index that you would like to delete.
2. In the dialog box that appears, uncheck the option X data on or Z data on.

To add an index, do the following:

1. Double-click on the row that should have an index.
2. In the dialog box, select the X Data tab for the column index or Z Data for the row index.
3. Check the option X data on or Z data on.
4. Select Index and enter the Starting value and Increment. Alternatively, you can take the data for the index from a data set.

Note: A row index can only be used for rows whose data set provides a data matrix.

Working with Cell Tables

Adding a Column

1. Double-click on the cell table in the worksheet or document to open it.
2. Click to select any cell in the column before which you want to insert a new column. If you want to add a new column behind the last column, no cell in the cell table should be highlighted.
3. Click Design[Table Layout] > Add Table Element and select Column.

Adding a Row

1. Double-click on the cell table in the worksheet or document to open it.
2. Click to select any cell in the row before which you want to insert a new row. If you want to add a new row behind the last row, no cell in the cell table should be highlighted.
3. Click Design[Table Layout] > Add Table Element and select Row.

Note: If you have highlighted the last cell in the bottom right-hand corner, you can easily add a new row by pressing the TAB key. The same thing works if you select the first cell in the upper left-hand corner and press the TAB key while holding down the SHIFT key.

Deleting Columns or Rows

1. Select the column or row that you want to delete by clicking on an area below or to the right of the edge of the column or row. You can select several columns or rows by holding down the CTRL key.
2. Use the command Design[Selection] > Delete Rows/Columns.

Rearranging Columns or Rows

1. Select the column or row that you want to move by clicking on an area below or to the right of the edge of the column or row.
2. Use your mouse to drag the selected column or row to the desired location and drop it there.

Specifying the Output Unit of Data Displayed in a Cell

1. Double-click on a value displayed in a table cell in order to change its output unit.
2. In the dialog box that appears, place the insertion point on the field to be changed, directly after the { character, which indicates the field. A typical field, for instance, is `%&{Velocity}`.
3. Enter the operator `Unit<`*Unit Symbol*`>` directly after the { character, e.g. `%&{Unit<km/h> Velocity}`.
4. Close the dialog box with OK.

Note: If you specify an output unit, the data must be present in a compatible unit.

Changing the Column Width

To change the width of a column:

- Move the right margin of the column by dragging it with the mouse.

To adjust the width of several columns:

1. Select the columns you want to make the same width by clicking on the area below the edge of the columns while holding down the CTRL key. You can select all columns by clicking on the table and pressing the ALT key.
2. Click Design[Selection] > Distribute Columns.

Note: The column width that you set is a minimum width. FlexPro expands the column automatically if the text cannot be displayed using the pre-configured width.

Changing the Row Height

To change the height of a row, do the following:

- Move the lower edge of a row by dragging it with your mouse.

To adjust the height of several rows, do the following:

1. Select the rows that you want to adjust for height by clicking on a location to the right of the row while holding down the CTRL key. You can select all rows by clicking the table and pressing the ALT key.
2. Click on the Design[Rows and Columns] > Distribute Rows.

Note: The row height is a minimum height. FlexPro expands the row automatically if the text cannot be displayed properly using the given height.

Editing Cell Text

1. Double-click on the cell table in the worksheet or document to open it.
2. Select the cell with text that you want to edit or in which you want to enter new text.
3. To insert new text into the cell, simply type it in.
4. If you want to edit existing text, press F2 or click again on the cell that is already selected and then edit the text.
5. If you not only want to enter text, but also want to format it and insert fields, press the ENTER key or double-click on the cell. A dialog box appears, allowing you to edit the text.

Displaying Data in a Cell

1. Use your mouse to drag the data set from the object list onto any cell in the cell table.
2. In the cell text, a field is automatically inserted for the value and the unit of the data set.

Changing the Display Format of Data in a Cell

1. Double-click on the value in a cell of the table with values for which you would like to change the formatting.
2. Place the cursor onto the formatter to be changed, directly after the % character, which indicates the formatter. A typical formatter, for instance, is `%&{DataSet}`.
3. Click on Edit Formatter above the input field to set the formatter.
4. Select the desired formatting.

Merging Cells

1. Double-click on the cell table in the worksheet or document to open it.
2. Select the adjacent cells that you want to merge into one cell by clicking on them while holding down the CTRL key.
3. Use your right mouse button to click on the table and then click Design[Rows and Columns] > Merge Cells.

Note: When merging cells, the cell in the upper left-hand corner of the selected range is enlarged so that it covers all of the other cells. You can use the command Split Cells to undo this step, making the hidden cells visible again.

6.3 Text

This presentation object shows a block of text with embedded calculation results. You can freely format the text as you would in a word processing program. To display values and other information, insert fields into the text. These fields are later replaced by data from data sets or by other information.

Multi-Page Text

When you insert a text object into a FlexPro document, the text can be wrapped over several pages. Wrapping is automatic if the complete text does not fit into the rectangle placed on the first page.

Working with Text

Creating Text

To add text:

1. In the Folders window, select the folder where you want to place the new text.
2. Click on the icon under Insert[Other] > Text.

To add text with embedded computational results:

1. In the Folders window, select the folder where you want to place the new text.
3. Select the formulas and data sets in the object list that you want to embed.
2. Click Insert[Other] > Text > Standard.

Editing Text

- If the text is in a document or a worksheet, double-click on the text to open it. Otherwise, double-click on the text in the object list.

The text window that appears for text stored in the object list has two views. In Draft view, enter the text and insert fields for calculated results. In Preview the text is displayed using the replaced fields.

Deleting Text

1. Select the text that you would like to delete.
2. To remove the text for re-insertion into a different location in the document, click Home[Clipboard] > Cut. To delete the text without copying it to the clipboard, hold down the DELETE key.

Notes: Click anywhere in the text to cancel a selection. To undo a deletion, click on Undo in the Quick Access Toolbar.

Copying Text

1. In the text containing the information to be copied, while holding down the left mouse button, drag the cursor across the information you would like to select.
2. Click Home[Clipboard] > Copy.

3. In the text where the information is to be inserted, click on the location where you would like to place the information.
4. Click Home[Clipboard] > Paste.

Note: You can insert the information multiple times.

Searching for Text

1. Click on the position in the text where you would like to start the search.
2. Click Text Design/Format[Edit] > Find and enter the text you want to search for in the Find what field.
3. To find another location with the same text, continue by clicking on Find Next.

Note: To find another location with the same text after you have closed the Find dialog box, press the F3 key.

To search and replace text, click Text Design/Format[Edit] > Replace instead of Find.

Searching and Replacing Text

1. Click on the location in the text from which you would like to start replacing text.
2. Click Text Design/Format[Edit] > Replace, and then enter the text you want to find and the text with which you want to replace it.
3. To replace the found text throughout the document without confirmation, click on Replace All.

To replace the found text at the particular places where it appears, click on Find Next and then Replace.

Changing the Font, Font Style and Font Size

1. Select the text that you would like to format.
2. Choose a font or font size under Text Design/Format[Font].

Note: You can specify the font for new text by changing the settings for the font before you start to enter the text.

Changing Font Color, Text Highlight Color and Background Color

To change the font color, do the following:

1. Select the text with the color that you want to change. Use your mouse or press CTRL+A to select the entire text.
2. Now choose a color from the following menu: Text Design/Format[Font] > Text Color.

To change the text highlight color, do the following:

1. Select the text that you want to highlight in color. Use your mouse or press CTRL+A to select the entire text.
2. Text Design/Format[Font] > Text Highlight Color.

To change the background color, do the following:

- Now choose a color from the following menu: Text Design/Format[Back Color].

Formatting Paragraphs

1. Click on the paragraph that you would like to format.
2. Click Text Design/Format[Paragraph] and select the desired justification and indents.

Note: To set the paragraph format for new text, change the format settings before you start to enter the text.

Setting Tabs in Paragraphs

1. Select the paragraph for which you would like to set tabs.
2. Click Text Design/Format[Ident] > Tabs.
3. To set a tab, enter the position in the Tab position field. Next, click on Set.

Notes

- To delete a tab, open the dialog box Text Design/Format[Indent] > Tabs, select it in the list of tabs and then click on Clear.

- To delete all of the tabs in the paragraph selected, click on Clear All in the Tabs dialog box.
- To place tabs with the ruler, click on the ruler where the tab stops are to be set.
- To delete tabs with the ruler, use your mouse to select them and drag them from the ruler.

Creating a List

1. Place the cursor at the location in the document where the list is to start.
2. Click Text Design/Format[Paragraph] > Bullets. Now enter the text for the list.
3. If you press the ENTER key, another bullet appears in the next line.
4. To end the list, click again on Text Design/Format[Paragraph] > Bullets.

Inserting a Date or Time Field

1. Click on the location in the text where the field is to be inserted.
2. Click Text Design/Format[Insert] > Date or Design/Format[Insert] > Time to add the current date or time.

Note: The field is replaced by the corresponding part of the system date whenever the text is updated.

Inserting a Field to Display Data or Other Elements

1. Click on the location in the text where the field is to be inserted.
2. Click Text Design/Format[Insert] > Field. The Field Wizard appears to guide you through the remaining steps.

Inserting Fields for the Data Set Name, Value and Unit

You can display the data set name, value and unit in a text object by simply using your mouse to drag the data set from the object list into the text object and drop it at the location where the fields are to be inserted.

Inserting a User Prompt

1. Click on the location in the text where the field is to be inserted.
2. Click Text Design/Format[Insert] > Field to activate the Field Wizard.
3. Select the Enter category and the User Input field.
4. Click on Next and follow the Wizard's instructions.

Post-Editing a Field Formatter

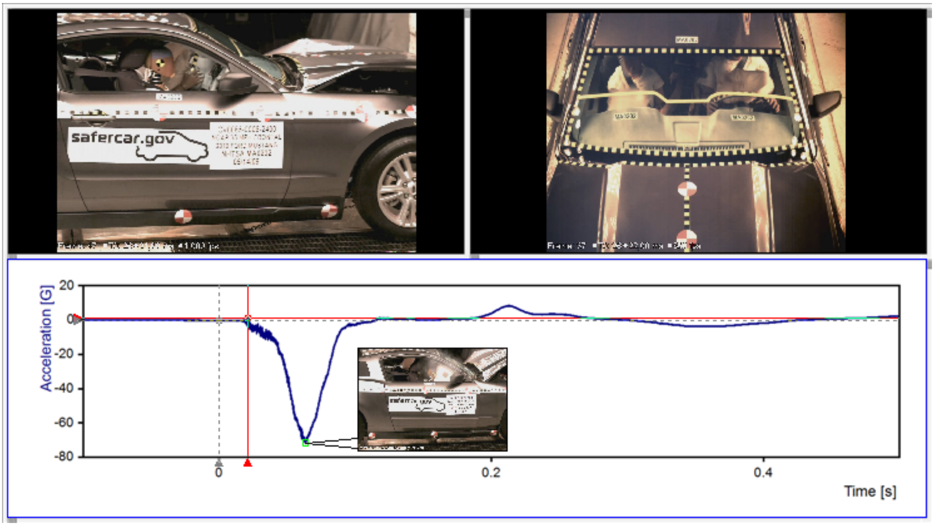
1. Place the cursor on the formatter to be changed, directly after the % sign, with which the formatter starts. A typical formatter is, for instance, `%f{DataSet}`.
2. Now click Text Design/Format[Selection] > Formatter.

A dialog box appears where you can set the desired options.

6.4 Media

You can use the Media presentation object to play video and audio files and to analyze videos and audio signals that you have recorded synchronously with physical quantities.

This presentation object is only available in FlexPro Professional and FlexPro Developer Suite.



You can use media such as diagrams and tables and embed or link them in documents and worksheets. FlexPro can synchronize the cursors of the curves whose X component has a valid time unit with the media. When you play the media, the cursors move over these curves. If you place the cursor at a certain position on the curve, the media will display the corresponding still image.

Source

The source to be played can be either a media file containing a video or audio signal, or a data set in the project database, which is then interpreted as an audio signal.

Time Base

The frame number or time assigned to the frame is used to assign frames in a video. The time base assigns these times to the frames. This is also used to synchronize the cursors in diagrams when you play media. FlexPro supports three modes:

- From media file
  The time information is determined from the number of frames and the playback length of the video. The time 0 s is assigned to the first frame in the video.
- Starting time and frame rate
  In this case, enter the start time and the frame rate. The start time is the time in

seconds assigned to the first frame in the video. The frame rate specifies the number of frames per second. You can use this time base for videos whose frame rate specified in the media file deviates from the frame rate in which the video was originally recorded. This is usually the case for videos recorded on high-speed cameras at 1000 frames per second but are to be played back at 24 frames per second, for instance.

- Two frames with frame times
  The time information is calculated from the times of two random frames. These variants are ideal for videos with opening credits, for instance, in which the time of the first frame is not really known. You can enter the times either using the format +-HH:MM:SS.sss or using any time unit, such as -25.3 ms. For the frame numbers, enter the one-based numbers of the corresponding frames.
- From data set
  Here you enter a data series with the time values for all frames. The time values must be increasing, but not equidistant. Use this setting for videos recorded at variable frame rates using an external timer.

Note: Some video formats do not support positioning via frame numbers. Naturally, the same applies to all audio formats as well. In this case, FlexPro adopts a total of 24 frames or positions per second to which you can move. The number of frames or navigation positions are then the result of the playback length of the media file.

Still Image

The Media object takes an individual image from the video and displays it when the video is not playing. Likewise, when you print out a document in which you placed a Media object, this still image is printed. You can choose the still image using the cursor or by entering the time of the image you want displayed in the Properties dialog box. As in other presentation objects, you can draw 379 in the Media object in order to mark points of interest in the still image with arrows.

Image Marker

The media object can be used as a source for image markers that you position on curves. When you place this type of marker on a curve, the still image associated with the X position is displayed in the marker.

Compatible Audio and Video Formats

The Media object is based on the same technology as Windows Media Player. All formats that Windows Media Player can play are therefore also compatible with FlexPro. If a media file cannot be played, you can subsequently install the relevant Windows Media Player codec.

* This presentation object is not available in FlexPro View and FlexPro Basic.

Working with Media

Creating Media

Manually Creating Media That Plays a Media File

1. In the Folders window, select the folder where the new media is to be placed.
2. If you want to play back audio signals stored in data sets, then mark these data sets now. If you select several data sets, their audio signals will be mixed.
3. If you want to embed the new media into a worksheet or document at this time, click on the open worksheet or document window. Otherwise, click in the object list.
4. Click Insert[Other] > Media.
5. In the dialog box that now appears, enter the path name of the media file on the Source tab.

Note: The media file is not copied to the object, but has to be found under the specified path name each time it is played. Therefore, you should either save your media files in a folder of your choosing or copy them to the same folder in which your FlexPro project database resides. If you choose the second method, you should only enter the file name and not the full path name in the Media object so that the link remains when you copy your project.

Manually Creating Media That Plays One or More Data Sets as an Audio Signal

1. In the Folders window, select the folder where the new media is to be placed.
2. Select one or more data sets that contain audio signals. If you select several data sets, their audio signals will be mixed.

3. If you want to embed the new media into a worksheet or document at this time, click on the open worksheet or document window. Otherwise, click in the object list.
4. Click Insert[Other] > Media.

Note: The data set must either contain one signal or a signal series with two columns. A signal series is played back as a stereo signal. If you select multiple data sets, they should be the same length.

Adding Media via Drag-and-Drop

- Select the media file in Windows Explorer and use drag-and-drop to drag it to the FlexPro Object List or to an open worksheet or document.

Creating the Time Base

1. Right-click with your mouse on the media and select Properties.
2. In the Properties dialog box that now appears, click on the tab called Time Base.
3. Choose the preferred Time Base and adjust the necessary settings. Launch Help in the dialog box if necessary.

Playing Media

Playing Media from the object list

1. Double-click on the object to open it.
2. Click Media Design[Play] > Play.
3. Click Media Design[Play] > Pause to pause playback.
4. Click Media Design[Play] > Stop to stop playback.

Note The media is played back from the beginning. If you want to play the media back from the position of the still image displayed, you have to activate the cursors before starting playback. When you disable the cursors again, FlexPro uses the current position as the new still image. If you want to keep the original still image, you have to choose Cursors[Zoom Curve] > Restore Image Section.

Playing Media from a Document or Worksheet

1. If the media is in a document, click Design[Cursors] > On/Off.
2. Click on the media that you want to play.
3. Click Cursors[Play] > Play.
4. Click Cursors[Play] > Pause to pause playback.
5. Click Cursors[Play] > Stop to stop playback.
6. If necessary, click Cursors[Zoom Curve] > Restore Image Section to restore the originally displayed still image.

Adjusting the Volume

With Cursors Enabled

- Move the vertical scrollbar on the media object to adjust the volume.

With Cursors Disabled

- Adjust the slider under Media Design[Play] > Volume to the desired setting.

Changing Playback Speed

- Enter the ratio between the desired and "natural" playback speed under Media Design[Play] > Playback Speed.

Note Not all media formats support the ability to increase or reduce playback speed. In this case, the speed set for the next time the media is played is reset to 1:1.

Choosing the Still Image for Display

With Cursors Enabled

- In the Properties window enter the zero-based position index of the preferred still image or its X position in seconds.

or

- Use the horizontal scrollbar to navigate to the desired image.

With Cursors Disabled

- Enter the number of the frame in the field under Media Tools/Design[Play] > Frame Number or its position in the field under Media Tools/Design[Play] > Position either in the format +-HH:MM:SS.ssss or in any time unit, such as "120.3 ms". "120.3 ms".

6.5 Document

FlexPro lets you create and manage multipage documents directly in the project database. In this type of document you can then embed diagrams 381, text 459, tables 437 and media 464 or create links to objects within the project database.

The powerful FlexPro Document Editor also lets you create your own drawings, such as the sketch of a test setup, thereby creating expressive and appealing documents. You can also embed any OLE objects and images, such as your company logo, into the documents.

Sections

You can split a single document into several sections to use different headers and footers or page layouts and orientation within that document. For example, you can start a new document section to display a diagram on a page.

Page Layout

The page layout of a document specifies the size of the document pages and the size of the margins. The page size does not necessarily have to be the same size as that used for printing. FlexPro can adjust the document to the paper size later. However, if you want a printout that is true to scale, you should also use your printer's page size for documents.

The page margins are only there to provide onscreen visual orientation. FlexPro lets you draw outside the margins that have been set.

Headers and Footers

You can create headers and footers semi-automatically by selecting the option Document Design[Header] > Header and Footer and then place shapes or images on the top or bottom of the document page. Alternatively, for individual shapes you can use the Add to Header command. These shapes are then displayed at the same location on every page of the document section. When printing on both sides of a page, the headers and footers of the left and right sides usually differ. For this purpose, select the option Different odd and even pages. The first page often differs from the subsequent pages. Selecting the option Different First Page gives the first page its own header.

Multipage Presentation Objects

The use of the document editor integrated in FlexPro also offers you additional options, which are not available in your word processing program. For example, 2D diagrams and column tables can wrap automatically on subsequent pages. The number of pages adjusts automatically to the amount of data. Diagrams in this case are the same size on all pages, but you can position them on each page individually. In the case of tables, it is possible to adjust not only the position, but also the height of each individual page.

Document Collection

You can add several subdocuments as links in the main document, such as the analysis results from measurements in a series of tests. To do this, in the main document you can add a document collection that you can use to search for documents via a search path or data query, which should be fully or partially embedded into the main document.

Also refer to the project database example Document Collection and Series Analysis.fpd in the folder `C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples` or `C:>Users>Public>Public Documents>Weisang>FlexPro><%VERSION_COMMERCIAL%>>Examples`.

Table of Contents and Table of Figures

You can add a table of contents as well as a table of figures for diagrams, tables, pictures and formulas to a document. These indexes are created automatically when FlexPro searches the document for captions, headings or text boxes for which you have defined an outline level or caption category. If you have embedded subdocuments into a document using a data collection, their headings will also be added to the index. If necessary, you can downgrade the outline level for embedded documents in order to integrate them into the outline of the main document.

Virtual and Real Document Pages

When you add a document collection, a multipage diagram, a multipage table or a table of contents to a document, you can define whether the additional pages of the object should be displayed on the existing real pages of the document or if virtual blank pages should be added. Virtual pages differ from real pages in that you cannot place any other shapes on them. Instead, for shapes on the real document page where the multipage object was placed, you can specify that they should also be displayed on the virtual subsequent pages. As a rule, real pages need to be added explicitly. When updating a document, FlexPro only adds blank real pages to the end of the document if they will be needed in order to display a multipage object in its entirety.

Shape Alignment Options

If you have created a group of shapes, you can align the entire group to the left, right, top or bottom of the page or centered on the page. You can align shapes relative to each other or to the margins of pages. You do not have to select or move the objects individually. When you move or resize objects, alignment guides appear which you can use very easily to align other objects.

You can also align objects using the grid. When you draw or drag objects, the corners are aligned to the nearest grid intersection point. Even if the grid is not visible, it still helps you to align the objects automatically. You can change the grid spacing and permanently or temporarily deactivate automatic alignment to the grid.

Grouping Shapes

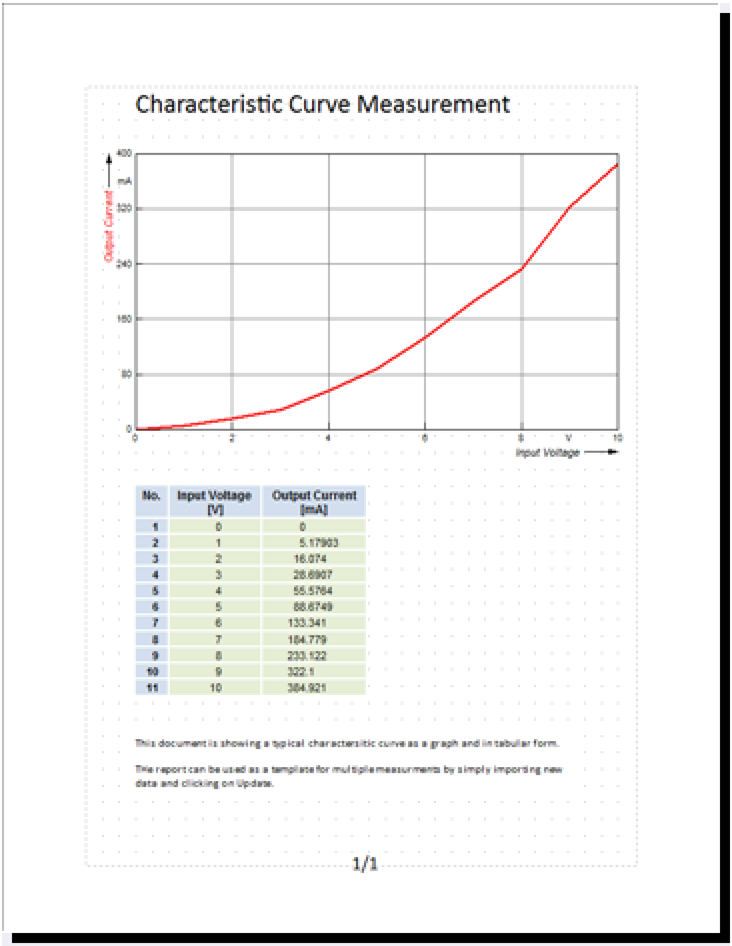
If you combine several shapes into a group, you can edit the objects as if they were one object. You can assign a property to all individual objects of a group simultaneously, such as hatching or a line color. The objects can also be resized, moved, tilted and rotated as a group.

Grouping the shapes is also very useful when creating complex graphics. You can create groups within groups and thus create a group hierarchy. You can draw a set of objects and group the objects, then draw the next set and group it, etc. Therefore, if you need to edit a section of the drawing, you can undo the grouping for this section without affecting the other groupings.

Using Cursors in the Document

You can also measure diagrams using cursors in the document. While the cursors in the worksheet are always active, in the document, you have to activate them first. You then have access to all Cursor 188 options.

The following illustration shows a typical document:



No.	Input Voltage [V]	Output Current [mA]
1	0	0
2	1	5.17903
3	2	16.074
4	3	28.6907
5	4	55.5764
6	5	88.6749
7	6	133.341
8	7	184.779
9	8	233.122
10	9	322.1
11	10	384.921

This document is showing a typical characteristic curve as a graph and in tabular form.

This report can be used as a template for multiple measurements by simply importing new data and clicking on Update.

Working with Documents

Creating a Document

To create a blank document:

1. In the Folders window select the folder where you want the new document to be placed.
2. Click on the icon under Insert[Containers] > Document.

To create a document with linked diagrams, tables and text:

1. In the Folders window select the folder where you want the new document to be placed.
2. In the object list select the diagrams, tables or text that you want to embed.
3. Click Insert[Containers] > Document > Standard.

To create a document that bundles several existing documents:

4. In the Folders window select the folder where you want the new document to be placed.
5. In the object list select the documents that you want to combine into a document.
6. Click Insert[Containers] > Document > Standard.

Setting Up the Page Layout, Orientation, Page Color and Margins

1. Open the document by double-clicking on its name in the object list.
2. Change the settings as needed. Document Design[Page Setup].

Notes

- The settings apply to the particular section in the document.
- If you change the page layout or orientation, all shapes on the pages in the section will adapt to the new page format. Click once on the Undo in Quick Access Toolbar if you do not want this to happen.
- The page margins are only there to provide onscreen visual orientation. FlexPro lets you draw outside the margins that have been set.

Adding, Moving and Deleting Pages

- Use the command Document Design[Page] > Insert After or Document Design[Page] > Insert Before to add a blank page after or before the current page. The new page will now appear in the window and can be edited.
- Use the command Document Design[Page] > Move to move the page currently displayed to a different position in the document.
- Use the command Document Design[Page] > Delete to remove the page currently displayed in the document.

Adding and Removing Sections

- Use the Document Design[Section] > Start New command to add a new document section starting from the current page.
- Use the Document Design[Section] > Cancel command to remove the document section that contains the current page. The section pages are not deleted, but are instead added to the previous section.

Inserting Page Numbers

1. Select Document Design[Fields] > Page Number. The page number and total number of pages are inserted at the bottom of the document.
2. If necessary, drag the page number field with your mouse to position it.
3. Double-click on the page number to edit the fields, such as if you want to remove the field that shows the number of pages.

Changing the Page Numbering

Changing the Numbering of a Document Section

1. Select the command Document Tools/Design[Page Setup] > Change Margins, Numbering and Paper Size.
2. In the Section list box, select the document section whose numbering you want to redefine.
3. Select the option Restart numbering with and enter the number of the first page of the document section.

Changing the Page Numbering Format

1. Double-click on a page number in the document to open the Edit Text dialog box.
2. Place the insertion point after the % sign of the %<PageNumber> field and click on Edit Formatter in the toolbar.
3. Click on the Special Format tab and select the desired format.

Adding Headings

1. Use the command Document Design[Illustrations] > Label.
2. Click on the location in the document where the left top corner of the heading should be positioned.
3. In the Edit Text dialog box that appears, enter the heading text without numbering it.
4. Next, under Level specify the outline level of the heading.
5. In the Format field you can adjust the numbering format. For example, you can set alphabetical or to Roman numeral numbering by changing the formatter of the fields %<H1> through %<H9>.
6. If you do not want the numbering to continue where it left off, and would instead like to restart numbering, select the option Restart numbering with and enter the start value.
7. Close the dialog box with OK.

Note FlexPro numbers the labels and text boxes declared as headings in the document in the order in which you have inserted them. The position on the page is ignored in this case. Therefore, to place a heading in front of another heading, you cannot simply place this one above the other. You would need to send it further to the back.

Adding a Table of Contents or Table of Figures

1. Add a blank page at the location in the document where the table index will be added.
1. Use the command Document Design[Document Layout] > Insert Document Element > Table of Contents or Document Design[Document Layout] > Insert Document Element > Table of Figures.
2. For a table of figures, in the Insert Table of Figures dialog box select the desired caption category and click OK to close the dialog box.
3. Double-click on the table index to change the title and style. You can also change the font and font size of the individual outline levels using the ribbon after you have selected an element.

Note If the table index shown appears empty, this is because no headings or captions were found.

Navigating Documents

- Use the commands Document Design[Page] > Previous or Document Design[Page] > Next to go to the previous or next page of the document.
- To view the first or last page of the document, use the commands Document Design[Page] > Previous > First or Document Design[Page] > Next > Last.
- Choose the desired item from the input field under Document Design[Page] > Go To to enter and view any page number.
- You can use the mouse wheel to scroll through several pages of the document.

Resizing Graphics

- Under View[Zoom] > Zoom you can select a display size between 25 % and 200 %. The display size 100 % can be reached by clicking on View[Zoom] > 100 %.
- Alternatively, you can change the display size using the mouse by turning the mouse wheel while holding down the CTRL key.
- You can also set the display size is automatically so that the document remains completely visible in the window. To do this, use the command View[Zoom] > Fit to Window. With Document Design[Tools] > Zoom you can zoom in on any section. Simply drag a rectangle to enclose the section.

- A grid can be displayed in the background to make aligning objects easier. You can switch the grid on and off with Document Design[Drawing Tools] > Show Grid. You can set the grid spacing and the width of the margins under Document Tools/Design[Page Setup]. Change Margins, Numbering and Paper Size.

Displaying Multiple Adjacent Pages

1. Select View[Window] > New Window to open a second window of the same page.
2. Now click View[Tab Group] > New Vertical Tab Group to display both windows side by side.
3. Now you can navigate both windows independently using the commands Document Design[Page] > Next and Document Design[Page] > Previous.

Changing the Drawing Grid

1. Click on Document design [Drawing aids] > Grid Settings to open the Page Layout tab.
2. Under Grid Spacing specify the preferred horizontal and vertical grid spacing.

Activating or Deactivating the Automatic Snap to Grid Feature

Select Document Design[Drawing Tools] > Snap to Grid to activate or deactivate automatic alignment to the grid.

Note You can temporarily deactivate the automatic snap to grid feature by holding down the ALT key.

Linking and Embedding Presentation Objects

Inserting a link from a presentation object created in the project database into the document

1. Use your mouse to drag the object from the object list directly to the location in the document where you want it to appear.
2. Position the object and adjust its size.

3. If the object covers multiple pages and you do not want virtual blank pages 472 added to the document, then you need to select the document collection and in the Properties window set the Insert pages property to False.

Embedding a New Presentation Object into the Document

1. In the object list, select the data sets that you want displayed in the object.
2. Click in the window of the open document.
3. Select the desired object on the Insert tab of the ribbon
4. Position the object and adjust its size.
5. If the object covers multiple pages and you do not want virtual blank pages added to the document, then you need to select the document collection and in the Properties window set the Insert pages property to False.

Displaying a Data Set in a New Embedded Diagram

Drag the data set from the object list to the position in the document where the diagram is to be placed. Depending on the data structure of the data set, there are various display options from which you can choose. A new embedded diagram is created in which the data set is displayed as a curve.

Embedding a Copy of an Existing Presentation Object into the Document

1. While holding down the CTRL button, use your mouse to drag the object from the object list or from another document directly to the location in the document where you want it to appear.
2. Position the object and adjust its size.
3. If the object covers multiple pages and you do not want virtual blank pages added to the document, then you need to select the document collection and in the Properties window set the Insert pages property to False.

Move an Existing Presentation Objects into the Document

1. Use your mouse to drag the object from another document directly to the location in the document where you want it to appear. To switch from one document window to another document in the background, briefly point the mouse on its title to move it to the foreground.
2. Position the object and adjust its size.

3. If the object covers multiple pages and you do not want virtual blank pages added to the document, then you need to select the document collection and in the Properties window set the Insert pages property to False.

Note If you want to copy or move an object from a worksheet, you have to use the clipboard.

Linking a Document

1. Open the main document to which you want to add another document.
2. Switch to the page *before* the location where you want to add the document.
3. Drag the document with your mouse from the object list and drop into the open document.
4. Position the document so that the left top corner matches up with the left top corner of the main document.

Note FlexPro inserts a blank page into the document in which a document collection will be placed, which links the document that will be added. If the page format differs from the main document, a document section will be created automatically and the correct page layout will be set for the page with the document collection.

Linking to a Document Collection

To create a new document with a document collection:

- Select Insert[Containers] > Document > Collection and continue with step 4 below to set up the document collection.

To insert a document collection into an existing document:

1. Open the main document to which you want to add the document collection.
2. Place a blank page in the location in the document where the documents will be added.
3. Use the command under Document Design[Document Layout] > Insert Document Element > Document Collection to place a document collection on the blank page.

4. In the Document Collection Properties dialog box that appears, you can choose from these two alternatives: Search path with placeholders, Paths from data set and Data Query. The data query is then only available when indexing is enabled for the project database.
   For example, to search for documents with the name Document in the folders Measurement 1, Measurement 2 and so forth:

   a) Search path with placeholders

   1. Select Search path with placeholders.
   2. For the search path enter \Measurement*\Document.DOC. You can also click on the button to the right of the input box and select the document in the Measurement 1 folder. FlexPro automatically replaces the 1 in Measurement 1 with the placeholder *.

   b) Data Query

   1. Select Data Query is available.
   2. Double-click on the empty line under Object type is document.
   3. In the Edit Search Criteria dialog box, for the Search item: Parent folder name, for the Search criteria: starts with and then in the input box enter Measurement.
   4. Close the dialog box with OK.
5. Now click on Run search to test the search. FlexPro shows you the number of documents found.
6. Under Page Selection specify whether all pages or only a range of pages should be copied from the found documents.
7. Select the option Show document headers if the headers and footers of the found documents instead of those of the main document should be displayed.
8. Close the dialog box with OK.

Notes

- In the main document the number of pages required to display the found documents is inserted automatically. If you do not want virtual blank pages 472 added to the document, then you need to select the document collection and in the Properties window set the Insert pages property to False.

- If you place additional shapes on the blank page with the document collection, then they will appear on all pages of the embedded documents, as long as the value of the property Insert pages is True.
- When you update the main document, the search will be repeated, if necessary, and the number of pages may be adjusted.

Editing Presentation Object Links

1. Select the object link in the document that you want to edit.
2. Select Format[Edit] > Edit or press the F2 key.
3. In the Choose Object dialog box choose a presentation object to which you want the link to refer.

Converting a Presentation Object into a Drawing

1. Select the presentation object in the document that you want to convert into a drawing.
2. Select Format[Object] > Convert into Drawing.

Notes

- Text objects and media cannot be converted into drawings.
- During conversion, the object is replaced by a representation of it, which means the properties of the converted objects can no longer be changed, cursors can no longer be used and the object can no longer be updated.
- The shapes of the converted object are grouped in multiple layers. You may have to ungroup them before you can edit individual shapes.

Embedding Pictures

1. Click Document Design[Illustrations] > Pictures.
2. Choose a file in the Insert Picture dialog box.
3. Click OK to close the dialog box. The newly inserted image is now displayed.
4. If necessary, drag the image with your mouse to position it.
5. Set the height and width on Format[Measures].

Linking and Embedding OLE Objects

FlexPro can embed any graphics, documents, tables, etc., which originate from other applications, into documents. The application from which an object is inserted must have an OLE server interface. These objects are called OLE objects.

1. Select Document Design[Document Layout] > Insert Document Element > New Object and carry out one of the following actions:

Inserting an Existing File as an Object

1. Select Create from File if the object already exists as a file.
2. Enter the file name and path on the hard drive or click on Browse to browse for the file.
3. If you want to insert a link to the file, select Link. In this case, you can edit the linked object later using the OLE server application associated with that file type.

Inserting a New Object

1. Select Create New.
2. Select an object type.
2. Click on OK.

Editing an Embedded or Linked OLE Object

1. Select the object that you want to edit.
2. To open the object in a separate window, click Format[Object] > Open Object. To edit the object directly in FlexPro, press F2 or click Format[Edit] > Edit. The toolbars will then appear for the OLE server application assigned to the object.
3. To close an embedded editor window, press the ESC key or click on any area outside of the editor window.

Editing OLE Object Links

- Click Home[Edit] > Links.

Note In the Links dialog box, you can update individual links or switch to automatic or manual update. With Change Source you can refer a different file for the object link.

Labeling Diagrams, Tables and Pictures

1. Click on the diagram, table, picture or embedded OLE object that you want to label.
2. Now right-click with the mouse to open the context menu and select Insert Caption.
3. In the Edit Text dialog box that appears, enter the caption text without numbering it.
4. In the Level field specify the caption category, if applicable.
5. In the Format field you can adjust the numbering format. For example, you can set alphabetical or to Roman numeral numbering by changing the formatter of the field %<N>.
6. If you do not want the numbering to continue where it left off, and would instead like to restart numbering, select the option Restart numbering with and enter the start value.
7. Close the dialog box with OK.

Note: FlexPro numbers the text boxes declared as captions in the document in the order in which you have inserted them. The position on the page is ignored in this case. Therefore, to place a caption in front of another caption, you cannot simply place this one above the other. You would need to send it further to the back.

Working with Shapes and Other Objects

Creating a Header and Footer

To create a header and footer manually:

1. Select all of the objects that are be displayed on all pages of the document and not just on the current page. You can click on the objects while holding down the CTRL key, or you can use your mouse to drag a rectangle to select an area.
2. Click Format[Arrange] > Add to Header.

To create a header and footer semi-automatically:

1. Check the option Document Design[Header] > Header and Footer.
2. Click in the window of the open document.

Note You can use the options Document Design[Header] > Different First Page and Document Design[Header] > Different Odd & Even Pages to assign different headers and footers to the first page or pages with even and odd page numbers.

Rearranging Shapes and Other Objects on a Plane

When several shapes or other objects are superimposed over each other, you can bring objects to the front or send them to the back.

1. Select all of the objects whose arrangement you would like to change. You can click on the objects while holding down the CTRL key, or you can use your mouse to drag a rectangle to select an area.

To bring selected objects one step closer to the foreground or one step further to the background:

2. Select Format[Arrange] > Bring Forward or Send Backward.

To bring selected objects closer to the foreground or send them further to the background:

2. Click on the arrow next to Format[Arrange] > Bring Forward and in the menu select Move to the Foreground or click on the arrow next to Send Backward and in the menu select Move to the Background.

Note Objects in the document header that are displayed on all pages always appear behind those that are only displayed on the current page.

Aligning Shapes

You can align shapes relative to each other or to the page:

1. Hold down the CTRL key and click on the objects you want to select for alignment.
2. Click Format[Arrange] > Align and select the desired options from the menu.

Note If you align an object related to the page and later change the page format, you will have to re-align the object to account for the new page dimensions.

Rotating Shapes

1. Hold down the CTRL key and click on the objects you want to rotate.
2. Click Format[Arrange] > Align and select the desired options from the menu.

Note Presentation objects and OLE objects cannot be rotated. A presentation object can, however, be converted into a drawing, which can then be rotated.

Grouping Shapes

1. Select all of the objects that you want to group together. You can click on the objects while holding down the CTRL key, or you can use your mouse to drag a rectangle to select an area.
2. Click on Format[Arrange] > Group.

Ungrouping Shapes

1. Click on the group containing the objects that you want to separate.
2. Click Format[Arrange] > Group or on the arrow next to it and select Ungroup.

6.6 Shapes

You can make the optical design of your FlexPro diagram and table documents and the presentation objects more interesting by using shapes. Use the buttons under Design[Illustrations] toolbar to create lines, arrows, ellipses, rectangles, circles, arcs, polylines, polygons and labels. After drawing an object, you can give it a color or a pattern, change the line type and color of the object, or change, move, rotate or resize it. In the document you can also combine the objects drawn with graphics, which you have inserted into the document. You can copy objects and insert the objects into a different location in the document or into a different document or presentation object.

You can re-shape polylines and polygons at a later time. You can combine any freeform surfaces from arcs, lines and polylines and then fill them or give them a pattern.

Use the Label and Text Box shapes for shorter text and headings that you want to insert into a document or a diagram. The text box can optionally have a border around it and the text can be aligned horizontally or vertically in the rectangular box. Larger blocks of text that use different fonts and font sizes should be embedded into the document as text objects 459.

You can protect shapes, such as those in the header and footer of a document, from being edited accidentally.

When you insert shapes into a diagram or table, they are displayed not only in the current view of the presentation object, but also in any links you had inserted into a document or worksheet. If the size, i.e. width and height, of a link differs from that of the diagram itself, then these shapes will be transformed to the required target dimensions when displayed in the link. Use the Format[Arrange] > Show in This View Only command for shapes that should not be displayed in all views.

Working with Shapes

Drawing a Line, Rectangle, Ellipse, Arc or Other Shapes

1. Click Design[Illustrations] > Shapes and select, for instance, Polyline.
2. Click on the location where you want to start the polyline.
3. Now move your mouse to the location where you want the next corner point to be and click there.
4. To draw a freehand curve, move the mouse while holding down the left mouse button.
5. Double-click to stop drawing.

Notes

- You can influence the appearance of the various shapes if you keep the SHIFT key and/or the CTRL key pressed while you draw. By doing this, you can, for instance, change the direction of an arc, a circular arc or a square, or force horizontal or vertical lines to be created.
- You can cancel drawing an object by pressing ESC.
- Now draw arcs and ellipse segments with a 90° angle. Use the command Format[Tools] > Edit Shape You can then change the angle.

Adding Labels

1. Click Design[Illustrations] > Label or Design[Illustrations] > Shapes > Text Box.
2. Click on the location where you want to start the label, or use your mouse to drag a rectangle for the text box.

3. In the dialog box that appears, enter the text for the label and choose the font.
4. In the text you can embed fields for the page number, data, etc. by placing the insertion point at the desired location and then selecting a field from the Fields list box.
5. You can also add a field to display calculation results by clicking on Insert Field.
6. In the Writing direction field you can enter any angle between 0 and 360°.

Note: Use of the shapes Label and Text box is preferred for smaller labels and headings. If you require more freeform text design, you should use the text object. Labels cannot be as freely formatted as text. However, unlike the text object, you can change the writing direction.

Cutting Shapes

You can cut out shapes to combine the fragments into freeform surfaces later, for instance.

1. Click Design[Tools] > Cut or Format[Tools] > Cut.
2. Click on the location where you want to start the cutting edge.
3. Move the mouse to the position where you want to end of the cutting edge.
4. Click to cut.

Protecting Shapes

To protect shapes from accidental editing:

1. Select the shapes you want to protect.
2. Right-click with your mouse on the selection.
3. In the menu that now appears, point to Protect and from the submenu select Protect.

To remove the protection:

1. Select the shapes whose protection you want to remove.
2. Right-click with your mouse on the selection.

3. In the menu that now appears, point to Protect and from the submenu select Cancel Protection.

Displaying Shapes on Virtual Subsequent Pages

To display shapes on virtual subsequent pages of the current page:

1. Select the shapes you want to display on virtual subsequent pages.
2. Right-click with your mouse on the selection.
3. In the menu that appears, point to Display on Virtual Subsequent Pages and select Display from the submenu.

To cancel the display on virtual following pages:

1. Select the shapes you no longer want to be displayed on virtual subsequent pages.
2. Right-click with your mouse on the selection.
3. In the menu that now appears, point to Display on Virtual Subsequent Pages and select Cancel Display from the submenu.

Connecting Shapes to Freeform Surfaces

You can combine any freeform surfaces, e.g. rectangles with rounded corners or sections of circles from polylines, lines and arcs. You can then give these surfaces a fill pattern, for example.

1. Arrange several polylines, lines and arcs so that they form a closed border. Make sure that the ends are as flush as possible next to each other.
2. Select all of the objects that are to form the freeform surface.
3. Choose the command Format[Edit] > Connect. The selected objects are transformed into a closed polygon to which you can add a fill pattern, for instance.

Adding and Deleting Arrowheads

1. Click on the arc or the line to which you would like to add one or two arrowheads.
2. Click Format[Lines] > Arrows and select an arrow type.

Adding, Changing or Deleting a Fill Color or a Pattern

1. Select the objects with the fill color or pattern you would like to change. You can select multiple objects by pressing the CTRL key.
2. Click Format[Fill] > Style and select the desired pattern.
3. Click on Format[Fill] > Color to select a color. Click on No Fill to remove the fill.

Reusing Display Attributes

To use display attributes of an existing shape for new shapes:

1. Click on an empty area in the document or presentation object to make sure that no shapes are selected.
2. Click Design[Tools] or Format[Tools] > Pick Attributes and then on the shape with the attributes you want to use.
3. Now insert shapes that are to adopt these attributes.

To transfer display attributes from one shape to another:

1. Select all of the shapes with the drawing attributes that you want to change.
2. Click Design[Tools] > Pick Attributes and then on the shape with the attributes you want to use.

To transfer display attributes from one shape or some other element to another:

1. Select the element with the drawing attributes that you want to transfer to a different element.
2. Select Home[Clipboard] > Transfer Format.
3. Now click on the element that you want to receive the display attributes. To edit several elements within a rectangular range, drag your mouse to form a rectangle around them.
4. Deselect Home[Clipboard] > Transfer Format.

Changing Line Style and Line Width

1. Select the objects with the lines or border you would like to change.
2. Under Format[Lines] click on Width or Dashes and select the desired width or style from the menu.

Changing Line and Fill Transparency

1. Select all of the objects with the line and fill transparency you would like to change.
2. Change the setting under Format[Lines] > Transparency or Format[Fill] > Transparency.

Note The higher the transparency, the more background will show through.

Changing the Color of a Line, Arc, Label or Border

1. Select the objects with the line color you would like to change.
2. Click Document Design[Tools] > You can change the text color for labels under Format[Font Color] > Font.

Inserting Symbols and Formatting Labels

In all text that you can place in tables and diagrams, as well as in labels, which you insert into documents, you can use a series of control sequences to insert Greek symbols into the text or to format individual text passages. The control sequences always consist of a \ character, followed by a letter. The upper case letter activates the corresponding attribute, and the same letter in lower case deactivates it. You can combine the attributes any way you like.

Control Sequences	Meaning
\S ... \s	Selects the symbol character set for displaying the included text.
\U ... \u	Displays the included text as superscript.
\L ... \l	Displays the included text as subscript.
\I ... \i	Displays the included text as italic.
\B ... \b	Displays the included text as bold.
\CRRGGBB ... \c	Displays the included text in the specified color. Enter the color as a hexadecimal RGB value. For example, \CFF0000Red\c outputs "Red" as 100 % red.
\FRRGGBB ... \F	Outputs the enclosed text with the specified background color.
\N	Cancels all of the above conditions and restores the normal appearance of the text.

Control Sequences	Meaning
\\	Displays a \ character. The character has to be entered twice so that it is not interpreted as a control sequence.

Hint You can use the Windows Character Map program to select the Greek symbols.

6.7 Control Panel and Controls

FlexPro offers you a range of controls, such as text fields, buttons, or list boxes, that you can use to create forms or dialog boxes. You can insert these controls, like shapes, either directly into documents, charts, or tables, or you can create a panel on which you position the controls. You can embed the control panel (similar to a diagram, for example) in a worksheet or document, or create it directly in the object list and then use it as a link in a worksheet or document, if necessary.

The controls that allow input can also be linked directly to a data set, in which FlexPro enters the user's current input. You can therefore include this input in your analyses.

In addition, FlexPro can automatically send an update command as soon as the user makes an entry. For example, this allows you to parameterize your analysis using a control panel and to update it automatically after making any change.

A more flexible control option is offered by VBA macro procedures. For each control, you can store a VBA macro to be called when the user modifies the control. This allows evaluations to be fully automated.

Note: To edit controls and access their properties, you must first activate the design mode of the respective object in which the control is located (see Developer[Controls] > Design Mode or Control Panel Tools\Design[Controls] > Design Mode).

Here is an overview of the available controls:

Control	Description
Button	A labeled area that the user can click to trigger an action
Combo box	A box with a drop-down list from which the user can select an item.
Checkbox	A labeled box that the user can select.

Spin box	Two buttons with arrows that can be used to increase or decrease a value.
List box	A list of several lines of text from which the user can select one or more.
Option button	Usually several labeled radio buttons are used in a group, allowing the user to select one of several options.
Group box	A labeled frame that can be used to group radio buttons, for example.
Designation	A description with which, for example, a text box can be labeled.
Scroll bar	A vertical or horizontal bar with a slider and two arrows at the ends.
Text box	A field where the user can enter any text.

Control panels can always be added with the command Insert[Other] > Control Panel. You can insert controls into control panels using Control Panel Tools\Design[Controls] > Controls and into presentation objects using the command Developer[Controls] > Controls.

* This object is not available in FlexPro View and FlexPro Basic.

6.8 Worksheet

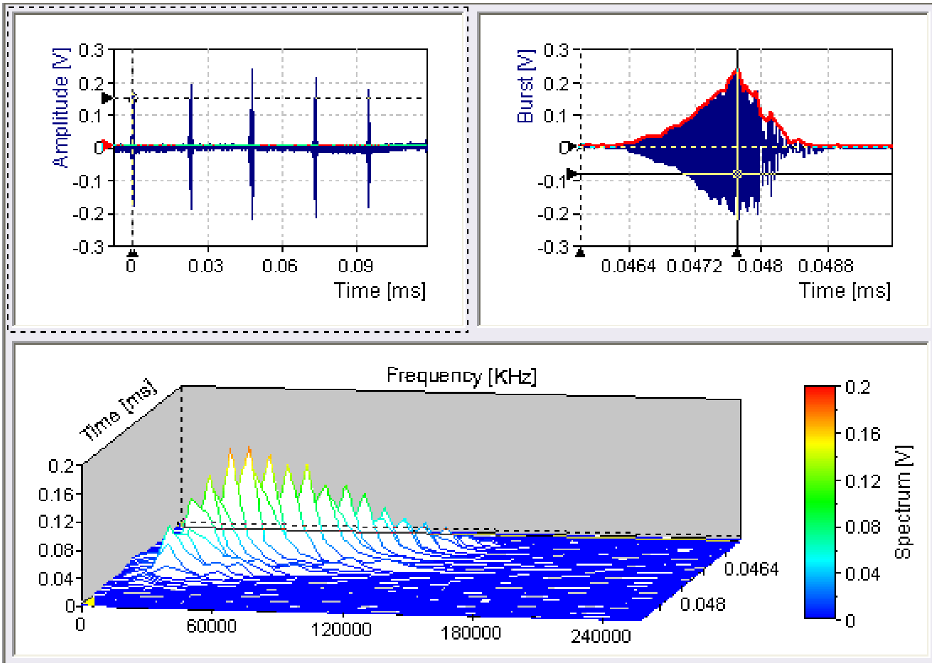
In the worksheet you can arrange presentation objects, i.e. diagrams, text, tables and media, and you can set the number of panes. The curves in the diagrams are measured using cursors. As in a document, you can also use linked and embedded objects in a worksheet.

The cursor operating options are completely identical in a worksheet and in a document. Unlike a document, which is oriented to paper output, a worksheet is the ideal environment for working on screen. The panes in the worksheet are connected seamlessly to each other and make efficient use of the screen. The size of the worksheet is not oriented to a page layout and can therefore take up the whole screen. If you display several panes in a worksheet, you can temporarily maximize any one of them. This pane then has the complete window at its disposal so that you can see the contents better.

If you work with object linking, you can use documents and worksheets together. Set up the diagrams to be edited in the project database and insert links to them in a document and a worksheet. Now use the worksheet to move the cursors and the document to print the diagrams.

If you add multi-page diagrams to the worksheet, then they are displayed completely on one page. For multi-page column tables, only the first page is displayed.

The following illustration shows a typical worksheet:



Working with Worksheets

Creating a Worksheet

1. In the Folders window, select the folder where you want the new worksheet to be placed.
2. Select the data sets and/or diagrams that are to be displayed in the worksheet in the object list. If you do not select anything, an empty worksheet will be created into which you can insert diagrams later.

3. Click Insert[Containers] > Worksheet and select the desired window arrangement. You will need to provide a plane for each object selected in the object list.

Creating a Worksheet for a Data Set with a Broad Overview

Do the following to create a worksheet with two panes, where the upper pane displays an entire data set and the lower pane always displays the enlarged section of the data set located between the two cursors in the upper pane:

1. In the Folders window, select the folder where you want the new worksheet to be placed.
2. Select one or more data sets that are to be displayed in the worksheet in the object list.
3. Click Insert[Containers] > Worksheet and use the magnifier to select the two-pane layout.
4. In the new worksheet, move the cursors in the upper pane to display a section enlarged in the lower pane.

Linking and Embedding Presentation Objects

Inserting a Link to a Presentation Object Set Up in the Project Database

Drag the object with the mouse from the object list directly to the pane where it should appear.

Embedding a New Presentation Object into the Worksheet

1. In the object list, select the data sets that you want displayed in the object.
2. Click on the empty pane in the worksheet where the new object is to be added.
3. Select an object from one of the galleries using the Paste tab of the ribbon.

Displaying a Data Set or Multiple Data Sets in a New Embedded Diagram

Simply drag the data sets from the object list into an empty worksheet pane. Depending on the data structure of the data sets, there are various display options from which you can choose. A new embedded diagram is created in which the data sets are displayed as curves.

Embedding a Copy of an Existing Presentation Object into the Worksheet

While holding down the CTRL key, use your mouse to drag the object from the object list or a document directly onto the pane where it is to appear.

Moving an Existing Presentation Object into the Worksheet

While holding down the SHIFT key, use your mouse to drag the object from the object list or a document directly onto the pane where it is to appear.

Note: If you want to copy or move an object from a different worksheet, you have to use the clipboard.

Adding or Removing Panes

To add a pane via the ribbon:

- Click Worksheet Design[Pane] > Add.

To remove a pane via the ribbon:

- Select the pane and then click Worksheet Design[Pane] > Delete.

To split a pane horizontally using the mouse:

1. Move the mouse to the upper right corner of the pane until the mouse arrow becomes a horizontal split arrow.
2. Now drag the split bar downward to the desired position.

To split a pane vertically using the mouse:

1. Move the mouse to the lower left corner of the pane until the mouse arrow becomes a vertical split arrow.
2. Now drag the split bar to the right to the desired position.

To remove a pane using the mouse:

- Move a horizontal or vertical split bar by dragging the mouse so that the pane to be removed disappears.

Changing the Window Layout

To change the number and arrangement of panes in a worksheet, click Worksheet Design[Pane] > Choose Layout and select the desired layout from the menu.

Note: You can also use your mouse to change the window layout.

Maximizing a Pane

If the worksheet contains several panes, you can display any pane temporarily as maximized by clicking on it and then selecting Worksheet Design[Pane] > Maximize or Cursors[Pane] > Maximize.

6.9 Exporting Presentation Objects, Documents and Worksheets

Exporting Presentation Objects, Documents and Worksheets as a File

With FlexPro you can export your diagrams, texts, tables, documents and worksheets in a variety of different file formats. The following table provides a summary of options:

Format	Description of Format	Supported Objects
Rich Text (.rtf)*	Text with formatting	Text
PDF (*.pdf)	Document exchange format	2D diagram, 3D diagram, text, column table, cell table, document, worksheet
Windows Enhanced Metafile (.emf)	Enhanced vector graphics	2D diagram, 3D diagram, text, column table, cell table, document, worksheet
Bitmap Picture (.bmp)	Bit-mapped image	2D diagram, 3D diagram, text, column table, cell table, medium, document, worksheet
JPEG Picture (.jpg)	Compressed image	2D diagram, 3D diagram, text, column table, cell table, medium, document, worksheet
PNG Picture (.png)	Compressed image	2D diagram, 3D diagram, text, column table, cell table, medium, document, worksheet

Format	Description of Format	Supported Objects
HTML Document (.htm)	File format of the World Wide Web. Can be published on the Internet/intranet and is displayed by web browsers.	All objects, including complete project databases.
Text (.txt)*	Text in ANSI code	Text, column table, cell table

The formats identified by * allow you to store several objects in one single file.

1. Select all of the objects in the object list that you want to export to the file. Alternatively, you can also select an object for export that is in a worksheet or document.
2. Choose the command Data[Export] > Export.
3. Select the target folder and specify the file name for the file.
4. Under File Type select the format for the file.
5. Select any export options for the file.

Note If you export a multi-page object in a data format that can only accommodate one page, then a file is created for each page.

Exporting Presentation Objects, Documents and Worksheets with OLE

You can export presentation objects, documents, worksheets and text as OLE objects and thereby embed copies or links to these objects into OLE client applications. FlexPro offers the objects as FlexPro Project Database Objects, Pictures (Enhanced Metafiles) and Device Independent Bitmaps. Embedding or linking depends on the third-party application into which the file is to be inserted. Here we describe the process using the popular word processor Microsoft Word.

To create a dynamic link:

1. Select a presentation object, document or worksheet from the object list.
2. Use the command Home[Clipboard] > Copy to copy the object to the clipboard.
3. Place the cursor in the location in the Word document where the object is to be inserted.

4. Use the Word command Home[Clipboard] > Paste > Paste Special.
5. In the Paste Special dialog box, select the FlexPro Project Database Object format and select the option Paste link.

Note You can also copy an open presentation object to the clipboard by using the command Copy in the window where the object is displayed. However, elements in the object must not be selected.

To insert a static copy:

1. Select a presentation object, document or worksheet from the Object List.
2. Use the command Home[Clipboard] > Copy to copy the object to the clipboard.
3. Place the cursor in the location in the Word document where the object is to be inserted.
4. Use the Word command Home[Clipboard] > Paste > Paste Special.
5. In the Paste Special dialog box, select the Device Independent Bitmap format.

Note You can only use the Project Database Object format for links and not for inserting.

6.10 Working with Presentation Templates

Creating a Presentation or Document Template

1. In the object list select the diagram, table, text or document that you want to save as a template.
2. Now click Home[Selected Objects] > Save as Template.
3. Follow the wizard's next steps and use Help in the wizard if you require additional assistance.

See Also

Presentation and Document Templates 378

Organizing Presentation and Document Templates 500

Template Databases 49

Using a Presentation or Document Template

1. Select the object from the object list that you would like to present.
2. On the Paste of the ribbon tab, open one of the following menus Diagram, Table, Text or Document and select From Template.
3. Follow the steps of the respective Wizard and use the wizard's Help if you require additional assistance.

See Also

Presentation and Document Templates 378

Creating a Presentation or Document Template 499

Template Databases 49

Organizing Presentation and Document Templates

Use the Organizer dialog box to delete presentation or document templates or to copy templates from one data base to another.

Opening the Organizer dialog box and selecting the template type

1. Click on File > Info > Organizer.
2. In the Organizer dialog box, click on the Templates tab.
3. Document, Diagram, Cell Table, Column Table, Cell table or Text.

Opening the Current Project Database or a Template Database for Organizing

- Choose a template database in one of the Items available in list boxes.

Opening a shared template database or any other database for organization

1. Click on one of the Close Project Database buttons to close the database currently displayed in the list.
2. Click on the Open Project Database button and select a project database on your hard drive or from the network.

Copying templates from one project database to another

1. Select the templates you would like to copy.

2. Click on Copy.

Deleting templates from a project database

1. Select the template that you would like to delete.
2. Click on Delete.

6.11 Hyperlink

FlexPro allows you to link objects using hyperlinks. A hyperlink is displayed as a highlighted section of text. If you click on the hyperlink, the event occurs that is associated with that hyperlink:

Type of Hyperlink	Action
Hyperlink to a FlexPro object in the project database	The object is opened.
Hyperlink to a file on the disk or on the network	The file is opened by the program associated with that file type.
Hyperlink to a Website	The page is displayed in the web browser.
A hyperlink that sends an e-mail	A blank e-mail to the specified address is provided.

Hyperlinks can be inserted anywhere that you can have labels, such as the axis label for a diagram, the title of a table or in a text field in a document. Currently, hyperlinks cannot be used in text objects. When exporting to HTML, these hyperlinks are converted into HTML and remain active.

In addition to the hyperlinks in labels, you can specify a hyperlink in the General tab of the Properties dialog box of any FlexPro object. This hyperlink is generally used as a reference to documentation of the object. When exporting to HTML format, these hyperlinks are incorporated into the heading of the HTML page.

Working with Hyperlinks

Inserting a Hyperlink into a Document

1. Click Document Design[Illustrations] > Label.
2. Click on the location where text with a hyperlink is to be inserted.
3. In the dialog box that appears, place the cursor in the edit box.

4. You can now enter a text and select the section to be highlighted as a hyperlink. Alternatively, you can leave the edit box blank. The selected text should not contain any fields.
5. Click on Insert or Edit a Hyperlink. A dialog box appears where you can enter the hyperlink.
6. Select one of the four hyperlink buttons, depending on the type of hyperlink you want, and enter the address.
7. The hyperlink has now been entered into the Edit Text dialog box.
8. The hyperlink consists of two tags: \<A href="*Address*"> and \</A>. The text to be highlighted in the document is located between these two tags. You are free to change this text.

Adding a Hyperlink to a Diagram or Table

1. Double-click on the text element to which you want to add a hyperlink, such as to an axis label of a diagram or the cell of a cell table.
2. In the Properties dialog box that appears, select the passage in the Text field that you want to highlight as a hyperlink. Alternatively, you can place the cursor at the point where you want to insert the hyperlink. The selected text should not contain any fields.
3. Click on Insert or Edit a Hyperlink. A dialog box appears where you can enter the hyperlink.
4. Select one of the four hyperlink buttons, depending on the type of hyperlink you want, and enter the address.
5. Close the Edit Hyperlink dialog box by pressing ENTER. The hyperlink has now been entered into the Properties dialog box.
6. The hyperlink consists of two tags: \<A href="*Address*"> and \</A>. The text to be highlighted in the diagram is located between these two tags. You are free to change this text.

Assigning a Hyperlink to a FlexPro Object

1. Right-click with your mouse on the object in the object list and select Properties.
2. Open the General tab of the Properties dialog box.
3. Depending on the hyperlink you want, click on one of the four buttons next to the Hyperlink input field and enter the address.

4. Close the Edit Hyperlink dialog box by pressing ENTER. The hyperlink has now been entered into the Hyperlink input field.

Following a Hyperlink

- To follow a hyperlink in a label, such as the axis label of a diagram, simply click on the highlighted text while holding down the CTRL key. For a read-only object, pressing the CTRL key is optional.
- To follow a hyperlink that you created in the General tab of a FlexPro object Properties dialog box, click Home[Object] > Follow Hyperlink. Alternatively, you can right-click on the object in the object list and select Follow Hyperlink from the context menu.

6.12 Formatters

Formatters control the output of numbers, angles, strings, calendar time values and timespans. They always start with a '%' character, which is followed by a code, such as *%5.3g*.

Standard Formatters

The standard formatters *%?* and *%&?* output any data types in a standard format. Floating point values are output using a number of valid places that can be set on the General tab in the Properties dialog box of the FlexPro project database. If you use a variant with the & character, data with a unit is output with the unit appended.

Numerical Value

This formatter controls how numerical data is displayed. You can choose from a variety of formatting options. You can align the digits in a field, specify whether an exponent should be displayed and if so, in which format and basis, specify the number of places after the decimal point, and much more.

Syntax

%[*Attribute*][*Field size*][**.***Places*]*Formatter*

The syntax of the numerical value formatter consists of the following elements:

Element	Description
Attribute	One or more of the following attribute characters:

Element	Description	
	0	Fills up the field size with leading zeros.
	&	Outputs quantities with an appended unit.
	#	Outputs the decimal separator even if no fractional digits are output.
	+ or space	+ also outputs the positive polarity sign. A space outputs a space instead of +.
	-	Places the formatted number aligned to the left and fills the remaining field width with spaces.
	'	Outputs and groups the integral part of the number using the 1000th separator for easier legibility.
	[*Invalid*]	Replaces the question mark usually output for void values with the text specified in square brackets.
Field size	Specifies with size (width) of the field into which the formatted number is to be placed as either left or right aligned. If necessary, it is filled with spaces. The entire number is always output, even if the *Field size* was not specified or a smaller size was specified. If applicable, the appended unit is ignored for the alignment and is instead placed after the aligned numerical value.	
Places	Depending on the formatter chosen, this selects the number of valid places or fractional digits. If you omit the element, the value preset on the General tab of the project database Properties dialog box is used.	
Formatter	f	Never includes the exponent.
	E, e *	Always includes the mantissa and exponent.
	G, g *	An exponent is only used if it leads to a more compact presentation.
	N, n *	An exponent to the base e is always output.
	B, b *	An exponent to base 2 is always output.
	M, m *	If the mantissa equals 1, then only the exponent is output without the mantissa. Otherwise, only the mantissa is output

Element	Description	
		without the exponent. This display is suited for scaling logarithmic axes, for instance.
	F	Instead of an exponent, an SI prefix is appended, e.g. 1.234 k instead of 1.234 E+003.
	x, X	Output as integer in hexadecimal system with lower case (x) or upper case (X) letters.
	d	Output as integer in dual system.
	i	Output as integer in decimal system. Use this formatter instead of %f to output large 64-bit integer values in full resolution.
	10^3 or 2^3. Otherwise, E+003 or B+003 will be output, for instance.	

Indexed Text

This formatter offers you the option of assigning texts to numerical values starting at 0, which are then displayed instead of the numbers. You can thus label the columns of a column chart, for instance, with individual text. Do not confuse this formatter with the string formatter, which cannot display any numerical data.

Syntax

%("$Text_0$**"[, "**$Text_1$**"[, ..., "**$Text_n$**"]])**

The syntax of the formatter for indexed text consists of the following elements:

Element	Description
$Text_0$ ... $Text_n$	The text to be output instead of the values 0 through n.

Calendar Time

FlexPro saves calendar times (date and time) as numerical values that contain the number of seconds since January 1, 1970 in "Universal Time Coordinated" (UTC). Using a calendar time formatter, you can display these numerical values as a calendar time. FlexPro converts the UTC values automatically to the time zone set in Windows Regional and Language Options. You can, however, specify an option that is to result in output in universal time coordinated. The formatter specifies which time value elements should be displayed. There is a corresponding formatting element for each of the elements listed below. This formatting element starts with a

% character. You can write any text between the individual formatting elements, where a percent sign has to be entered as %%.

Syntax

%[.*Fractional digits*]**(**[*UTC*] *Date***)**

The syntax of the calendar time formatter consists of the following elements:

Element	Description
Fractional digits	Selects the maximum number of fractional digits for seconds. If you omit the element, then the default value used is 6, which is equivalent of a 1µs resolution.
UTC	Outputs the date as Universal Time Coordinated (UTC).
Date	Any text in which the following element formatters can be embedded: Name of the day of the week, abbreviated — %a Name of the day of the week — %A Name of the month, abbreviated — %b Name of the month — %B Date and time, based on Windows Region and Language Options settings — %c Day of the month in decimal format (00 - 31) — %d Hour in 24-hour format (00 - 23) — %H Hour in 12-hour format (01 - 12) — %I Day of the year as a decimal number (001 - 366) — %j Month as a decimal number (01 - 12) — %m Minute as a decimal number (00 - 59) — %M A.M./P.M. indicator, based on Windows Region and Language Options settings — %p A.M./P.M. indicator — %P Second as a decimal number (00 - 59) — %S Calendar week, starting with Sunday, as a decimal number (00 - 53) — %U

Element	Description	
	Calendar week, starting with Monday, as a decimal number (00 - 53)	%W
	Day of the week as a decimal number (0 - 6; Sunday is 0)	%w
	Date, based on Windows Region and Language Options settings	%x
	Time, based on Windows Region and Language Options settings	%X
	Year, two-digit, as a decimal number (00 - 99)	%y
	Year, four-digit, as a decimal number (00 - 99)	%Y
	Name of the time zone, abbreviated	%z
	Name of the time zone	%Z
	Percent sign	%%

Examples

%(%A, the %th of %B %Y %H:%M)	An example of this output would be: "Tuesday, the 16th of February 2010 17:15".
%(UTC%A, the %th of %B %\	An example of this output would be: "Tuesday, the 16th of February 2010 04:15:00 PM"
%.0(%H:%M:%S) or %.0(%X)	Outputs only the time, so the seconds are output without fractional digits.

Time Span

Timespans can be formatted as *H:M:S* or *D:H:M:S*. The number of fractional digits as well as the number of places for the hour or day can be adjusted.

Syntax

%[Places][**.***Fractional digits*]*Formatter*

The syntax of the timespan formatter consists of the following elements:

Element	Description
Places	Specifies the number of digits for the day or hours. If necessary, it is filled with leading zeros.
Fractional digits	Selects the maximum number of fractional digits for seconds. If you omit the element, then the default value used is 6, which is equivalent of a 1µs resolution.
Formatter	t Output as H:M:S. T Output as D:H:M:S.

Examples

%3.2t	Formats the timespan as *HHH:MM:SS.ss*.
%.0T	Formats the timespan as *D:HH:MM:SS*. The seconds are output without fractional digits.

Angle

Angles can be displayed in the form of degrees, minutes, seconds (° ' '') or as a fraction of π. Angle data present in degrees or radians can be formatted.

Syntax

%[**.***Places*]*Formatter*

or

%[**.***Denominator*]*Formatter*

The syntax of the angle formatter consists of the following elements:

Element	Description
Places	When outputting in degrees, minutes, seconds, the maximum number of digits is specified for the seconds. If you omit the element, the default value 6 is used.
Denominator	Outputting as a fraction of π specifies the denominator of the fraction. If you omit the element, then instead of outputting a fraction, a decimal number with the maximum number of fractional digits predefined on the General tab of the project database Properties is output.
Formatter	r, R Outputs degrees, minutes, seconds. p, P Outputs a fraction of π.

Element	Description
	Use upper case letters if the data to be formatted is in radians (0 to 2 π) and lower case letters if the data is in degrees (0 to 360).

Examples

%.2r	An example of this output would be: "15° 12´ 24,34´´".
%.4P	An example of this output would be: "3/4 π".
%P	An example of this output would be: "0.75 π".

String

You can limit strings to a maximum length or fill them up to a minimum length. The text can be displayed as left or right justified.

Syntax

%[-][*MinFieldSize*][**.***MaxFieldSize*]**s**

or

%[-][*MinFieldSize*][**.***MaxFieldSize*]**S**

The syntax of the string formatter consists of the following elements:

Element	Description	
-	If specified, the field alignment is left-justified; otherwise, it is right-justified.	
MinFieldSize	Specifies the minimum field size in characters. The value is only relevant if the string is shorter than the specified value. It is then filled with spaces.	
MaxFieldSize	Specifies the minimum field size in characters. The value is only relevant if the string is longer than the specified value. Excess characters are then cut off.	
Formatter	S	Output with double '\' character. This prevents the '\' character in the text from being interpreted as the start symbol in control sequences, such as '\B' for bold.
	s	Output unchanged. Control sequences are interpreted and taken into account in the

Element	Description
	text formatting.

Examples

%s	An example of this output would be: "Test".
%.3s	An example of this output would be: "Tes".
%6s	An example of this output would be: " Test".

Special Format

The following special formats are suitable in particular for numbering in tables and for chapter titles in documents.

Syntax

%Formatters

The syntax of the special format formatter consists of the following elements:

Element	Description	
Formatters	a, A	Alphabetical numbering with upper or lower case letters.
	o, O	Roman numeral with upper or lower case letters.
	c, C	Chinese number, normal or accounting.
	j	Japanese Number.

Examples

%a	1, 2, 3, 4 is formatted as a, b, c, d.
%A	1, 2, 3, 4 is formatted as A, B, C, D.
%o	1, 2, 3, 4 is formatted as i, ii, iii, iv.
%O	1, 2, 3, 4 is formatted as I, II, III, IV.
%c	1, 2, 3, 4 is formatted as 一, 二, 三, 四.
%C	1, 2, 3, 4 is formatted as 壹, 贰, 叁, 肆.

%j	1, 2, 3, 4 is formatted as 一, 二, 三, 四.

6.13 Field and Placeholder

A field consists of an FPScript expression that supplies a value or string to be displayed and a formatter that determines how the data is to be formatted for output.

You can insert fields in any text, for instance, in axis labels, text objects, document labels, etc. Values to be displayed can be calculation results, header information such as a data set name and physical unit, or the current date and time.

The following field shows, for instance, the first value in the X component of a signal with three fractional digits:

```
%.3f{Signal.x[0]}
```

In this example, `Signal.x[0]` is the FPScript expression that supplies the value, and `%.3f{...}` is the formatter that determines that the number is to be formatted with three fractional digits instead of an exponent. Formatting information is optional here; the simple formatter `%{...}` outputs any data types in a standard format. Floating point values are output using a number of valid places that can be set on the General tab in the Properties dialog box of the FlexPro project database.

Fields always start with a % sign. Therefore, if you want to insert a single percent sign into a text, you need to type in the percent sign twice to prevent it from being interpreted by the formatter. For instance:

```
The probability of error is %{Probability}%%.
```

Accessing Object Attributes in Fields

Since FPScript has full access to FlexPro's Automation object model, you can display any object attributes, also known as properties. The pre-defined fields for axis labeling in diagrams is used frequently here, for instance, to determine the names, comments and units of the data sets to be plotted over an axis. The FPScript fields used for this can be somewhat long, which is why FlexPro uses Placeholders. Placeholders also have a formatter, which is followed by a name instead of FPScript code. The standard text for labeling the Y axis of a 2D curve, for instance, looks like this:

%<NameOrQuantityOrComments> %<[Unit]>

FlexPro replaces both placeholders when text is created using the following FPScript fields:

```
%{.Data.YValueObject(%<ListElement>).NameOrQuantityOrComments(.Data.YComponent)}
%{ThisFPObject.YAxes(.YAxis).Scaling.CurrentUnitInBrackets(%<YAxisElement>)}
```

FPScript defines a standard object, which can be activated using the optional keyword ThisObject. The expression .Data is thus equivalent to the expression ThisObject.Data. When evaluating the axis label of a curve, ThisObject is a reference to the curve. With ThisFPObject you have access to the FlexPro object, such as the diagram or table containing the field.

If the curve is assigned a list with several data sets, the data sets are evaluated for each element in the list. %<ListElement> is a placeholder that is replaced before the evaluation by the index of the list element.

The following table specifies which objects ThisObject and ThisFPObject reference:

Field contained in	ThisObject points to	ThisFPObject points to
Axis labeling of an axis	Axis	Diagram
Axis labeling, curve labeling or legend entry of a curve	Curve	Diagram
Legend title	Legend	Diagram
Color legend label	Color Legend	3D diagram
Column table title	Column table	Column table
Column title in a column table	Table column	Column table
Cell table title	Cell table	Cell table
Cell in a cell table	Cell	Cell table
Text object	Text object	Text object
Label in a document/diagram	Label	Document/diagram

Placeholders for FPScript Fields

The following list contains die placeholders that you can embed in labels using the drop-down menu.

Most placeholders access object attributes. The way some placeholders work depends on the type of label in which the placeholder is used. The %<Comments> placeholder, for example in the title of a table column, references the comments of a data set displayed in the column. If on the other hand it is used in a diagram title, it references the diagram comments.

Description	Placeholder	Purpose
FPScript Expression	%{ ... }	Adds a field in which you can embed any FPScript code.
Parameters	%{ ... .Parameters("*Name*")}	The value of the *Name* parameter of the object that contains the placeholder, or the value of the *Name* parameter of the data object to which the placeholder refers.
Project database name	%<DatabaseName>	The name of the project database.
Project database path	%<DatabasePath>	The name of the project database, including its path on the hard disk.
Path	%<Path>	The name of the object that contains the placeholder, including its path in the project database.
Name	%<Name>	The name of the object that contains the placeholder, or the name of the data object to which the placeholder refers.
Element name	%<ElementName>	The name of the list element or of the component of the data object to which the placeholder refers.
Name or comments	%<NameOrComments>	For the Y component, this is the name, and for the remaining components, this is the comments of the component of the data object to which the placeholder refers.
Name or quantity or comments	%<NameOrQuantityOrComments>	For the Y component, this is the name, and for the remaining components, if present, this is the physical quantity; otherwise, this is the comments of the component of the data object to which the placeholder refers.
Comments	%<Comments>	The comments of the object that contains the placeholder, or the Y comments of the data object to which the placeholder refers.
Comments or name	%<CommentsOrName>	Depending on the availability, this is the comments or name of the object containing the placeholder, or the comments or name of the

Description	Placeholder	Purpose
		component of the data object to which the placeholder refers.
Quantity, comments or name	%<QuantityOrCommentsOrName>	Depending on the availability, this is the physical quantity, the comments or name of the component of the data object to which the placeholder refers.
Unit	%<Unit>	The physical unit of the component of the data object to which the placeholder refers.
[Unit]	%<[Unit]>	The physical unit, in square brackets, of the component of the data object to which the placeholder refers. If the unit is empty, no brackets are output.
(Unit)	%<(Unit)>	The physical unit, in parentheses, of the component of the data object to which the placeholder refers. If the unit is empty, no brackets are output.
Quantity	%<Quantity>	The physical quantity of the component of the data object to which the placeholder refers.
Author	%<Author>	The author of the data object to which the placeholder refers.
Creation date	%<CreationDate>	The creation date of the object that contains the placeholder, or the creation date of the data object to which the placeholder refers.
Creation time	%<CreationTime>	The creation time of the object that contains the placeholder, or the creation time of the data object to which the placeholder refers.
Modification date	%<ModificationDate>	The last modification date of the object containing the placeholder, or the last modification date of the data object to which the placeholder refers.
Modification time	%<ModificationTime>	The last modification time of the object containing the placeholder, or the last modification time of the data object to which the placeholder refers.
Origin	%<Origin>	The origin of the data object to which the placeholder refers.

Description	Placeholder	Purpose
Z value	%<ZValue>	When displaying a signal series, this is the Z value assigned to the signal of the signal series displayed as a curve.
Z comments	%<ZComments>	When displaying a signal series, this is the comments of the Z component of the data object to which the placeholder refers.
Z unit	%<ZUnit>	When displaying a signal series, this is the physical unit of the Z component of the data object to which the placeholder refers.
Z unit	%<(ZUnit)>	When displaying a signal series, this is the physical unit, in square brackets, of the Z component of the data object to which the placeholder refers. If the unit is empty, no brackets are output.
(Z unit)	%<(ZUnit)>	When displaying a signal series, this is the physical unit, in parentheses, of the Z component of the data object to which the placeholder refers. If the unit is empty, no brackets are output.
Date	%<Date>	The date on which the placeholder was last updated.
Time	%<Time>	The time at which the placeholder was last updated.
Folder name	%<FolderName>	The name of the folder in the project database with the object that contains the placeholder.
Folder comments	%<FolderComments>	The comments of the folder in the project database with the object that contains the placeholder.
Active subfolder name	%<ActiveFolderName>	The name of the active subfolder within the folder in the project database with the object that contains the placeholder.
Active subfolder comments	%<ActiveFolderComments>	The comments of the active subfolder within the folder in the project database with the object that contains the placeholder.
User	%<User>	The name of the user who is currently logged in on Windows.
Document path	%<DocumentPath>	The path in the project database of the container object in which the object containing the

Description	Placeholder	Purpose
		placeholder is found. In the case of a diagram embedded in a document or worksheet, this is the path of the document or worksheet; otherwise, it is the path of the folder that contains the diagram.
Document name	%<DocumentName>	The name of the container object in which the object containing the placeholder is found. In the case of a diagram embedded in a document or worksheet, this is the name of the document or worksheet; otherwise, it is the name of the folder that contains the diagram.

Integrated placeholders

FlexPro also uses some placeholders that do not store FPScript code:

Placeholder	Description	Used in	Purpose
%<DataSet>	Data set index	Axis label, curve label and legend entry for 3D curve	Represents the number of the respective data series in the curve for which an axis label, curve label or legend entry is created and is used in FPScript code as an index for the Curve3DDataSets collection of the Curve3D object.
%<Index>	Data series index	Axis label, curve label and legend entry for 2D or 3D curve. Marker on 2D or 3D curve and coordinates at cursor.	Represents the number of respective data series in a 3D data set for which a label is created. Used in the text to display the data series index or in the FPScript code as an index variable.
%<[Index]>	Data series index, in brackets	"	Like %<Index>, but the output result is in brackets. This placeholder cannot be used in the FPScript code.
%<ListElement>	List element index	Axis label, curve label and legend entry for	Represents the number of respective list elements in the curve for which an axis label,

Placeholder	Description	Used in	Purpose
		2D or 3D curve and column title of a column table.	curve label or legend entry is created. Used in the FPScript code as an argument for the ValueObject property.
%<YAxisElement>	%<YAxisElement>	Axis label, curve label and legend entry for 2D curve	Represents the number of the scale of the Y axis over which the curve is displayed for which an axis label, curve label or legend entry is created. Used in the FPScript code as an argument for the CurrentUnit property.
%<XAxisElement>	%<XAxisElement>	"	Represents the number of the scale of the X axis over which the curve is displayed for which an axis label, curve label or legend entry is created. Used in the FPScript code as an argument for the CurrentUnit property.
%<YName> %<XName> %<ZName>	Y name X name Z name	Data label and marker on 2D or 3D curve and coordinates at cursor.	The names of the individual components of the data point to be labeled. %<ZName> can only be used in 3D curves. The names correspond to the data set name (Y) or the relevant comments (Y, Z).
%<YValue> %<XValue> %<ZValue>	Y value X value Z value	"	The values of the individual components of the data point to be labeled. %<ZValue> can only be used in 3D curves.
%<Y2Value> %<X2Value> %<Z2Value>	Y2 value X2 value Z2 value	Range marker on a 2D or 3D curve	The values of the individual components of the second data point of the range marker. %<Z2Value> can only be used in 3D curves.
%<YUnit> %<XUnit> %<ZUnit>	Y unit X unit Z unit	"	The physical units of the individual components of the data point to be labeled. %

Placeholder	Description	Used in	Purpose
			<ZUnit> can only be used in 3D curves.
%<DeltaY> %<DeltaX> %<DeltaZ>	Delta Y Delta X Delta Z	Range marker on a 2D or 3D curve and coordinates at cursor	The difference values of the two cursors or the two positions of the range marker. %<DeltaZ> can only be used in 3D curves.
%<DeltaYPerc> %<DeltaXPerc> %<DeltaZPerc>	Delta Y % Delta X % Delta Z %	Range marker on a 2D or 3D curve and coordinates at cursor	The difference values of the two cursors or the two positions of the range marker as a percentage, related to the first value. %<DeltaZPerc> can only be used in 3D curves.
%<OrderTrackingName>	Order tracking name	Marker with order line or frequency hyperbola and coordinates at cursor in speed-frequency spectrum or order spectrum.	Outputs either "order" (speed-frequency spectrum) or "frequency" (order spectrum).
%<OrderTrackingValue>	Order tracking value	"	The order is the ratio of speed and frequency at the marked position.
%<OrderTrackingUnit>	Order tracking unit	"	The unit of the order tracking value.
%<AreaName>	Area name	Range marker with area under curve as marker cursor	Outputs "area".
%<AreaValue>	Area value	"	Outputs the content of the area under the marked curve range.
%<AreaUnit>	Unit of area	"	The unit of the area content.
%<SlopeName>	Slope name	Range marker with tangent as marker cursor	Outputs "slope".

Placeholder	Description	Used in	Purpose
%<SlopeValue>	Slope value	"	Outputs the slope at the marked point of the curve.
%<SlopeUnit>	Slope unit	"	The unit of the slope.
%<Length>	Length	Range marker with dimension line	The length of the dimension line in the unit of the curve.
%<DataSetName>	Data set name	Data label in a 2D or 3D curve	The name of the additional data set for data labeling.
%<DataSetValue>	Data set value	"	The value of the additional data set specified for the data labeling assigned to the data point to be labeled.
%<DataSetUnit>	Data set unit	"	The unit of the additional data set for data labeling.
%<XIndex>	X index	Data label or marker	The row index of the data point to be labeled/marked.
%<ZIndex>	Z index	Data label or marker in a 3D curve	The column index of the data point to be labeled/marked in a 3D data set.
%<PlotOrder>	Plot order	Marker in a diagram	The markers are numbered in ascending order over the plot order. Markers with a larger plot order appear in the foreground. When reading the markers without sorting, they are returned according to their plot order.
%<PageNumber>	Page number	"Label" shape in document	The page number in the document where the label shape is found.
%<NumberOfPages>	Number of pages	"Label" shape in document	The number of pages in the document in which the label shape is found.
%<H1> ... %<H9>	Outline level counter 1 - 9	"Label" shape in document	The current counter value for the corresponding outline level of the document.

Placeholder	Description	Used in	Purpose
%<N>	Label counter	"Label" shape in document	The current counter value for the outline level assigned to the label.

Fields for Outputting Complex Numbers

If the data set with the contents you want to display contains complex numbers, you can use the following functions in the field:

Function	Description
RealPart(DataSet)	Displays the real part of the complex number in the data set.
ImaginaryPart(DataSet)	Displays the imaginary part of the complex number in the data set.
Absolute(DataSet)	Displays the absolute value of the complex number in the data set.
Phase(DataSet)	Displays the phase of the complex number in the data set.

Indexing Scalar Values for Display

If the data set with the value that is to be displayed does not contain a scalar value, but instead contains a data series or data matrix, you can use an Index to display a particular value. If you do not use an index, the first value is taken. If you enter the number -1 as the index, the last value is always taken.

If the data set contains a signal, signal series or space curve, you can select the desired component and append `.X`, `.Y` or `.Z` to the name of the desired Component. If you do not specify a component, the Y component is used.

Examples

`%{Signal.Y[10]}`	Displays the 11th value of the Y component of the signal.
`%{Signal.X[-1]}`	Displays the last value of the X component of the signal.
`%{DataMatrix[0][10]}`	Displays the 11th value of the first column in the data matrix.

%{SignalSeries.Z[%<Index>]}	Displays the Z value of the series of a signal series displayed as a curve.

Fields with a User Prompt

You can use special fields to embed user prompts into the text. Use this option to prompt input from the user, for instance, when documents are updated for subsequent display in the document.

There are two options for embedding a user prompt into a text. You can either build a user prompt into the FPScript expression of a field, such as %{TextInput("Please enter your name")}, or you can use a special formatter that requests input, such as %"Recorder:\DM2000\MX2000". First, you can specify default values for input, such as "DM2000" or "MX2000". These are then displayed in a drop list box. Second, FlexPro combines several user prompts embedded in the same text, and third, FlexPro remembers past entries so that you only have to enter changes when making revisions.

7 Customizing FlexPro

7.1 Customizable User Interface

Ribbon

The ribbon lets you quickly find commands that you need to complete a task. Commands are arranged in logical groups that are grouped together on tabs. Additional tabs are displayed depending on the workspace, i.e. depending on the window that is currently active in the workspace.

You can personally customize the ribbon. You can add and remove commands and macros, create your own custom tabs and groups, and show, hide or move tabs. You can undo settings you have changed at any time by resetting the ribbon to its default settings.

Quick Access Toolbar

This toolbar, which is displayed in the FlexPro title bar or under the ribbon, contains frequently used commands that you can always access without having to switch the tab of the ribbon. You can add additional commands and macros to the toolbar. You can reset the Quick Access Toolbar to its default settings at any time.

Keyboard Shortcuts

You can carry out frequent tasks faster using keyboard shortcuts. This is done by pressing one or more keys to execute a task. For example, you can save the open project database by pressing the CTRL+S key combination instead of clicking on File > Save.

You can customize keyboard shortcuts in FlexPro. You can assign keyboard shortcuts that are not yet assigned to another function and you can delete keyboard shortcuts that you do not need. The settings that you change can be reset at any time by resetting the keyboard shortcuts to the default settings.

Note The FlexPro View user interface cannot be customized.

Customizing the User Interface

Customizing the Display and Help Language

You can now change the language for the FlexPro user interface and online help as follows:

1. Select File > Options.
2. In the Options dialog box, click on the Language and Region tab.
3. Select the desired language from the Display and Help Language list box.

Notes

If you select Match Microsoft Windows, FlexPro tries to use the same language that you have set in Windows. If this is not available, English will be used.

Some of the online help contents are not available in all languages, so they are displayed in English.

The customized changes to the FlexPro Ribbon are managed separately for each language. If you change the language, you must repeat any customizations you have made.

Customizing the Ribbon

Opening the "Customize the Ribbon" Window

1. Click on File > Options.
2. In the Options dialog box, select Customize Ribbon.

Note You can also open the Customize the Ribbon window by right-clicking with your mouse on any ribbon tab and then clicking Customize the Ribbon.

Working with Tabs

Adding a Custom Tab

- Under the Customize the Ribbon list, click New Tab.

Renaming a Default or Custom Tab

1. In the Customize the Ribbon list, click on the tab that you want to rename.
2. Click Rename and then enter a new name.

Hiding a Default or Custom Tab

- In the Customize the Ribbon list, deselect the box next to the default or custom tab that you want to hide.

Note You can hide custom and default tabs. You can remove custom tabs.

Changing the Order of Default or Custom Tabs

1. In the Customize the Ribbon list, click on the tab that you want to move.
2. Click on the Move Up or Move Down arrow until the order you want is displayed.

Removing a Custom Tab

1. In the Customize the Ribbon list, click on the tab that you want to remove.
2. Click on Remove.

Note: In the Customize the Ribbon list, custom tabs and groups include "(custom)" after the name, but "(custom)" does not appear in the ribbon.

Working with Groups

Adding a Custom Group to a Tab

1. Below the Customize the Ribbon list, click on the tab to which you would like to add a group.
2. Click New Group.
3. If you want to rename New Group (Custom) click on the group to select it, click Rename and then enter the new name.

Note: You can add a custom group to a custom or default tab.

Renaming a Default or Custom Group

1. In the Customize the Ribbon list, click on the group that you want to rename.
2. Click Rename and then enter a new name.

Changing the Order of Default or Custom Groups

1. In the Customize the Ribbon list, click on the group that you want to move.
2. Click on the Move Up or Move Down arrow until the order you want is displayed.

Removing a Default or Custom Group

In the Customize the Ribbon list, click on the group that you want to remove.

Click on Remove.

Replacing a Default Group with a Custom Group

Note: You cannot remove any commands from a default group (a group integrated in FlexPro). However, you can create a custom group to replace the default group. The custom group then contains only the commands that you want displayed.

1. In the Customize the Ribbon list, click on the default tab to which you want to add the custom group.
2. Click New Group.
3. Right-click with your mouse on the new group and then click Rename.
4. Give the new group a name.
5. In the Choose commands from list box, click Main Tabs.
6. Click on the arrow (>) next to the default tab containing the group that you want to customize.
7. Click on the arrow (>) next to the default group that you want to customize.
8. Click on the command that you want to add to the custom group and then click Add.
   Note: You do not have to add all commands. You can add just the commands that you prefer.
9. Right-click with your mouse on the default group and then click Remove.

Working with Commands

Adding Commands to a Custom Group

1. In the Customize the Ribbon list, click on the custom group to which you want to add a command.
2. In the Choose commands from list box, select the list from which commands are to be added (such as Popular Commands or All Commands).

3. Click on a command from the list that you have selected.
4. Click Add.

Note: You can also rename a command and then add an icon that represents the command by clicking on the command and then clicking Rename.

Removing a Command from a Custom Group

Note: Commands can only be removed from custom groups.

1. In the Customize the Ribbon list, click on the command that you want to remove.
2. Click on Remove.

Renaming a Command That Was Added to a Custom Group

1. In the Customize the Ribbon list, click on the command that you want to rename.
2. Click Rename and then enter a new name.
3. When you rename a command that you have added to a custom group, you can also click on an icon that represents this command.

Changing the Order of Commands in Custom Groups

In the Customize the Ribbon list, click on the command that you want to move.

Click on the Move Up or Move Down arrow until the order you want is displayed.

Resetting the Ribbon

- Click Reset.

Note This discards *all* ribbon customizations.

Minimizing/Maximizing the Ribbon

To minimize the maximized ribbon

- Click on Minimize ribbon in the bottom right-hand corner of the ribbon.

Note To select a command in the minimized ribbon, you must first click on the desired tab to display it.

To maximize the minimized ribbon

1. Click on a tab in the ribbon to display it
2. Click on Pin ribbon in the bottom right-hand corner of the ribbon.

See Also

Customizing the Quick Access Toolbar 527

Customizing Keyboard Shortcuts 529

Ribbon 522

Customizing the Quick Access Toolbar

Adding a Command to the Quick Access Toolbar

1. Click on the relevant ribbon tab or group to display the command that you want to add to the Quick Access Toolbar.
2. Right-click with your mouse on the command and then in the context menu select Add to Quick Access Toolbar.

Adding a Command That Is Not on the Ribbon to the Quick Access Toolbar

1. Click Customize Quick Access Toolbar > More Commands.
2. In the Choose commands from list box, select Commands Not in the Ribbon.
3. Find the command in the list and click Add.

Removing a Command from the Quick Access Toolbar

- Right-click with your mouse on the Quick Access Toolbar command that you want to remove and then in the context menu select Remove from Quick Access Toolbar.

Changing the Order of Quick Access Toolbar Commands

1. Right-click with your mouse on the Quick Access Toolbar and then in the context menu click Customize Quick Access Toolbar.
2. In the dialog box of the list on the right, click on the command that you want to move and then click the Move Up or Move Down arrow.

Grouping Commands by Adding a Separator Between Them

1. Right-click with your mouse on the Quick Access Toolbar and then in the context menu select Customize Quick Access Toolbar.
2. In the Choose commands from list box, select Popular Commands.
3. Click <Separator> and then click Add.
4. To place the separator where you want it, click on the Move Up or Move Down arrow.

Moving the Quick Access Toolbar

You can display the Quick Access Toolbar in the title bar above or below the ribbon:

1. Right-click with your mouse on the Quick Access Toolbar.
2. In the context menu click Show Quick Access Toolbar below the Ribbon or Show Quick Access Toolbar above the Ribbon.

Customizing the Quick Access Toolbar Using the "Options" Command

You can use the Options command to add, remove or change the order of the commands.

1. Click on File > Options.
2. In the Options dialog box, select the Quick Access Toolbar tab.
3. Make any necessary changes.

Reverting to the Default Quick Access Toolbar Setting

1. Right-click with your mouse on the Quick Access Toolbar and then in the context menu select Customize Quick Access Toolbar.
2. In the dialog box, click Reset.

Customizing Keyboard Shortcuts

You can assign keyboard shortcuts to commands, macros or frequently used icons.

1. Click on File > Options.
2. Click Customize Ribbon.
3. Click on the Customize button next to Keyboard Shortcuts to open the Customize Keyboard dialog box.
4. In the Categories box click on the category that contains the command or element.
5. In the Commands box click on the name of the command or element.
   In the Current box you can see all the shortcuts that are currently assigned.
6. Carry out one of the following actions:

   Assigning a Shortcut

   1. Keyboard shortcuts must start with CTRL, ALT or another function key.
   2. Click on the New Shortcut Key box and then press the key combination that you want to assign to the action. For example, press ALT and the desired key.
   3. In the Assigned to box check whether the key combination is already assigned to another command or element. If it is, select a different key combination.
   4. Click Assign.

   Removing a Shortcut

   1. In the Current box click on the shortcut that you want to remove.
   2. Click on Remove.

   Resetting All Shortcuts

   - Click on the Reset All button.

7.2 User Profile and User Permissions

In your user profile, FlexPro stores several settings that you make while working. The following information, among other things, is included in the profile:

- All adjustments that you have made to the FlexPro user interface.
- Settings that you have made in the Analysis Wizard.
- Custom models that you have created for non-linear curve fitting.

You can export the user profile to a file and import it from a file to transfer the profile to another user.

User Permissions

In conjunction with the customizability of FlexPro's user interface, the user permissions help to distinguish between developers who create analyses and documentation with FlexPro and users who use such ready-made analyses to evaluate their measurements. By restricting user permissions, the analyses and documentation created with FlexPro can be specifically protected against incorrect operation and manipulation. User permissions are assigned by the administrator to the user logged on to Windows by means of a group policy (GPO). A configuration file (ADMX/ADML) for the Windows Group Policy Editor is available for FlexPro as a separate installation program.

Note: User profiles and user permissions cannot be edited using FlexPro View.

Working with User Profiles

Creating a User Profile for Another User

1. First save your own user profile:
   1. Click on File > Options and then click on the User Profile tab.
   2. Click Export to export your user profile to a file.
   3. If necessary, select a different folder and click Save.
2. Customize the FlexPro user interface the way that you want the other user to see it. You can find information on how to do this under Customizing the User Interface 522.
3. You can now save the user profile to a file:

   1. Click Export.
   2. If necessary, select another folder.
   3. Enter a file name, for instance, the name of the user or the name of the application to be run using this user profile.
   4. Click Save.
4. Now restore your own user profile:
   1. Click Import to import your user profile from a file.
   2. Select the file you saved in the first step.
   3. Click Open.

Note: User profiles cannot be created using FlexPro View.

Installing a User Profile for Another User

To install profile that is stored in a file for another user, the following two methods can be used:

Installation using Windows Explorer

Copy the user profile to the user computer using the file name `UserProfile.fpup` In the folder `C:\Users\`*User login name*`\AppData\Local\Weisang\FlexPro\<%VERSION_COMMERCIAL%>` or `C:>User>`*User login name*`>AppData>Local>Weisang>FlexPro><%VERSION_COMMERCIAL%>`.

Notes

- If on the System Settings tab of the FlexPro Options dialog box you selected the option Save user profile and personal template database on server, then you will have to replace the folder `Local` with `Roaming` in the paths mentioned above.
- The folders are hidden. If you cannot see them in Windows Explorer, you will need to go to the Windows Explorer View tab in the Show/Hide group and select the Selected items option.
- FlexPro should not be running on your computer while you are installing the profile.

Installation using FlexPro

1. Log into the user's computer using the user's login name.
2. Launch FlexPro.

3. Select File > Options and then click on the User Profile tab.
4. Click Import.
5. Select the file and click Open.

Saving a User Profile On a Server (Roaming)

Do the following if you want to copy your user profile to all computers you log into on the LAN:

1. Select File > Options.
2. In the Options dialog box, click on the System Settings button.
3. Select the option Save user profile and personal template database on server.

Your computer must be a member of a Windows domain and your Windows roaming user profile must be enabled.

7.3 Using Startup Switches to Launch FlexPro

1. Do one of the following:

   If you use Windows Start menu to launch FlexPro, use Windows Explorer to locate the FlexPro shortcut icon in the Start Menu folder of your Windows folder. Right-click on the shortcut icon, click on Properties and then click on the Shortcut tab.

2. Under Target, type a space after the path to the FlexPro program, and then type one or more of the parameters listed in the following table.

   For example, to open the project database Measurement.fpd, type a space after the path and then type: `C:\Project Databases\Measurement.fpd`

To Start FlexPro and...	Enter
Open a specific project database using a backup copy	*project database path\file name* or /om:b *project database path\file name*
open a specific project database directly	/om:d *project database path\file name*
open a specific project database as read-only	/om:r *project database path\file name*
Prevent display of the FlexPro startup screen and a new blank project database	/e

To Start FlexPro and...	Enter
Specify the working folder	/path *folder path\folder name*

The folder specified in the startup switch /path is used instead of the location specified in the Default working folder on the System Settings tab (File > Options).

7.4 Customizing the FlexPro Undo Function

1. Select File > Options.
2. In the Options dialog box, click on the Operation tab.
3. Select the desired options.

7.5 Customizing FlexPro Messages

You can turn off some of the FlexPro messages to avoid unnecessary repetitions. FlexPro will then display the option **Don't ask this question again** under the message.

To re-activate these messages, do the following:

1. Select File > Options.
2. In the Options dialog box, click on the Messages tab.
3. Select the desired options.

7.6 Customizing the Unit for Dimension Measures

FlexPro can display dimension measures, such as the size of page margins, in different units. To select this unit, you proceed as follows:

1. Select File > Options.
2. In the Options dialog box, click on the Language and Region tab.
3. Select the desired unit under Unit of measure.

7.7 Customizing the Decimal Symbol

You can customize the decimal symbol and the 1000th separator used for the input and output of floating point values as follows:

1. Select File > Options.
2. In the Options dialog box, click on the Language and Region tab.
3. Select the appropriate mode from the Decimal symbol list box.

Note: This setting does not affect the decimal symbol in formulas. FPScript 246 always uses the decimal point.

7.8 Customizing the Output Format of Numerical Data

1. Select File > Info > Project Database Properties.
2. In the project database Properties dialog box, click on the General tab.

To set the number of valid decimal places when outputting floating point values:

- Under Maximum number of valid places for output of numeric data enter the desired number of significant digits.

To set the output format for complex numbers:

- In the Display complex numbers list box, select with imaginary unit if the complex numbers are to be output in the format *a*+*b*i, or select separated with separator if they are to be displayed in the format (*a* , *b*).

Note: FPScript accepts both notations for complex numbers. The comma must be always used as the list separator, however.

7.9 Application Profile

By authorizing Weisang to record and transmit a user profile for FlexPro, you are assisting the Weisang developer team in its further development.

The application profile contains information about how you use the software, i.e. which commands you use and how often you use them, the help topics you view, and dialog boxes you open. It also provides information on the operating system you are using and the system settings you use to run FlexPro.

The application profile does not contain any personal or confidential information such as your name, keyboard input, passwords or hard drive data.

When launching FlexPro for the first time, you can specify whether you authorize recording of the application profile. FlexPro then transmits the data every 30 days to Weisang. If you are interested in the recorded profile, FlexPro will show it to you before it is transmitted.

You can change your settings for recording and transmitting the application profile at any time in the Options dialog box on the User Profile tab. To do this, select File > Options.

8 Automating Processes

8.1 Creating and Using Macros

Using Macros to Automate Tasks

What Is a Macro?

If you want to repeat a task in FlexPro, you can automate it using a macro. A macro consists of a series of FlexPro commands and statements grouped into a single command for automating a task. Instead of manually carrying out a series of time-consuming, repeated actions in FlexPro, you can create and run a single macro, which is a custom command that carries out the required task.

Macros are typically used for purposes such as the following:

- Speeding up editing and formatting tasks
- Combining several commands, such as commands for importing data and creating a diagram
- Providing better access to an option in a dialog box
- Automating a Complex Series of Tasks

FlexPro offers two ways to create macros. You can record a macro or use the Visual Basic Editor. The Visual Basic Editor is only available in FlexPro Professional and Developer Suite.

Recording a Macro

Recording a macro is the easiest way to create a macro. Even if you work with FlexPro Professional or FlexPro Developer Suite and you have Visual Basic Editor available for programming, it makes sense to start first with a macro recording and then modify the code as necessary. A macro is recorded in FlexPro as a series of FlexPro commands in Visual Basic for Applications. When recording macros, you can select commands and options using your mouse. You can halt recording of the macro at each step and later resume where you stopped.

To make things easier, you can use the Quick Macro in FlexPro for repetitive processes. You can easily start and stop recording this macro using the shortcut CTRL+SHIFT+R. You do not have to name or create this macro. If you want to start a new recording, any existing Quick Macro will be deleted automatically. You can run the Quick Macro as often as you like using the shortcut CTRL+SHIFT+P.

To make it easy for you to access a saved macro by name, you can add the macro to the ribbon or Quick Access Toolbar, or assign a keyboard shortcut to it. To run the macro you then only have to click on its ribbon icon or toolbar, or press the shortcut keys. You can also select Developer[Code] > Macros and then click on the name of the macro you want to run.

Creating a New Macro

Using the Visual Basic Editor in FlexPro Professional or FlexPro Developer Suite, you can create powerful custom macros with Visual Basic statements that cannot be recorded.

Where Macros Are Stored

You can save macros in template databases or in project databases. By default, FlexPro macros are saved in the personal template database so that they are available to all FlexPro project databases. If you only want to use a macro for a particular project database, save that macro to the specific project database. Individual macros in a project database are saved in macro projects which you can copy between project databases.

Protection from Documents That May Contain Viruses

Understanding Macro Viruses

A macro virus is a computer virus stored in a Macro inside a project database, a template database or an add-in. If you open this type of project database or start a procedure that calls a macro virus, the macro virus can become enabled, can be transferred to your computer and be stored in your personal template database. From that point on, every project database you open could be automatically "infected" with the macro virus - and if other people open these infected project databases, the macro virus is transmitted to their computers.

Security Levels in FlexPro

FlexPro offers the following levels of security to reduce the chances that macro viruses will infect your project databases, templates databases or add-ins:

- High   You can run only macros that have been digitally signed and that you confirm are from a trusted publisher. Before trusting a source, you should confirm that the source is reliable and uses a virus scanner before signing macros.

Unsigned macros are automatically disabled, and FlexPro opens the project database without any warning.

- Medium FlexPro displays a warning whenever it encounters a macro from a source that is not on your list of trusted publishers (described further down). You can choose whether to enable or disable the macros when you open the project database. If the project database might contain a virus, you should choose to disable macros.
- Low If you are sure that all the project databases and add-ins you open are safe, you can select this option, which turns off macro virus protection in FlexPro. At this security level, macros are always enabled when you open project databases.

If your network administrator has not enforced a security level for your organization, you can change the security level 540. If the security level for FlexPro is set to Medium or High, you can maintain a list of trusted macro sources. When you open a project database or load an add-in that contains macros developed by any of these sources, the macros are automatically enabled. Learn about Security Levels in FlexPro 550.

Digital Signatures

If you have Microsoft Internet Explorer version 4.0 or later installed on your computer, one way to identify that a macro is safe to use is by its digital signature. A digital signature on a macro is like a stamp on an envelope - it confirms that the macro originated from the macro developer who signed it and that the macro has not been altered.

When you open a project database or load an add-in that contains a digitally signed macro, the digital signature appears on your computer as a certificate. The certificate names the macro's source, plus additional information about the identity and integrity of that source. A digital signature does not necessarily guarantee the safety of a macro, and you must decide whether you trust a macro that has been digitally signed. For example, you might trust macros signed by someone you know or by a well-established company. If you are unsure about a project database or add-in that contains digitally signed macros, carefully examine the certificate before enabling macros or, for the highest level of security, disable the macros. If you know you can always trust macros from a particular source, you can add that macro developer to the list of trusted publishers when you open the project database or load the add-in.

If you yourself develop macros with FlexPro Professional or FlexPro Developer Suite, you can sign macros from within the Visual Basic Editor 539.

Digital Signatures

How Digital Signatures Work

FlexPro has security levels that allow users to run macros based on whether or not they have been digitally signed by a macro developer on their list of trusted publishers. FlexPro also checks whether the digital signature is valid. For example, FlexPro checks whether the Macro Project was changed or signed before the digital certificate expired or before it was declared invalid. In addition, it checks whether the digital certificate was distributed by a valid certification authority.

Digital signatures work only on computers that have Microsoft Internet Explorer version 4.0 or later installed. If a user tries to open a project database that contains macros (signed or not) on a computer that does not have Internet Explorer 4.0 or later installed, a standard macro virus dialog box appears giving the user options for enabling or disabling macros before the project database is opened.

Signing Macro Projects

How to proceed FlexPro uses Microsoft Authenticode technology to enable you to digitally sign macro projects you develop. First, you must obtain a digital certificate 545 and install it. Then, test your solution - whenever code in a signed macro project is modified in any way, its digital signature is removed. When your solution is ready for distribution, sign the macro project 544. If you have the proper digital certificate on your computer, the macro project will automatically be re-signed when you save it. If you want to prevent users of your solution from accidentally modifying your macro project and invalidating your signature, lock the macro project before you sign it.

What a digital signature does not do Your digital signature says only that you guarantee that this project is safe. It does not prove that you wrote the macro project. Locking and signing your macro project does not prevent another macro developer from replacing the digital signature with another signature. For example, corporate administrators may re-sign templates and add-ins so that they can control exactly what users can run on their computers.

Add-ins and digital signatures If you create an add-in that adds code to a macro project, it is recommended that your code checks whether the project is digitally signed. Users should be notified of the consequences of modifying a signed project before they continue. Modifying a project database that contains a macro project does not invalidate the signature. It is not the project database that is signed, but the macro project.

Adding a Macro Developer to the List of Trusted Publishers

To use this procedure, you must have Microsoft Internet Explorer version 4.0 or later installed on your computer and you must set the Security Level to "Medium" or "High" 540.

1. Open the project database or load the add-in that contains macros from the source you want to add to the list.
2. In the Security Warning field, check the box Always trust macros from this source.

Note: If the box Always trust macros from this source in the Security Warning field is not available, then the macros were not digitally signed. You cannot add this macro developer to the list of trusted publishers without a digital signature.

Changing the Security Level for Macro Virus Protection

1. Click Developer[Code] > Macro Security.
2. Click on the Security Level tab and then choose the preferred security level.

Creating a Macro

You can create a macro by using the macro recorder to record a sequence of actions. With FlexPro Professional or FlexPro Developer Suite, you can create a macro from scratch by entering Visual Basic for Applications code in the Visual Basic Editor. When you are recording a macro, you can use the mouse to click commands and options.

To Record a Macro in FlexPro

1. Click Developer[Code] > Record Macro.
2. In the Macro name field, enter the macro name.
3. Under Store macro in, enter the templates database or the project database where you want to store the macro.
4. In the Description field enter a description for the macro.
5. Click on OK to start recording the macro.
6. Perform the actions you want to include in your macro.

7. To stop recording the macro, click on Stop recording.

To Record a Quick Macro in FlexPro

1. Click Developer[Code] > Record Quick Macro or use the CTRL+SHIFT+R shortcut.
2. Perform the actions you want to include in your macro.
3. To stop recording the macro, click on Stop Recording or use the CTRL+SHIFT+R shortcut again.

To Pause and Resume a Macro Recording

1. To pause recording, select Pause Recording.
2. Perform any actions you don't want to record.
3. To resume recording, deselect Pause Recording.

To Create a Macro from Scratch

1. Click Developer[Code] > Macros.
2. In the Macros in list, click on the templates database or the database where you want to store the macro.
3. In the Macro name field, enter the macro name.
4. Click on Create to open the Visual Basic Editor.

Tips for Recording a Macro

- Before you record or write a macro, plan the steps and commands that you want the macro to perform. If you make a mistake when you record the macro, corrections you make will also be recorded. If you are working with FlexPro Professional or FlexPro Developer Suite, you can edit the macro later and remove unnecessary steps you recorded.
- Try to anticipate what is going to happen. For example, if your macro includes a command to close the project database, FlexPro asks you to save the project database if it contains unsaved changes. To avoid this message, record saving the project database before closing it.
- If you want to use the macro you are recording in other project databases, make sure that the macro does not depend on the current project database's contents.

- If you use a particular macro often, assign it to a ribbon button or keyboard shortcut. This allows you to run the macro directly without having to open the Macros dialog box.

Editing a Macro

FlexPro Professional and FlexPro Developer Suite allow you to edit your macros. Open a macro in the Visual Basic Editor and make corrections, remove unnecessary steps, rename or copy individual macros, or add instructions that you cannot record in FlexPro. Changes you make in the Visual Basic Editor to procedures and macro project items are reflected in FlexPro in the Macros and Organize Macro Project Items dialog boxes.

1. Click Developer[Code] > Macros.
2. Under Macro name click on the name of the macro you want to edit.

If the macro is not displayed in the list, select a different project database under Macros in.

3. Click on Edit.

Running a Macro

To Run a Named Macro

1. Click Developer[Code] > Macros.
2. Under Macro name, click on the name of the macro you want to delete.
3. If the macro is not displayed in the list, select a different project database under Macros in.
4. Click on Run.

To Run the Quick Macro

- Click Developer[Code] > Run Quick Macro or use the CTRL+SHIFT+P shortcut.

Deleting a Macro

1. Click Developer[Code] > Macros.
2. Under Macro name, click on the name of the macro you want to delete. If the macro is not displayed in the list, select a different project database or a different template database under Macros in.
3. Click on Delete.

Note: To delete several macros, hold down the CTRL key and click on the macros you want to delete in the Macro name box and then click on Delete.

Copying a Macro Project

Using the Organize Macro Project Elements dialog, you can copy a macro project to use it in another project database or in another template database.

1. Click Developer[Code] > Macros.
2. Click on Organize.
3. Select the macro project you want to copy from one of the lists, and then click on Copy.

FlexPro displays the macro projects used in the active project database in the list on the left and the macro projects in the personal templates database in the list on the right.

Note To copy an individual macro, select the macro in the dialog box Developer[Code] > Macros, click on Edit, and use the standard editing features of the Visual Basic Editor. (Requires FlexPro Professional or FlexPro Developer Suite)

Renaming a Macro Project

You can use the Organize Macro Project Items dialog box to rename a macro project. You can also rename an individual macro. 542.

1. Click Developer[Code] > Macros.
2. Click on Organize.

3. Select the macro project you want to rename from the respective list, and then click on Rename.
4. The macro projects in the currently active project database are displayed in the list on the left and the macro projects in the personal templates database in the list on the right.
5. Specifies a new name for the macro project under New name.

Deleting a Macro Project

You can use the Organize Macro Project Items dialog box to copy a macro project. You can also delete an individual macro 543.

1. Click on Developer[Code] > Macros.
2. Click on Organize.
3. Select the macro project you want to delete from the appropriate list, and then click on Delete.
4. FlexPro displays the macro projects used in the active project database in the list on the left and the macro projects in the personal templates database in the list on the right.

Digitally Signing a Macro Project

For additional information on how macro virus protection works, you can learn about digital signatures 539 and how security levels and digital signatures work together 550.

1. Open the project database or template that contains the macro project you want to sign.
2. Click Developer[Code] > Visual Basic.
3. In the Project Explorer, select the project that you want to sign.
4. In the Tools menu, point to Digital Signature.
5. Do one of the following:
    - To use the current certificate, click OK.
    - If you did not previously select a digital certificate, or if you want to use different one, click Choose, select the certificate, and then click OK.

This function is only available in FlexPro Professional and FlexPro Developer Suite.

Obtaining a Digital Certificate

Creating Your Own Digital Certificate

You can create your own digital certificate. You can create your own digital certificate by installing and using the digital certificate program that came with FlexPro Professional and FlexPro Developer Suite.

- In Windows Explorer, locate the `SelfCert.exe` file in the `C:\Program Files` (or `Program Files (x86)`) folder and double-click it.

Note: Since a digital certificate that you create yourself is not issued by a formal certification authority, macro projects that you digitally sign by using such a certificate are referred to as self-signed projects. Depending on how FlexPro digital signature features are handled in your organization, you may be prevented from using this type of certificate, and other users might not be able to run self-signed macros for security reasons.

Request a digital certificate from your organization

Some organizations and corporations may have a security administrator or group act as their own certification authority and produce or distribute digital certificates by using tools such as Microsoft Certificate Server. Microsoft Certificate Server can act as a stand-alone certification authority or as part of an existing certification authority hierarchy. This depends on how the FlexPro features for digital signatures are used in your organization. You can sign a macro project by using a digital certificate from your organization's internal certification authority. Or you might need to have an administrator sign macro projects for you by using an approved certificate. For information about your organization's policy, contact your network administrator.

Request a digital certificate from a commercial certification authority

To obtain a digital certificate from a commercial certification authority (such as VeriSign, Inc.), you or your organization must submit an application to that authority.

Depending on your status as a macro developer, you should apply for either a Class 2 or Class 3 digital certificate for software publishers:

- A Class 2 digital certificate is designed for people who publish software as individuals. This class of digital certificate provides assurance as to the identity of the individual publisher.
- A Class 3 digital certificate is designed for companies and other organizations that publish software. This class of digital certificate provides greater assurance about the identity of the publishing organization. Class 3 digital certificates are designed to represent the level of assurance provided today through retail distribution channels for software.

When you receive your digital certificate, you are given instructions on how to install it and use it to sign your FlexPro solutions.

Removing a Macro Developer from the List of Trusted Publishers

To perform these procedures, you must have Microsoft Internet Explorer version 4.0 or later installed on your computer.

1. Click Developer[Code] > Macro Security.
2. Click on the Trusted Publishers tab.
3. Select the source you want to remove from the list.
4. Click on Remove.

Troubleshooting Macro Warnings and Security Levels

Receiving Macro Warnings

No Warning Displayed When Opening a Project Database or Loading an Add-In Containing Macros.

The security level for FlexPro might be set to Low

To have FlexPro warn you that a database or add-in contains macros, change the security level to 'Medium' 540.

The security level for FlexPro might be set to 'High', and the macros are not digitally signed

To have FlexPro warn you that a database or add-in contains macros that are not digitally signed, change the security level to 'Medium' 540.

You or someone you have designated as a trusted publisher might have developed the macros

If you designated the macro developer as a trusted publisher, FlexPro opens the database and enables macros. If you no longer consider the macro developer a trusted source, you can remove the macro developer from the list of trusted publishers 546. To have FlexPro prompt you every time you open a database or load an add-in that contains macros, remove every publisher from the list of trusted publishers. If you are using FlexPro Professional or FlexPro Developer Suite, you can check whether the macros contained in the database have been signed. Click Developer[Code] > Visual Basic. In the Visual Basic Editor, click on Digital Signature in the Tools menu.

A Macro Warning Keeps Appearing.

The macro you want to run might not be from a trusted publisher

You will receive a macro warning if the security level for FlexPro is set to Medium or High, and you open a database or load an add-in that contains digitally signed macros that are not from a trusted publisher. If you are sure that the macro developer is a trusted publisher, add the name to the list of trusted publishers 540.

The database may contain a macro virus

If you do not expect the database to contain a macro, your computer might have a virus that is adding a macro virus to the database. Check your computer for viruses, and try to get an uninfected copy of the database from the source.

The database contains legitimate but unsigned macros

FlexPro cannot distinguish between safe macros and unsafe macros. If you know the macros are legitimate and safe, you might want to digitally sign those macros 544 and then add their names to the list of trusted publishers 540.

FlexPro Should Not Display Macro Warnings.

To stop displaying macro warnings, change the security level for FlexPro to 540 Low. To prevent your computer from becoming infected by macro viruses, you should only set your security level to Low if you use a separate anti-virus application that can check the FlexPro databases and add-ins for macro viruses. Otherwise, you should make sure that all macros used by you come from trusted publishers.

A Security Warning Dialog Box Appears Stating That the Source has not been Authenticated.

This warning appears in the Security Warning box if the security level for FlexPro is set to High or Medium, and you open a database or load an add-in that contains digitally signed macros, but the digital certificate has not been authenticated. For example, if the macro developer has created his or her own digital certificate, you will receive this warning. This type of unauthenticated certificate can be forged by malicious users to falsify the identity of the source of the certificate. For example, a malicious user could create a certificate with the description "Weisang & Co." The only warning that the certificate is false would come from the warning described here. You should expect professional software developers to sign using authenticated certificates. You should only accept unauthenticated certificates from co-workers you know or friends. Don't accept any from sources you don't know.

If the security level for FlexPro is set to High and you trust the macro source, you can select the Always trust macros from this source check box and enable the macros. If the security level for FlexPro is set to Medium, you can enable the macros without adding the macro developer to the list of trusted publishers. If you don't consider this macro developer a trusted publisher, do not enable the macros or do not open the database until you have verified that the publisher can be trusted.

Working with Macros

Cannot Use a Macro in an Open Project Database or Add-In.

Macros might have been automatically disabled

If the security level for FlexPro is set to High and you open a database or load an add-in that contains unsigned macros, the macros are disabled and you cannot run them. You can enable macros that are not digitally signed by changing the security level to Medium, 540 closing the database or unloading the add-in, and then opening the database or loading the add-in again. Remember to change the security level back to High if you want FlexPro to automatically disable unsigned macros in the future.

You might have chosen to disable macros when you opened the database

If the security level for FlexPro is set to Medium or High, and you have determined that the macros are to be disabled because their source is not trustworthy, you cannot run the macros. To run the macros, close the database or unload the add-in, re-open the database or load the add-in and then click Enable Macros.

Changing the Security Level

Cannot Change the Security Level for FlexPro.

Your network administrator might have enforced a security level for your workgroup or corporation to make sure you use only macros that have been deemed virus free.

For more information, contact your network administrator.

Adding a Macro Developer to the List of Trusted Publishers

Cannot Add a Specific Macro Developer to the List of Trusted Publishers.

The list of trusted publishers might be locked

Your system administrator may have locked your list of trusted publishers so that you cannot add new macro developers to it.

The macro might not be digitally signed

In order to add a macro developer to the list of trusted publishers, the macro must be digitally signed by the macro developer.

The security level for FlexPro might be set to Low

In order to add a macro developer to the list of trusted publishers, you must set the security level to 'Medium' or 'High' 540.

The Source of the Certificate I Chose to Trust Was Not Added to the List of Trusted Sources.

If you use Microsoft Internet Explorer with FlexPro, the Certificate Properties dialog box appears when you click the Details button in the Security Warning dialog box to get information about a digital certificate for a digitally signed macro in a FlexPro database.

The Trust tab in the Certificate Properties dialog box contains an option for categorizing the certificate as generally trustworthy. FlexPro, however, ignores all options that you have activated on the Trust tab. To add a macro developer to the list of trusted publishers 540 when you load databases, check the box Always trust macros from this source in the Security Warning dialog box.

Troubleshooting When Recording and Running Macros

While recording a macro an unwanted action was inadvertently recorded.

If you are using FlexPro Professional or FlexPro Developer Suite and want to undo an unwanted action in a macro, you can open the macro in the Visual Basic Editor and remove any unwanted steps. How to edit a macro 542.

If you don't want to use the Visual Basic Editor, or it is not available to you, record the macro again without the unwanted action.

Recording a macro occasionally produces an error message.

A macro you record may not run properly in every situation. If the macro cannot run, FlexPro displays an error message. Some macros depend on certain options or settings in FlexPro. For example, a macro that edits a selected object won't run properly if no object is selected. If a macro you've recorded produces an error message, note the error number. You can then search for "error messages" in Help and find information about the message you received.

Cannot edit macros while project database is open.

If you are editing macros in a read-only database, you cannot save the changes you made. For a read-only database, FlexPro adds [Read-Only] to the file name in the title bar.

Macro cannot be recorded if project database is open.

Recorded macros cannot be saved in a read-only database. For a read-only database, FlexPro adds [Read-Only] to the file name in the title bar.

Cannot run macro while project database is open.

Changes resulting from running macros cannot be saved in a read-only database. For a read-only database, FlexPro adds [Read-Only] to the file name in the title bar.

Security Levels in FlexPro

The following table summarizes the functionality of macro virus protection, taking into account the settings specified on the Security Level tab in the Security (Developer[Code] > Macro Security) dialog box and various conditions.

Condition	High	Medium	Low
Unsigned macros.	Macros are deactivated automatically and the database is opened.	Users are asked to activate or deactivate macros.	No message. Macros are activated.
Signed macros from a reliable source. Signature is valid.	Macros are activated automatically, and the database is opened.	Macros are activated automatically, and the database is opened.	No message or verification of signature. Macros are activated.

Condition	High	Medium	Low
Signed macros from an unknown author. Signature is valid.	The program shows a dialog box with information on the certificate. Users can only activate macros if they regard the author and the certification authority as a reliable source. A network administrator can block the list of reliable sources and thus prevent users from adding the developer of the macro to the list and activating macros.	The program shows a dialog box with information on the certificate. Users are prompted to activate or deactivate macros. Users can decide whether to trust the developer of the macro and the certification authority.	No message or verification of signature. Macros are activated.
Signed macros from any author. The signature is invalid, possibly because of a virus.	Users are warned of a potential virus. Macros are deactivated automatically.	Users are warned of a potential virus. Macros are deactivated automatically.	No message or verification of signature. Macros are activated.
Signed macros from any author. Verification is not possible because the public key is missing or incompatible encryption methods have been used.	User is warned that signature verification is not possible. Macros are deactivated automatically.	User is warned that signature verification is not possible. The user is asked to activate or deactivate macros.	No message or verification of signature. Macros are activated.
Signed macros from any author. The macro project was signed after the certificate had expired or had been revoked.	User is warned that the signature has expired or has been revoked. Macros are	User is warned that the signature has expired or has been revoked. The user is asked to activate or deactivate macros.	No message or verification of signature. Macros are activated.

Condition	High	Medium	Low
	deactivated automatically.		

8.2 Automation Using FlexPro Visual Basic

Getting Started with FlexPro Visual Basic

Objects, Understanding Properties and Methods

Objects are the fundamental building blocks of Visual Basic. Virtually everything you do in Visual Basic involves modification of objects. Every FlexPro element (databases, diagrams, tables, documents, data sets, etc.) can be represented by a Visual Basic object.

What are Objects and Collections?

An object represents a FlexPro element, such as a database, a diagram, or an individual curve in a diagram. A collection is an object containing several other objects that are usually of the same kind. All the curves in a diagram, for example, are contained in a single collection object. Using properties and methods, you can modify an individual object or a whole collection of objects.

What is a Property?

A property is an attribute of an object or an aspect of its behavior. For example, properties of a diagram include its name, the appearance of the division grid and whether or not the cursors are enabled. To change the characteristics of an object, you need to change the values of its properties.

To specify the value of a property, first reference the object, then enter a dot, the property name, an equals sign and the new value of the property. In the following example, a new comment is assigned to the root folder of the MyDatabase database.

```
Databases("MyDatabase.fpd").RootFolder.Comments = "My Database"
```

In this example, Databases references the collection of open databases. The name MyDatabase.fpd identifies a single database in the collection. The Comments property specifically applies to the root folder of this database.

Some properties cannot be set. The help topic for a property specifies whether this property can be set (read/write access) or whether it can only be read (read-only access).

You can obtain information about an object by calling the value of one of its properties so that it will be returned. In the following example, the name of the active database is returned.

```
databaseName = ActiveDatabase.Name
```

In this example, ActiveDatabase references the database in the active window in FlexPro. The name of this database is assigned to the variable databaseName.

Note: The help topic of a property specifies whether you can set this property (write-read access), only read it (read-only access) or only write to it (write-only access). The Object Browser in the Visual Basic-Editor also displays the read/write status at the bottom of the browser window when the property is selected.

What is a Method?

A method is an action that can be executed by an object. For example, since a FlexPro document can be printed, the Document object also features a PrintOut method. Methods often have arguments that more clearly define the way the action is to be executed. In the following example, the first three pages of the active document are printed.

```
ActiveDatabase.ActiveObject.PrintOut FirstPage:=1, LastPage:=3
```

In most cases, methods are actions and properties are characteristics. The use of a method means that something happens to an object, whereas the use of a property returns information on the object or causes an attribute of the object to be changed.

Returning an Object

Most objects are returned by returning an individual object from a collection. The Databases collection, for example, contains the open FlexPro databases. Use the Databases property of the Application object (the topmost object in the FlexPro object hierarchy) to return the Databases collection.

Once you have accessed the collection, you can return an individual object by specifying an index value in brackets (similar to working with arrays). The index value is usually a number or a name.

In the following example, the Databases property is used to access the Databases collection. The index number is used to return the first database in the Databases collection. The Close method is then applied to the Database object to close the first database in the Databases collection.

```
Databases(1).Close
```

In the following example, a name (specified as a string) is used to identify a Database object within the Databases collection.

```
Databases("Report.fpd").Close
```

Collection objects often have methods and properties which can be used to modify the complete collection of objects. The Databases object has a Save method that saves all documents in the collection. In the following example, the open documents are saved using the Save method.

```
Databases.Save
```

The Database object also features a Save method for saving an individual database. In the following example, the Report.fpd database is saved.

```
Databases("Report.fpd").Save
```

If an object located further down in the FlexPro object model is to be returned, you have to "drill down" to this object by using properties and methods that return objects.

To see how this works, open the Visual Basic Editor and in the View menu click on Object Browser. In the Classes list on the left, click on Application. In the members list on the right, click on ActiveDatabase. The text at the bottom of the Object Browser shows that ActiveDatabase is a read-only property that returns a Database object. At the bottom of the Object Browser, click on Database. The Database object is then automatically selected in the Classes list and the Members list shows the members of the Database object. Scroll through the list of members until you find Close. Click on the Close method. The text at the bottom of the Object Browser window shows the syntax for the method. For more information about this method, press F1 or click on the Help button and navigate to the help topic covering the Close method.

Leveraging this information, you can write the following statement for closing the active database.

```
ActiveDatabase.Close SaveChanges:=fpSaveChanges
```

In the following example, the caption of the active window is changed.

```
ActiveDatabase.ActiveWindow.Caption = "My Window"
```

The ActiveWindow property returns a Window object that displays the active window. Use the Caption property to specify the title of the window.

In the following example, a new database is created and the Save As dialog box appears so that the database can be given a name.

```
Databases.Add.Save
```

The Databases property returns a Databases collection. The Add method creates a new database and returns a Database object. The Save method is then applied to the Database object.

As you can see, methods and properties are used to drill down to an object. This means that you return an object by applying a method to a higher-level object in the object hierarchy or accessing a property. Once you have returned the required object, you can use the methods of this object and control its properties. For an overview of the object hierarchy, go to FlexPro Object Model 562.

How to Get Help on Objects, Methods and Properties

Until you are familiar with the FlexPro Object Model, you can use a few resources that will help you to drill down through the hierarchy.

- Listing Members automatically. Typing a period (.) after an object in the Visual Basic Editor displays a list of the available properties and methods. If, for example, you enter Application., a drop-down list of the methods and properties of the Application object appears.
- Help. You can also use Help to find out which properties and methods can be used for an object. Every object topic in Help contains properties and methods sections, which list the properties and methods for the object. In the Object Browser or in a module, press F1 to get to the corresponding help topic.
- FlexPro Object Model 562. This topic explains how FlexPro objects are arranged in the hierarchy. Click on an object in the image to display the corresponding help topic.
- Object Browser. The Object Browser in the Visual Basic Editor displays the members (properties and methods) of the FlexPro objects.

Understanding Events

Events take place when certain program situations occur. These events can be handled in event procedures initiated by the user.

Using Events

In FlexPro, you can implement event procedures at the database, object class (AnyCursorObject, AnyDocObject, AnyFolder, AnyFormula, AnyFpObject, AnyValueObject), object or application level.

The ObjectOpened event, for example, occurs at the object and object class level, whereas the DatabaseSavedevent is available at the database and the application level. The DatabaseSaved event for a database occurs after this database has been saved. At the application level, the DatabaseSaved event occurs once one of the open databases has been saved.

Enabling Events

Use the EnableEvents property of the Application object, you can enable or disable the events.

For example, saving a database with the Save method triggers the BeforeDatabaseSave event. This can be prevented by setting the EnableEvents property to False before you call up the Save method.

```
Application.EnableEvents = False

ActiveDatabase.Save

Application.EnableEvents = True
```

By default, events in FlexPro are disabled, i.e. Application.EnableEvents is set to False. You can enable events by setting the EnableEvents property to True or change the default settings in the File > Options dialog box on the System Settings tab and select Enable Macro Events.

Note: If you are creating an event procedure in an object module for the first time, you will be notified if the events option has been disabled.

Event Types

There are basically two types of events in FlexPro. First, there are events that occur once a certain condition occurs. In this case, the application developer is only notified of the change of condition (e.g. ValueModified). The second type includes those events that can be influenced by the application developer. These events form sequences of up to three events. The sequence appears as follows: It starts with a QueryCancel*EventName* event, which allows the application developer to decide whether a certain program condition may occur. If this event returns True, FlexPro discontinues processing, triggering the *EventName*Cancelled event. If the QueryCancel*EventName* event is not processed or if it returns False, FlexPro will continue processing, triggering the Before event just before the event occurs. Eventually, once the program condition has been reached, the *EventName* event will be initiated.

The following VBA code describes this logic using the ObjectOpened event as an example:

```
' The oObject object is opened by double-clicking in the
' list view.
If QueryCancelOpenObject(Object) Then
    OpenObjectCanceled oObject
Else ...
    BeforeOpenObject oObject
    oObject ... ' Object is opened ...
    ObjectOpened oObject
End If
```

Forwarding Events

In FlexPro, events initiated for an object are not just signaled with this object, they are also forwarded to the template databases and/or parent objects in the object model 562. Thus, events of similar objects in the templates databases or in parent objects of the project database can be processed simultaneously.

For example, once the database "Data.fpd" has been saved, the following event procedures are executed one after the other, depending on whether or not they exist:

- DatabaseSaved event procedure of ThisDatabase of the "Data.FPD" database
- DatabaseSaved event procedure of ThisDatabase of the personal template database
- DatabaseSaved event procedure of Databases collection of the "Data.FPD" database
- DatabaseSaved event procedure of Application object

The first argument of an event is always the object concerned. Therefore, an event procedure to which the event was forwarded will know for which object the event just processed has occurred.

The object documentation in the Reference section will tell you to which objects events are forwarded.

Forwarding Events for FlexPro Objects

Forwarding events is particularly important for FlexPro objects. Since a FlexPro database may include many FlexPro objects for which there is no corresponding object module in the Visual Basic Project Explorer, the best practice is to complete the event procedures for one, several or all FlexPro objects in the class object modules AnyCursorObject, AnyDocObject, AnyFolder, AnyFormula, AnyFpObject and AnyValueObject.

For example, if the ObjectModified event of the FpObject object occurs, the following event procedures are executed one after the other, depending on whether or not they exist:

- ObjectModified event procedure of the object concerned
- ObjectModified event procedure of AnyFpObject in the database containing the object
- ObjectModified event procedure of AnyFpObject in the personal template database

The described event cascading applies to all FlexPro objects.

Tips on Working with FlexPro VBA

This topic provides some information on issues that might occur when programming in Visual Basic.

Using the Set Statement When Assigning the Value Property to Variables

When assigning the Value property of a ValueObject object to a variable, you have to consider that the Value property is of the Variant type and can also return complex values, signals and lists in addition scalar values and arrays. Complex values, signals and lists are themselves objects in FlexPro and have to be assigned to variables using the Set statement.

The following statement only represents a valid assignment of a value if the value of oValueObject is neither complex nor a signal or list:

```
Dim vtValue As Variant
Dim oValueObject As ValueObject
    ...
vtValue = oValueObject.Value
```

If oValueObject.Value is complex or a signal or list, the error message 'Object does not support this property or method' appears. When assigning complex values, signals, or lists, you should therefore use the Set statement:

```
set vtValue = oValueObject.Value
```

When you are writing a macro and you do not know whether a value will be complex or real at runtime, using the IsObject statement can avoid errors when the macro is run:

```
If IsObject(oValueObject.Value) Then
    Set vtValue = oValueObject.Value
Else
    vtValue = oValueObject.Value
End If
```

If the value of oValueObject is an object, the Set statement is used; otherwise, the value has to be assigned using '='.

Problems when Using OLE_COLOR Color Values

The OLE_COLOR type is an OLE standard data type for storing RGB color values. In C/C++, the data type is declared as an unsigned Long value. VBA does not support unsigned values. Consequently, use of OLE_COLOR may occasionally cause problems. The following code might generate the error message 'Variable uses Automation type not supported in Visual Basic':

```
If ActiveDatabase.ActiveObject.Selection.FillFormat.Color <> _
    fpColorRed Then
    ...
End If
```

There are two ways to avoid this error message. The first is to convert the color values to a Long value before the comparison:

```
If CLng(ActiveDatabase.ActiveObject.Selection.FillFormat.Color) <> _
    fpColorRed Then
    ...
```

```
End If
```

The second is to introduce an OLE_COLOR variable, assign the color value to it, and then use the new variable for the comparison:

```
Dim clr As OLE_COLOR
clr = ActiveDatabase.ActiveObject.Selection.FillFormat.Color
If clr <> fpColorRed Then
    ...
End If
```

Deleting FlexPro Objects from a Collection within a For...Each Loop

Deleting FlexPro objects from a collection using a For...Each loop should be avoided, since it could cause unwanted results, and some elements in the collection might be missed.

If the elements of any collection of FlexPro objects are to be deleted, the following code leads to the desired result:

```
While oColl.Count > 0
    oColl.Remove 1
Wend
```

or

```
While oColl.Count > 0
    oColl(1).Delete
Wend
```

Converting Numbers to Strings Using the CStr Function

When using the CStr function to convert numbers to strings, you should consider that the function always uses the current Region and Language Options settings (Locale) for the conversion. On a German system, a ',' is used to separate the integral part of a decimal number from its fractional part, whereas the '.' is used on a British or American system. If you want to convert numbers independently of the system settings, you have to use the Windows API function SetLocaleInfo to change the system settings temporarily. In the A Custom Import Filter 588 example you can see how this API function is used.

Preventing Error Messages for Fully Automated Processes

For fully automated processes that are to run without user interaction, you first have to make sure that all errors are handled by the macros themselves (consistent use of the On Error statement). Furthermore, make sure the following setting is enabled in the Visual Basic environment: under Tools > Options on the General tab, the setting for Error Trapping must be set to Break on Unhandled Errors. Otherwise, the execution of the program might be interrupted if allowed errors occur, and the error might be displayed, requiring user interaction (also see the DisplayAlerts property of the Application object).

Evaluation of Expressions in VBA (Note for C/C++ Developers)

As opposed to C/C++, where expressions are only evaluated from left to right for as long as necessary during condition evaluation, VBA always evaluates the complete expression.

The following C/C++ expression

```
(1) if(exp1 && exp2)
        ...
```

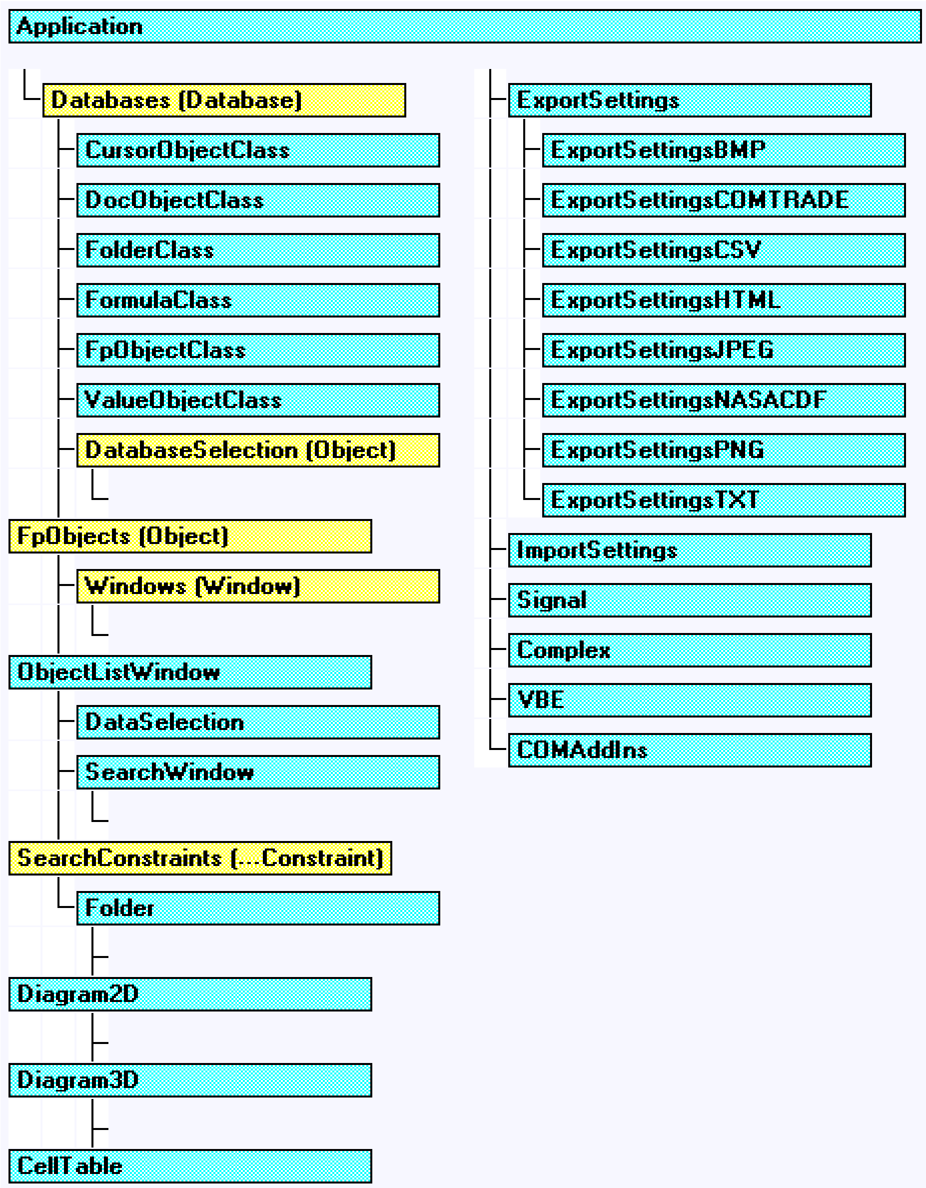
corresponds to the following in VBA code

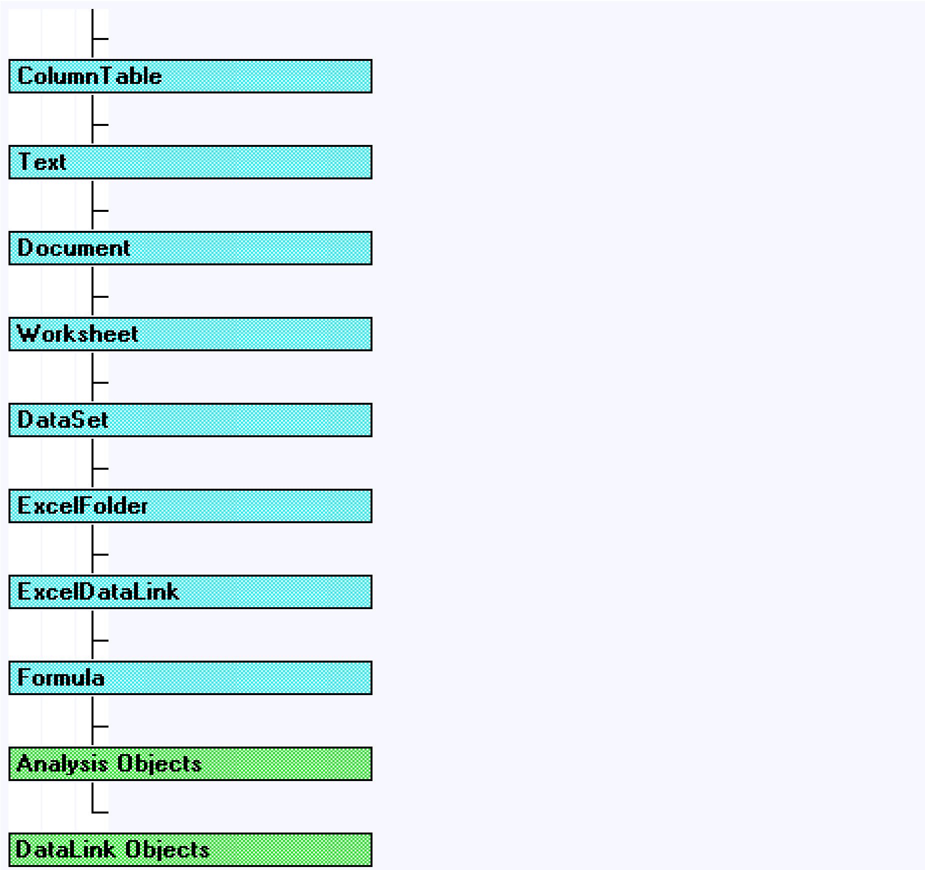
```
(2) If exp1 And exp2 Then
        ...
```

If in (1) exp1 returns False, exp2 will no longer be evaluated and the code in the If statement will be executed immediately. In (2) exp2 is evaluated regardless of whether exp1 returns False or not. If exp1 must return False to validate exp2, the statement in VBA has to be disassembled:

```
If exp1 Then
    ...
    If exp2 Then
        ...
    End If
        ...
End If
```

The FlexPro Object Model





Legend

Object and Collection

Object only

Virtual object, only for structuring

Understanding the FlexPro Object Hierarchy

As opposed to most Automation object models such as that of Microsoft Office, FlexPro objects are structured hierarchically. This means that properties and methods that apply to several objects are combined to form a base object. Structuring is multi-layered, ranging from general to specific. The object hierarchy of FlexPro applies a powerful principle of object-oriented programming, called inheritance. The following illustration shows the object hierarchy of the FlexPro 2D diagram.

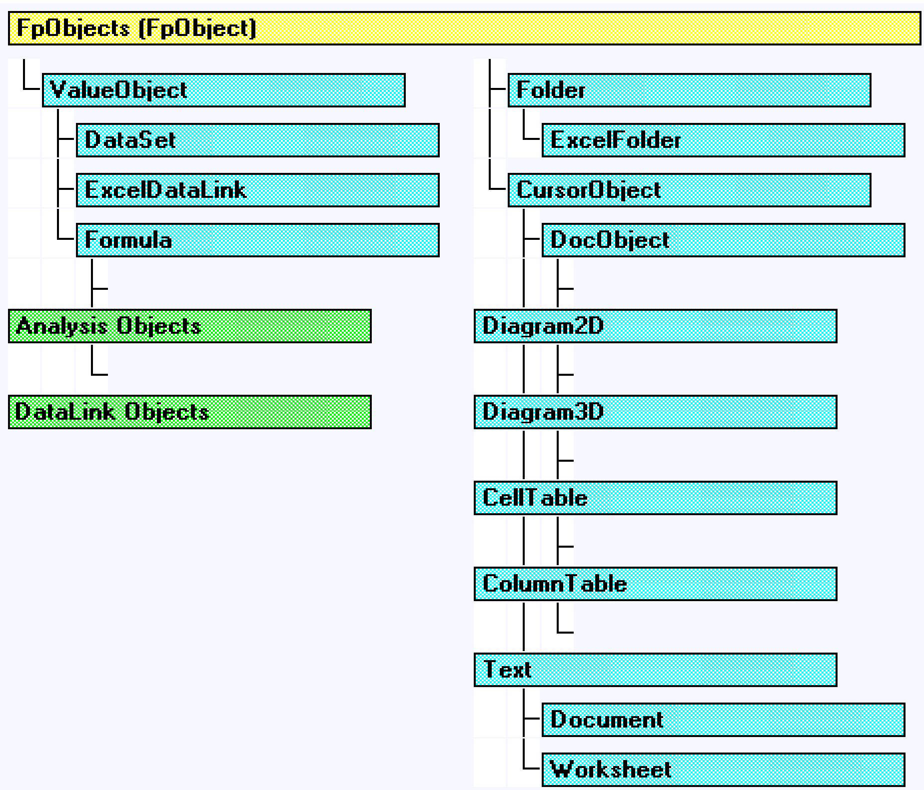
Inheritance offers a number of advantages with regard to both implementation of FlexPro and use of the FlexPro object model. For the implementation of FlexPro, inheritance means that the implementation of a base object can be re-used in all objects that are based on this base object. The software is therefore more compact and easier to maintain. In the above, example, the CursorObject is the base object for the 2D diagram (Diagram2D, the 3D diagram (Diagram3D), the worksheet (Worksheet) and the document (Document). One implementation is being used for four different objects. The further up you move in the object hierarchy, the greater this so-called "code re-use". For example, all objects that can be stored in the FlexPro database are derived from FpObject.

In addition, during the development of automation applications that access the FlexPro object model, the FlexPro object hierarchy represents a major advantage. For example, a program that only uses the properties and methods of the CursorObject interface will run with all FlexPro CursorObjects (2D diagram, 3D diagram, etc.) without requiring changes even if later versions of FlexPro contain additional objects derived from CursorObject. The compiler (Basic, C++, Java ...) used for the implementation of the application can generate very fast code because all methods and properties of CursorObject are known at compile time. Thus, calls can be directly converted to addresses, which considerably accelerates the subsequent processing of the program. In Basic you can benefit from this by choosing the CursorObject type for the object variable. You therefore tell the compiler which set of properties and methods will be available at the time the program is compiled.

If FlexPro did not have an object hierarchy, an object variable that is to have any object that supports cursors would have to be declared an unspecified Object type.

Such a program could also work with new objects supporting cursors without requiring changes. The processing speed, however, would be much lower. Since the compiler does not know the properties and methods of the actual object held by the object variable during compiling of the program, the address, such as that of a method, cannot be established until the program is executed. Using the name of the method, its index in the interface is identified. The method is then called using the generic Invoke function. All arguments have to be packed in what are called variants, since their data types are not known at the time of compilation. Another serious disadvantage of this procedure is that many errors, such as "wrong method name," "wrong number of arguments" or "wrong argument data type," will not be identified until the program is executed.

Overview of FlexPro's Object Hierarchy



Legend

Object and Collection

Object only

Virtual object, only for structuring

What's New in the FlexPro 2021 Object Model

The following sections include a list of all changes made to the FlexPro Object Model.

The changes in the FlexPro 2021 Automation Object Model from the previous version mostly involve backward compatible extensions. The few changes that may require you to change your code are in red in the text below.

If you want to import macros or Automation code from FlexPro 6 through FlexPro 2017 to FlexPro 2021, you should first take a look at the changes listed in the linked Help pages from the previous versions of FlexPro.

Note: You can obtain more information about the significant features of the listed objects, methods and properties by clicking on the links.

New Objects

Presentation

The Axis2DFloor object represents the floor lines of an X or Y axis of a 2D diagram.

The sections of a document were not mapped as explicit objects in the previous version of the object model, but only addressed implicitly via the consecutive page number. This has been improved. The new DocumentSections collection contains the sections of a document and the DocumentSection object represents a document section.

The ControlPanel object represents the new presentation object Control Panel.

The Button, DropDown, CheckBox, Spinner, ListBox, OptionButton, GroupBox, ScrollBar and EditBox objects represent different controls that you can use like shapes in documents, presentation objects and panels.

The FormControlUpdate object represents the settings for the object update of a control triggered by a control.

Analysis

The SyncOrderAnalysis object represents the new analysis object Revolution Synchronous Order Tracking.

The SyncOrderFilter object represents the new analysis object Revolution Synchronous Order Filter.

The RevolutionSyncSamplingAnalysis object represents the new Revolution Synchronous Sampling.

The HarmonicFilter object represents the new analysis object Revolution Synchronous Harmonic Filter.

The RosetteTransformationAnalysis object represents the new analysis object Rosette Transformation.

Changes to Existing Objects

The Application object has been expanded to include the UpdateUI method.

The Database object has been expanded to include the properties Active, Visible, WindowState, Left, Top, Width, Height, Caption and ShowHiddenObjects, which refer to the main window where the project database is displayed. The OpenObjects collection contains all objects open in the project database.

The objects Window and SearchWindow have also been expanded to include the Active property.

The argument PageNumber of the Export method, which can be used to export any object, has been replaced by the variant argument PageNumbers which can contain not only a single page number, but also a field with multiple page numbers.

All FlexPro objects have the additional properties DoNotIndex and Hidden.

The AlgorithmDerivation property of the SignalAnalysis object has been renamed to AlgorithmDerivative.

The Import method of the Folder object has been expanded to include the optional ItemFilter argument and the Interactive argument has been replaced by the NameClashHandling argument, which allows finer behavior control.

The CursorObject object, which represents all objects that use cursors, has been extensively revised. Some properties were moved to the CursorProperties object, but are still supported to preserve backward compatibility. The new collections MultiMarkers, ImageMarkers and RangeMarkers contain the markers of the corresponding type. The ShowCoordinates property has been renamed to ShowCoordinatesWindow.

The DocObject object, which represents all presentation objects, has been extended with the SharedPages property, which represents a collection of pages with shapes that are displayed in all views of the object.

The CursorProperties object has been expanded to include the properties AuxiliaryCursors, LineFormat, InactiveColor, CoordinatesPosition, Text, Font, NumberOfSidebandCursors, SidebandCursorSpacing, NumberOfOrderTrackingCursors and OrderTrackingCursorSpacing. The properties CursorType, CursorSynchronizationMode and CursorSynchronizationDataSet have been renamed to Type, SynchronizationMode and SynchronizationDataSet. The properties SynchronizeByIndex and HarmonicCursor have been removed, but are still supported to preserve backward compatibility.

The properties PositionIndex, PositionX, PositionY and PositionZ of the FpCursor object have been expanded to include the optional ZIndex argument. The methods DeleteValue, DeleteRangeBetweenCursors, XShiftRangeBetweenCursors and YShiftRangeBetweenCursors and the properties Marker, CursorProperties andAuxiliaryPositionsParam have been added. The PlaceMarker method has been expanded so that all types of markers can be set. The methods PlaceRangeMarker and PlaceImageMarker have been removed, but are still supported. The signatures of the methods CopyRange and CopyMarkers have been expanded. The properties HarmonicsPositionIndex, HarmonicsPositionX and HarmonicsPositionY have been renamed to AuxiliaryPositionsIndex, AuxiliaryPositionX and AuxiliaryPositionsY. However, the previous names will continue to be supported. The RemoveAllMarkers method has been expanded to include the optional argument CursorScope argument.

The CopyCoordinate method of the CoordinateWindow object has been expanded to include the optional ToClipboard argument.

The Markers collection has been extensively revised. The properties PositionIndex, PositionZIndex and PositionCurve have been renamed to PositionIndex, PositionZIndex and PositionCurve and expanded to include the optional arguments Curve and ZIndex. However, the previous names are still supported. The following properties have been added: The properties PositionX, PositionY and PositionZ and the properties PositionCurve2, PositionIndex2 and PositionZIndex2 have also been expanded to include the optional arguments Curve and ZIndex. Since the properties provide an array, there may be an incompatibility here in Visual Basic. For example, in the previous version of FlexPro, the 0 in Markers.PositionX(0) was interpreted as an index to the array returned by PositionX. In FlexPro 2021, the 0 is now interpreted as an optional argument Curve of the PositionX(Curve, ZIndex) property. You must therefore change the code in Markers.PositionX()(0). The same applies to the other properties. The features PositionX2, PositionY2, PositionZ2, PositionSlope, PositionOrderTrackingValue and AreaOfRange have been added. The properties

FrameColor, LineColor and BackgroundColor have been replaced by the properties Border, FillFormat, CursorFormat and SymbolFormat. However, the previous names are still supported. The Position property has been renamed to LabelPosition and the Type property has been replaced by three properties: LabelType, CursorType and SymbolType. The SymbolSize and TextAngle properties have been added. The properties LabelAlignment, LabelLeftPadding, LabelRightPadding, LabelTopPadding and LabelBottomPadding have been renamed to TextAlignment, TextLeftPadding, TextRightPadding, TextTopPadding and TextBottomPadding.

Similarly, the Marker object has been extensively revised. The properties PositionIndex, PositionZIndex and PositionCurve have been renamed to PositionIndex, PositionZIndex and PositionCurve and have been expanded to include the optional Index argument. However, the previous names are still supported. The properties HorizontalOffset, VerticalOffset, PositionX, PositionY and PositionZ have been expanded to include the optional Index argument. The properties TextColor, FrameColor, LineColor and BackgroundColor have been replaced by the properties Font, Border, FillFormat, CursorFormat and SymbolFormat. However, the previous names are still supported. The following properties have been added: PositionX2, PositionY2, PositionZ2, PositionSlope, PositionOrderTrackingValue and AreaOfRange. The Position property was renamed to LabelPosition and the three properties LabelType, CursorType and SymbolType were added. PlotOrder, ShowInAllViews, SymbolSize, TextAngle, TextAlignment, TextLeftPadding, TextRightPadding, TextTopPadding and TextBottomPadding.

The DocObjectSelection object has been extended to include the OpenObject, MoveFromPageToSharedPage, MoveFromSharedPageToPage and Protect methods.

The WorksheetSelection object has been extended to include the OpenObject method.

The Diagram2D object has been expanded to include the TemplateCurve properties.

The Axis2D object has been expanded to include the Floorproperties.

The Axis2DScaling object has been expanded to include the properties SymmetricalScaling, AlignOrigin and AlignDivisions.

The objects Curve2DSymbols, Curve2DIndicators, Curve3DSymbols and CurveLabels have been expanded to include the properties PlacementMode, OffsetPercentage and Count.

The properties ZoomToFitPage and PrintBlackAndWhite have been removed from the objects DocObject, Worksheet and Document, but are still supported. The properties PrintPageLayout, TopPageMargin, LeftPageMargin, RightPageMargin, BottomPageMargin and MirrorMargins have been added. The properties LeftMargin, RightMargin, TopMargin and BottomMargin.have been removed from the Document object.

The Document object has been extended by the properties PageColor, HeaderAndFooter, ResetNumbering, StartAt, Sections and CurrentSection.

The Shapes collection has been extended by the method AddFormControl, LeftAnchor, RightAnchor, TopAnchor and BottomAnchor extended.

The Shape object has been expanded to include the properties Protected, LeftAnchor, RightAnchor, TopAnchor, BottomAnchor and ShowOnInsertedPages extended. Furthermore, the object was extended by the properties Item and Count, which provide the contained elements and their number for a shape of type Group. The existing Count property, which returned the number of vertices in the shape, has been renamed toNumberOfPoints.

The Axis3DScaling object has been expanded to include the SymmetricalScaling property.

The data type of the UseAlternatingColors property of the ColumnTable object has been changed to FpTriState.

The ColumnTable object has been expanded to include the following properties: Orientation, Pagination, TableSpacingHorizontal and TableSpacingVertical.

The objects ColumnTableTitle and ColumnTableColumnTitle have been expanded to include the Angle property.

The objects ColumnTableColumnData, ColumnTableColumnRowIndex and ColumnTableColumnColumnIndex have been expanded to include the properties Angle, WrapText and AlignToDecimalSeparator.

The StoredInDatabase property of the Media object has been replaced by the Source property and the CursorFormat property has been renamed to CursorOutputFormat.

The objects FourierSpectralAnalysis, TimeFreqSpectralAnalysis and CrossSpectralAnalysis have been expanded to include the properties Angle, StepMode and GapSize.

Working with Objects

Working with FlexPro Project Databases

FlexPro stores all objects in a project database. You can apply a hierarchical structure to this project database by creating folders. Thus, a FlexPro project database resembles a hard disk on your computer. Instead of files on the hard drive, a FlexPro project database contains data sets, formulas, presentation objects, worksheets and documents that you need for your analysis. Just like on a hard drive, you can build up a hierarchy of folders in a FlexPro project database to organize your FlexPro objects.

Creating a Project Database

FlexPro can open multiple databases at once that are then stored in the Databases collection. To create a new, empty database, use the Add method of the Databases collection. The Add method provides the newly created database in such a way that you can assign it to a variable for later use:

```
Dim Doc As Database
Set Doc = Databases.Add
```

Opening a Database

Use the Open method of the Databases collection to open a database stored in a file.

```
Dim Doc As Database
Set Doc = Databases.Open("C:\Databases\MyDatabase.fpd", fpOpenModeBackup)
```

Saving a Database

If the database is new and has not been saved yet, use the SaveAs method. The second argument determines how the database is to be saved.

```
Doc.SaveAs "C:\Databases\MyDatabase.fpd", fpSaveModeOneFile
```

Use the Save method to save a database that has already been opened as a file and has been saved at least once before.

```
Doc.Save
```

Use the Save method of the Databases collection to save all databases that are currently open. For those databases that do not have a file name yet, a dialog box appears, in which you can specify the name.

```
Databases.Save
```

Closing a Database

You can close an individual database using its Close method.

```
Doc.Close fpSaveChanges
```

Alternatively, you can close all databases that are currently open.

```
Databases.Close fpPromptToSaveChanges
```

Accessing Databases

To access a database in the Databases collection, use the path name or the name that is displayed in FlexPro's main window.

```
Set Doc = Databases("C:\Databases\MyDatabase.fpd")
```

Use the ActiveDatabase method of the Application object to access the currently active database.

```
ActiveDatabase.Close
```

Creating Objects and Folders

Use the Add method of the Folder object to create objects and folders in the database. Use the RootFolder property of the Database object to access the root folder.

```
Dim Fld As Folder

Set Fld = ActiveDatabase.RootFolder.Add("Folder", fpObjectTypeFolder)
```

The Add method can also add other objects, returning the new object as the result.

```
With Fld.Add("Series", fpObjectTypeFormula)

    .Formula = "(100, 0.0, 0.1)"

End With
```

Accessing Objects

To access an object in a folder, use the Object method of the folder. You can specify the object name and object type or just the name with the extension related to the specific type.

```
ActiveDatabase.RootFolder.Object("Diagram", fpObjectTypeDiagram2D) _

    .Comments = "Measurement 1"
```

or

```
ActiveDatabase.RootFolder.Object("Diagram.2D").Comments = "Measurement 1"
```

Using the Parent property, which is available to every FlexPro object, you can access the object (usually a folder) that contains the object. The following example accesses the comment added to a folder containing the data set that has been assigned to the "Dataset" object variable.

```
Dataset.Parent.Comments = "Measurement 1"
```

Searching the Database for Objects

Use the Objects method of the Folder object to search for objects. Several options are available.

The following example finds all data set objects that have names beginning with "sig".

```
Dim Objects
Set Objects = ActiveDatabase.RootFolder.Objects("^sig*\.dat$")
```

The regular expression is structured as follows:

Sequence	Meaning
"^sig"	Specifies that the object name must start with "sig".
".*"	Represents a string of any length, consisting of any characters.
"\.dat$"	Specifies that the object name must end in ".dat", i.e. that the object must be a data set.

The following example finds a dataset object called "Signal" and a 2D diagram called "Plot".

```
Dim Objects
Set Objects = ActiveDatabase.RootFolder.Objects("Signal", "Plot.2d")
```

The following example finds all objects in which the value of the Origin property is "Measurement 1".

```
Dim Constraints As New SearchConstraints
With Constraints.Add
    .SearchItem = fpSearchItemOrigin
    .CompareOperation = fpSearchCompareOpTextIs
    .SearchValue = "Measurement 1"
End With
Dim Objects
Set Objects = ActiveDatabase.RootFolder.Objects(Constraints)
```

Use the Objects method of the Database object to access objects in different folders simultaneously. The following example deletes two objects.

```
ActiveDatabase.Objects(Array("\Diagram.2d", _
    "\Measurement1\Signal")).Delete
```

Use the Search method of the Folder object to search through all or parts of the database for objects. The following example searches for all data sets of which the maximum value is larger than 10 and moves these data sets to the "Data Folder" folder.

```
Dim Constraints As New SearchConstraints
With Constraints.Add
    .SearchItem = fpSearchItemObjectType
    .CompareOperation = fpSearchCompareOpObjectTypeIsKindOf
    .SearchValue = fpObjectTypeDataSet
End With
With Constraints.Add
    .SearchItem = fpSearchItemMaximum
    .CompareOperation = fpSearchCompareOpValueGreater
    .SearchValue = 10#
End With
ActiveDatabase.RootFolder.Search(True, Constraints). _
    MoveTo ActiveDatabase.Object("\Data Folder.FLD")
```

Working with Data Sets

The FlexPro data sets are largely determined by these four properties: DataStructure, DataType, NumberOfRows and NumberOfColumns. The DataStructure property determines the data structure of the data set. FlexPro offers support for nine data structures, six of which are composed of up to three components: The DataType property determines the data type of the data set or, if the data set has a composite data structure, the data types of the individual components it contains. FlexPro also supports several data types for numbers, text, Boolean values, times and duration. Not all data types are allowed for all components of a data set. Strings, for example, cannot be used as a data type for data sets that have a composite data structure. The two properties NumberOfRows and NumberOfColumns determine the size of a data set. NumberOfRows can be used with all data structures except scalar values and determines the number of data rows. NumberOfColumns is only relevant for the two-dimensional data structures data matrix and signal series, and determines the number of columns. For all other data structures, this property has the value One and cannot be changed. A change in the NumberOfRows or NumberOfColumns may effect several components of a data set. If, for example, you increase the number of rows in a signal, its X and Y components will increase.

Creating a Data Set

The best practice is to use FlexPro to generate the right code for the creation of a data set by activating the macro recorder and then creating a data set using the wizard. For example, for a signal series with three columns containing 100 values each, the code should look something like this:

```
With ActiveDatabase.RootFolder.Add("SignalSeries", fpObjectTypeDataSet)
    .DataStructure = fpDataStructureSignalSeries
    .DataType(fpDataComponentY) = fpDataTypeFloat64
    .DataType(fpDataComponentX) = fpDataTypeFloat64
    .NumberOfColumns = 3
    .NumberOfRows = 100
    .FillColumns "(NumberOfRows(i), FloatingPoint64 0, FloatingPoint64 0)", _
           fpDataComponentY
    .FillColumns "(NumberOfRows(i), FloatingPoint64 0, FloatingPoint64 1)", _
           fpDataComponentX
    .Update
End With
```

An important aspect in creating the data set efficiently is to apply the properties DataStructure, DataType, NumberOfRows and NumberOfColumns in the correct sequence. Since the data set is restructured with every instance of write access to one of these properties, this data set should only reach its full size when it is last accessed. The wizard uses the correct sequence. First, the data structure of the data set is determined. Now the components of the data set are available, and their data types can be determined. Finally, the number of columns is specified. The data set only consists of one row containing three columns, i.e. three values. Once the number of rows is specified, the data set grows to include 300 values. Finally, the contents of the columns are initialized. This step can be omitted if the data set is to be filled with data at a later time.

Reading and Writing the Data of a Data Set

If you use the Value method of the DataSet object without specifying arguments you can access the complete content of the data set. This approach is recommended for smaller data sets only, since the complete contents have to be transferred into a variant array, i.e. the main memory. If the data set has a composite data structure, Value passes all components as the signal object. If you want to assign a variable in

Visual Basic, you have to use the Set statement. The following example reads a signal and then calculates the Delta-t from the X component.

```
Dim S As Signal

Set S = SignalDataSet.Value

Dim Delta As Double

Delta = S.X(1) - S.X(0)
```

Please note that the first X value in S.X has the index zero. During write-access to the Value property, you also have to consider that the value with the index zero is contained in the array.

```
Dim V(2) As Double

V(0) = 1

V(1) = 2

V(2) = 3

DataSet.Value = V
```

Thus, an array that is dimensioned with (n) in Basic has n + 1 values!

FlexPro can effectively process data sets of several gigabytes in size. This is possible because data sets that are a certain customizable size or larger are managed as files on the hard disk and therefore do not have to be loaded completely into main memory. When accessing the data of a data set via the Automation Object Model, the data to be transferred are always moved as a variant array and therefore copied completely into main memory. With very large data sets, you should therefore only access parts of the data. By using the optional arguments Row and Column of the Value property, you can directly access individual values, rows or columns in a data set.

The following example accesses the values in a rectangular area in the Y component of a signal series and sets these to zero.

```
Dim Row, Col As Long

For Row = 1 To 4

    For Col = 1 To 2

        SignalSeriesDataSet.Value(fpDataComponentY, Col, Row) = 0

    Next

Next
```

You can also transfer data block by block. To do this, you can use the ValueObjectRange object, which you can access using the Range property of the data set. This enables read/write access to a section of the data in the data set.

The following example copies the first five values in the Y component of a signal to the positions 6 to 10.

```
SignalDataSet.Value(fpDataComponentY, , 6, , 10).Value = _
    SignalDataSet.Range(fpDataComponentY, , 1, , 5).Value
```

Please note the following:

- Read and write access to the ValueObjectRange object are passed on directly to the data set. The ValueObjectRange object does not keep a copy of the data in the section.
- The ValueObjectRange object can also be used for formulas, data link objects and analysis objects. In this case, however, only read-access is allowed.
- By using the Value property of the ValueObjectRange object, not only can you access the complete range, but also columns, rows or individual values as well.
- The sequential indices of the ValueObjectRange object always start with One, even if the range in the data set starts with a higher index.

```
Dim Range As ValueObjectRange
Set Range = SignalSeriesDataSet.Range(fpDataComponentY, 1, 2, 2, 5)
Dim Data As Variant
Data = Range.Value 'Data is now of type variant/double(0 to 1, 0 to 3)
```

Transferring Large Data Sets Section by Section

Saving data that is generated continually as a data set, e.g. during a measuring process, is very common. The obvious solution for this problem is to create the data set with the correct data structure and data type and to initialize the number of rows with zero. Upon receiving the data, the number of rows is then increased; a ValueObjectRange object is used to enter the most recent value into the data set. For a data series, the code for this procedure looks as follows:

```
With DataSeries
    .NumberOfRows = .NumberOfRows + 1
    .Value( , , .NumberOfRows) = NewValue
End With
```

The disadvantage of this procedure is, however, that the processing speed is comparatively low. With every increase of the data set, FlexPro has to copy the values internally, which is very expensive, particularly with larger data sets. Furthermore, the data volume transferred per each access to the Automation interface of FlexPro is very low. You can avoid these problems by first gathering the generated data in an array and then transferring the content of the array as a whole.

```
Dim Data(1000000 - 1) As Double
For i =  0 to 100000 - 1
    Data(i) = NewValue
Next
With DataSeries
    .NumberOfRows = .NumberOfRows + 1000000
    .Range(fpDataComponentY, , .NumberOfRows - 1000000 + 1, _
           , .NumberOfRows).Value = Data
End With
```

Communicating with Other Applications

Apart from working with FlexPro data, you might want your application to exchange data with other applications, such as Microsoft Excel. You can communicate with other applications by using Automation (formerly OLE Automation).

Automating FlexPro From Within Another Application

With Automation, you can return, edit and export data by addressing objects, properties and methods of another application. Application objects that can be addressed by another application are called Automation objects.

To make FlexPro available to another Automation application, the first thing you have to do is create a reference to the FlexPro Application object. In Visual Basic, you use the function CreateObject or GetObject to return a reference to the FlexPro Application object. In a Microsoft Excel procedure, for example, you could use the following statement.

```
Set app = CreateObject("FlexPro.Application")
```

This statement makes the Application object in FlexPro available for Automation. You can control FlexPro by using the objects, properties and methods of the FlexPro Application object. The following statement, for example, creates a new FlexPro database.

```
app.Databases.Add
```

The function CreateObject starts a FlexPro session that is not closed by Automation if the object variable addressing the Application object becomes invalid. If the object reference is set to the keyword Nothing, FlexPro is not closed. Instead, use the

Quit method to close the FlexPro application. In the following example, the FlexPro start path is shown. The Quit method is used to close the new instance of FlexPro after the start path has been displayed.

```
Set app = CreateObject("FlexPro.Application")

MsgBox app.Path

app.Quit
```

Automating Another Application From Within FlexPro

If you want to use Automation from within FlexPro to exchange data with another application, the function CreateObject or GetObject provides a reference to the application. Then, use the objects, properties and methods of the other application to add, change or delete data. When you have finished making your changes, close the application. In the following FlexPro example, the Microsoft Excel start path is displayed. You can use the Set statement with the Nothing keyword to delete an object variable. This has the same effect as closing the application.

```
Set myobject = CreateObject("Excel.Application")

MsgBox myobject.Path

Set myobject = Nothing
```

For information on referencing an object library and using the objects it contains, go to Understanding Automation and Set a Reference to a Type Library.

Programmatic OLE Identifiers

You can use a programmatic OLE identifier (sometimes also called ProgID) to create an automation object. The following tables show the programmatic OLE identifiers for the ActiveX controls and FlexPro.

ActiveX controls

If you want to create the ActiveX controls listed in the following table, use the corresponding programmatic OLE identifier.

Control	Identifier
CheckBox	Forms.CheckBox.1
ComboBox	Forms.ComboBox.1
CommandButton	Forms.CommandButton.1
Frame	Forms.Frame.1

Control	Identifier
Image	Forms.Image.1
Label	Forms.Label.1
ListBox	Forms.ListBox.1
MultiPage	Forms.MultiPage.1
OptionButton	Forms.OptionButton.1
ScrollBar	Forms.ScrollBar.1
SpinButton	Forms.SpinButton.1
TabStrip	Forms.TabStrip.1
TextBox	Forms.TextBox.1
ToggleButton	Forms.ToggleButton.1

FlexPro

To create the FlexPro objects listed in the following table, use the corresponding programmatic OLE identifier. If you use an identifier without the version number suffix, you will create an object of the latest FlexPro version available on the computer on which the macro is executed.

Object	Identifier
Application	FlexPro.Application, FlexPro.Application.14
Database	FlexPro.Database, FlexPro.Database.14
SearchConstraints	FlexPro.SearchConstraints, FlexPro.SearchConstraints.14

Working with Events

Using Events with the Application or Databases Object

To create an event routine for an event affecting the Application or Databases object, carry out the following three steps:

1. Declare an object variable in a class module so that the events will be answered.
2. Write the specific event procedures.
3. Initialize the declared object from another module.

Declaring the Object Variable

Before you can write procedures for the Application or Databases object events, you have to create a new class module and declare an object of the Application or Databases type with events. Let us assume that a new class module called "EventClassModule" is created. The new class module contains the following code:

```
Public WithEvents App As FlexPro.Application
```

or

```
Public WithEvents Databases As FlexPro.Databases
```

Writing Event Procedures

After the new object with events has been declared, it appears in the Object drop-down list box in the class module, and you can write event procedures for the new object. (If you select the new object in the "Object" list, the valid events for the object are listed in the Procedure drop-down list box.) Select an event from this list. An empty procedure is added to the class module.

```
Private Sub App_DatabaseModified(ByVal Database As Object)

End Sub
```

or

```
Private Sub Databases_BeforeDatabaseClose(ByVal Database As Object)

End Sub
```

Initializing the Declared Object

Before the procedure can run, a connection has to be established between the declared object in the class module ("App" or "Databases" in this example) and the Application or Databases object. This connection can be set up using the following code from any module:

```
Dim X As New EventClassModule

Sub Register_Event_Handler()

    Set X.App = FlexPro.Application

    Set X.Databases = FlexPro.Databases

End Sub
```

Start the "Register_Event_Handler" procedure. Once the procedure has been executed, the App object in the class module references the FlexPro Application object and the database object with regard to the Databases object. The event procedures in the class module are then called as soon as the events occur.

Using Events with the Database Object

The Database object includes support for the following events: DatabaseSaved, DatabaseCreated and DatabaseOpened. Procedures, which respond to these as well as the other available events, are written into the ThisDatabase class module (open the Procedure drop-down list box to view the complete list of events). You can create an event procedure by following these steps:

1. Double click on ThisDatabase in your template project or database project in the "Project Explorer" window. (In the folder view, ThisDatabase can be found in the FlexPro Objects folder.)
2. Select Database from the Object drop-down list box.
3. Select an event from the Procedure drop-down list box.

An empty sub-routine is added to the class module.

4. Add the Visual Basic instructions to be executed when the event occurs.

The following example shows a DatabaseCreated event procedure in a template project, which is executed when a new database is created.

```
Private Sub Database_DatabaseCreated(ByVal Database As Object)
    MsgBox "New database created"
End Sub
```

The following example shows a DatabaseModified event procedure in a database project that runs only after the database has been changed.

```
Private Sub Database_DatabaseModified(ByVal Database As Object)
    MsgBox "Database modified"
End Sub
```

If an auto-macro 587 exists in a database and template database, only that automatic macro, which is stored in the database, is executed. If an event procedure for a Database event exists in a database and template database, both event procedures are executed.

For information on the creation of event procedures for the Application or Databases object, go to Using Events with the Application or Databases Object 580.

Using Events with FlexPro Objects

To create an event routine for a FlexPro object event, perform the following three steps:

1. Declare an object variable in a class module so that the events will be answered.
2. Write the specific event procedures.
3. Initialize the declared object from another module.

Declaring the Object Variable

Before you can write the procedures for the events of a FlexPro object, you need to create a new class module and declare an object of the desired type (e.g. FpObject) with events. Let us assume that a new class module called "EventClassModule" is created. The new class module contains the following code:

```
Public WithEvents oObject As FpObject
```

Writing Event Procedures

After the new object with events has been declared, it appears in the Object drop-down list box in the class module, and you can write event procedures for the new object. (If you select the new object in the Object list, the valid events for the object are listed in the Procedure drop-down list box.) Select an event from this list. An empty procedure is added to the class module.

```
Private Sub ObjectoObject_ObjectModified(ByVal Object As Object)

End Sub
```

Initializing the Declared Object

Before the running the procedure, a connection has to be established between the declared object in the class module "oObj" and the FpObject object. This connection can be set up using the following code from any module:

```
Dim X As New EventClassModule

Sub Register_Event_Handler()

    Set X.oObject = ActiveDatabase.RootFolder.Object("Formula.FML")

End Sub
```

Start the "Register_Event_Handler" procedure. Once the procedure has been executed, the oObj object in the class module references the desired FpObject object. The event procedures in the class module are then called as soon as the events occur.

Using Events with Class Objects

To create an event routine for an event affecting all FlexPro objects of a certain type, the event procedure has to be entered into one of the object modules AnyCursorObject, AnyDocObject, AnyFolder, AnyFormula, AnyFpObject or AnyValueObject. You can create such an event procedure by following these steps:

1. For instance, double-click on AnyFpObject in your template project or database project in the "Project Explorer" window. (In the folder view, AnyFpObject can be found in the FlexPro Objects folder.)
2. Select FpObjectClass from the Object drop-down list box.
3. Select an event from the Procedure drop-down list box.

An empty sub-routine is added to the class module.

4. Add the Visual Basic instructions to be executed when the event occurs.

The following example shows a ObjectModified event procedure in a template project, which is executed if an FpObject object is changed.

```
Private Sub FpObjectClass_ObjectModified(ByVal Object As Object)

    MsgBox Object.Name & " was modified."

End Sub
```

If an event procedure for a FpObject event exists in a database and template database, both event procedures are executed.

You can find out whether a certain FlexPro object belongs to one of the object classes AnyCursorObject, AnyDocObject, AnyFolder, AnyFormula, AnyFpObject or AnyValueObject in Overview of FlexPro's Object Hierarchy 565.

Working with Macros

Recording a Macro to Generate Code

If you are not sure which Visual Basic method or property to use, you can activate macro recording and execute the process manually. Macro recording converts your actions to Visual Basic code. Once your actions have been recorded, you can adapt the code to your needs. If, for example, you are unsure which property or method is used to activate the cursors in the current diagram, do the following:

1. Click Developer[Code] > Record Macro.

2. If necessary, change the default name for the macro and click OK to start recording.
3. Click on 2D Diagram Design [Cursors] > On/Off.
4. Click Developer[Code] > Stop Recording.
5. Click Developer[Code] > Macros.
6. Select the newly recorded macro from the list and click Edit.

Take a look at the Visual Basic code that sets the property corresponding to the activation state of the cursors. Position the cursor on CursorActive and press F1, or click Help. Within a topic, you can then view examples and objects that support the CursorActive property (click on Applies to).

Remarks Recorded macros often use the Selection object to manipulate the elements selected. For example, the following statement carries out a recording when you assign the color red to the selected curve of a diagram.

```
ActiveDatabase.ActiveObject.Selection.LineFormat.Color = fpColorRed
```

If you do not want the recorded code to use the Selection object, you have to edit the object using its Properties dialog box while recording a macro. In this case, the following code is recorded:

```
ActiveDatabase.ActiveObject.Curves(1).Line.LineFormat.Color = fpColorRed
```

You can also edit the recorded macro manually. You can find information on how to do this under Revising Recorded Visual Basic Macros 585.

Revising Recorded Visual Basic Macros

Macro recording is very useful when trying to find the Visual Basic methods and properties that you would like to use. If you do not know which properties and methods to use, activate macro recording and execute the procedure manually. Macro recording compiles your actions into Visual Basic code. There are, however, some restrictions to recording macros. The following cannot be recorded:

- Conditional branches
- Variable assignments
- Loop structures
- Custom forms
- Error handling
- Customizing the FlexPro user interface

To enhance your macros, you might want to edit the code that was recorded in your module.

Removing the Selection Property

Macros created through macro recording often depend on the selection you make. At the beginning of most recorded macros, you will see "Selection". Recorded macros use the Selection property to return the Selection object. In the following example, a line is added to a data set by entering the value 0 into the gray field in the editor beneath the last line of the data set.

```
Sub Macro1()
    ActiveDatabase.ActiveObject.Selection.ActiveValue = 0#
    ActiveDatabase.ActiveObject.Selection.Move fpDataSelectionDown
End Sub
```

Although this macro executes the task, it has the disadvantage that it only runs correctly if the data set is currently being displayed in the data editor and the entry field beneath the last line is selected. These two problems can be solved by editing the macro in such a way that it does not use the Selection object. Here is the revised macro:

```
Sub MyMacro()
    ActiveDatabase.ActiveObject.NumberOfRows = _
        ActiveDatabase.ActiveObject.NumberOfRows + 1
    ActiveDatabase.ActiveObject.Value(fpDataComponentAll, 1 _
        , ActiveDatabase.ActiveObject.NumberOfRows) = 0#
End Sub
```

The first statement adds a line to the active data set. The second statement uses the Range method to return a Range object that precisely represents the newly inserted value. By assigning the value 0 to the Value property of this Range object, the newly inserted value is initialized. For more information on how to use the Range object, go to Working with Data Sets 574.

Using With...End With

Macro statements that refer to the same object can be simplified by using the structure With...End With. For example, the following macro was recorded when the selected text was changed to red, a bold font and a 10-point font size.

```
Sub Macro1()
```

```
ActiveDatabase.ActiveObject.Selection.Font.Bold = True
ActiveDatabase.ActiveObject.Selection.Font.Size = 10
ActiveDatabase.ActiveObject.Selection.LineFormat.Color = fpColorRed
End Sub
```

The Selection property is used with every statement to return a Selection object. The macro can be simplified in such a way that the Selection property is used only once.

```
Sub MyMacro()
With ActiveDatabase.ActiveObject.Selection
    .Font.Bold = True
    .Font.Size = 10
    .LineFormat.Color = fpColorRed
End With
End Sub
```

Auto Macros

By giving the macro a specific name, you can run it automatically when you carry out certain operations (e.g. launching FlexPro or opening a database). FlexPro recognizes the following names as auto macros:

Macro Name	Execution
AutoExec	When launching FlexPro or loading a template database
AutoNew	When creating a new database
AutoOpen	When opening an existing database
AutoClose	When closing a database
AutoExit	When exiting FlexPro or closing a template database

Auto macros are recognized in code modules if these contain a procedure that is named after the auto macro. Just like all other macros, auto macros can be saved in the personal template database or in a project database.

In case of a naming conflict (i.e. several auto macros have the same name), an auto macro in the active project database takes precedence over a macro of the same name in the personal template database.

Remarks If you hold down the SHIFT key, auto macros are not executed. If, for example, you are creating a new database and the personal template database contains an AutoNew macro, you can prevent this AutoNew macro from being executed by clicking on File > New while holding down the SHIFT key until the new database is displayed.

Storing Values When a Macro is Finished

When a macro ends, the values contained in its variables are not automatically saved to a data carrier. If a value is to be preserved, it has to be saved outside of the macro before the macro finishes running.

Each FlexPro object in a FlexPro database, such as a data set, diagram or folder, has a Parameters list that can take any number of parameters. When working with FlexPro, you can edit this list of parameters by right-clicking with your mouse on an object, selecting Properties and then under the Properties dialog box, switching to the Parameters tab.

You can also use a macro to add parameters to an object, assign a different value to an existing parameter or retrieve the current value. In the following example, a parameter is added to the root folder of the active database.

```
ActiveDatabase.RootFolder.Parameters.Add "Charge", 54700
```

In the following example, the existing parameter is retrieved.

```
charge = ActiveDatabase.RootFolder.Parameters("Charge")
```

The following example assigns a different value to an existing parameter.

```
ActiveDatabase.RootFolder.Parameters("Charge") = 58000
```

Examples

A Custom Import Filter

This example implements a complete import filter for a binary file format. The resulting import filter supports virtually all options available for realizing custom import filters via Automation. It was specifically designed to be able to serve as the framework for your own import filters.

To keep the example simple, it does not include error handling.

The Format to be Imported

The format to be imported is a binary format that was generated for illustrative purposes but features a common pattern. A program that can produce files in the sample format, and that was used to create the Demofile.tst sample file, is located as C++ source code in the same subfolder as the sample ImportFilter.fpd database and sample file. The project database path name is usually C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples\VBA\ImportFilter\ImportFilter.fpd or C:\Users\Public\Public Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples\VBA\ImportFilter\ImportFilter.fpd.

You can implement a custom FPScript function in FPScript as well. See Custom FPScript Functions Tutorial.

The binary files consist of three parts:

- a file information structure that includes the number of channels and the sampling rate. Here is how it is defined in Visual Basic:

```
Private Type FileHeader
    strID As String * 8
    nVersion As Integer
    strOrigin As String * 32
    nNumberOfChannels As Long
    nNumberOfSamples As Long
    fSamplingRate As Double
    nTrigger As Long
End Type
```

- a number of channel header structures corresponding to the number of channels that describe the included channels:

```
Private Type ChannelHeader
    strName As String * 8
    strDescription As String * 64
    strUnit As String * 8
End Type
```

- this is followed by the data, first all the data from the first channel, then the data from the second channel and so on.

Import Filter

To realize an import filter, a class module has to be created in a database; the module implements the IImportFilter interface (in the example: class module DemoImportFilter). By entering

```
Implements IImportFilter
```

in the code window of the class module, the entry IImportFilter becomes available in the object window. If this entry is selected, the two procedures of the interface IImportFilter are listed in the Procedure window. Once the two interface procedures have been selected in the Procedure window, the corresponding procedure roots are added to the Code window, including the correct arguments.

ImportSpy Procedure

Once a user has started an import procedure by calling the Import dialog box and selecting a file, the ImportSpy procedure of all import filters registered with FlexPro is initiated in order to find out which import filters can import the file.

In the example, IImportFilter_ImportSpy first checks which filter was specified. This step only takes place for the sake of thoroughness. In this case it is not imperative because only one format has been registered for the import filter.

```
...
If Filter = m_strFilter Then
...
```

The filter verifies whether the name extension of the file corresponds to the one used for registering the filter:

```
...
If Right(UCase(PathName), 4) = ".TST" Then
...
```

The file is not opened and the file header is not read until the verification yields a positive result:

```
...
Open PathName For Binary Access Read As #1 Len = Len(TheFileHeader)
Get #1, , TheFileHeader
Close #1
TheFileHeader.strID = CutString(TheFileHeader.strID)
If TheFileHeader.strID = "DEMO    " And TheFileHeader.nVersion = 1 Then
    IImportFilter_ImportSpy = True
End If
...
```

The example file format contains an identifying text and a version number that help to identify the format. If the identifying text is correct and the file format features a version that is supported, the function will return True.

If additional specific import filters that support this file format exist, a selection dialog box appears during import in which the user can select the import filter to be used. FlexPro then calls the Import procedure of the import filter in order to import the file.

Import Procedure

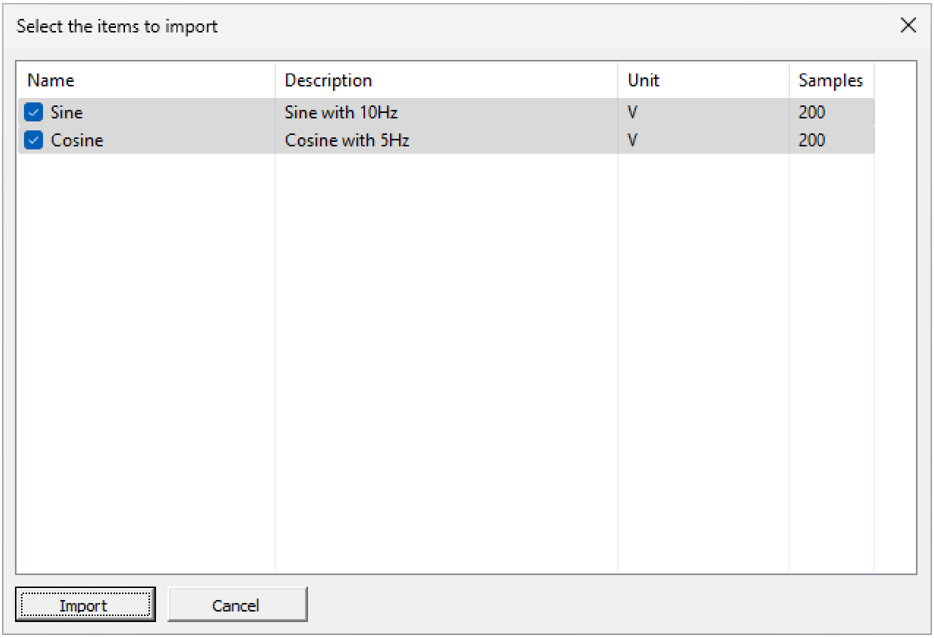
The Import procedure is only called if the ImportSpy procedure of the import filter has been called successfully.

As with the ImportSpy procedure, the first step is a verification of the filter. Then, the file header of the specified file as well as the header information for all elements or channels are read:

```
...
Open PathName For Binary Access Read As #1
Get #1, , TheFileHeader
For i = 1 To TheFileHeader.nNumberOfChannels
    Get #1, , TheChannelHeader
    Set oImportItem = New ImportItem
    oImportItem.strName = CutString(TheChannelHeader.strName)
    oImportItem.strDescription = _
            CutString(TheChannelHeader.strDescription)
    oImportItem.strUnit = CutString(TheChannelHeader.strUnit)
    oImportItem.nSamples = TheFileHeader.nNumberOfSamples
    oItemColl.Add oImportItem
Next i
nDataStartPos = Seek(1) - 1
Close #1
...
```

The elements found are stored in a Collection object together with all the data relevant for presentation in the dialog box or for import. If the file is automatically imported in its entirety, a dialog box appears where the user can choose the elements to be imported.

```
...
If (Flags And fpImportOptionAutomatic) = 0 Then
    Set oFrm = New ImportItemsSelectFrm
    oFrm.InitAndShow oItemColl, Flags
    bCancel = oFrm.m_bCancel
    Unload oFrm
End If
...
```



Eventually, the selected elements are then imported, taking into account the Flags parameter. If a new subfolder is to be created for each file to be imported, these subfolders are created first:

```
...
If Flags And fpImportOptionSubfolder Then
    Set oFile = oFS.GetFile(PathName)
    Set oImportFolder = Folder.Add(Left(oFile.Name, InStrRev(oFile.Name, ".")_
                        - 1), fpObjectTypeFolder)
```

```
Else    '   import into the given folder
    Set oImportFolder = Folder
End If
...
```

If the data are not to be imported as signals, an X data set is created first; it can be referenced by all further data sets as an X component. In the example, there is only one sampling interval and one trigger, so that the X component can be calculated using a formula:

```
...
If (Flags And fpImportOptionSignal) = 0 Then
    Set oXItem = oImportFolder.Add("XItem", fpObjectTypeFormula)
    oXItem.Origin = TheFileHeader.strOrigin
    oXItem.Component = fpDataComponentX
    oXItem.Formula = "(" & CStr(TheFileHeader.nNumberOfSamples) & ", " & _
        CStr(-(TheFileHeader.nTrigger * oBinaryDataLink.SamplingInterval)) _
         & ", " & CStr(1 / TheFileHeader.fSamplingRate) & ")"
End If
...
```

Lastly, all selected Y components in the file are imported. In the example, this can be done by taking advantage of the capabilities FlexPro binary import offers. For every channel to be imported, a BinaryDataLink object is created and parameterized according to the specifications. Lastly, the procedure decides whether the object created is evaluated into a data set (the flag fpImportOptionLink was set) or whether the BinaryDataLink object is preserved as a link.

```
...
For i = 1 To oItemColl.Count
    If oItemColl(i).bSelected = True Then
        Set oBinaryDataLink = oImportFolder.Add(oItemColl(i).strName_
             , fpObjectTypeBinaryDataLink)
        '   general properties
        oBinaryDataLink.CommentsY = oItemColl(i).strDescription
        oBinaryDataLink.Origin = TheFileHeader.strOrigin
        oBinaryDataLink.Author = Application.UserName
        '   import properties
```

```
        oBinaryDataLink.FilePath = PathName
        oBinaryDataLink.NumberOfBlocks = 1
        oBinaryDataLink.BlockSize = 200
        oBinaryDataLink.ByteDistance = 0
        oBinaryDataLink.ResultDataType = fpBinaryDataLinkResultDataTypeFloatingPoint64
        oBinaryDataLink.DataType = fpBinaryDataLinkDataTypeFloatingPoint64
        oBinaryDataLink.ByteOffset = nDataStartPos + ((i - 1)_
                                     * TheFileHeader.nNumberOfSamples * 8)
        '   as signal ?
        If Flags And fpImportOptionSignal Then
            oBinaryDataLink.AsSignal = True
            oBinaryDataLink.SamplingInterval = 1 / TheFileHeader.fSamplingRate
            oBinaryDataLink.SamplingOrigin = -(TheFileHeader.nTrigger_
                                             * oBinaryDataLink.SamplingInterval)
        Else    '   assign the X component created above to this item
            oBinaryDataLink.Component = fpDataComponentY
            oBinaryDataLink.AssignedX = oXItem.Name
        End If
        '   if import action is copy than evaluate the binary data link
        If (Flags And fpImportOptionLink) = 0 Then
            oBinaryDataLink.Evaluate
        End If
    End If
Next i
...
```

If the import has been completed successfully, the procedure will return True.

Registering an Import Filter

For a customized import filter to be available in FlexPro's Import dialog box (the file format is listed under File Type, it has to be generated using

```
Dim oImportFilter As New DemoImportFilter
```

and then be registered with the FlexPro Application object by calling RegisterImport.

```
RegisterImport oImportFilter.m_strFilter, _
        fpImportOptionSpecific Or _
        fpImportOptionLink Or fpImportOptionNoLink Or _
        fpImportOptionSubfolder Or fpImportOptionNoSubfolder Or _
        fpImportOptionAutomatic Or fpImportOptionManual Or _
        fpImportOptionSignal Or fpImportOptionNoSignal Or _
        fpImportOptionNoCalendarTime, _
        oImportFilter
```

During registration, the import filter itself is referenced and specifies which of the options available in the Import filter dialog box are supported.

If, for example, an import filter does not support links, only the fpImportOptionNoLink flag is specified in the first argument of RegisterImport. The Create links option will be grayed out and will not be set if this import filter is selected in the Import dialog box. If only fpImportOptionLink is specified, the option Create links is also gray, but activated. If both options are supported (fpImportOptionLink or fpImportOptionNoLink), the user can freely choose how the import will take place.

This procedure also applies to the following options:

- Create new folder for each file
- Import with absolute time
- Import as signals

If an import filter is to be available in all databases of one user or to all users in general, it is recommended that you implement it in the personal template database and to register it in the procedure AutoExec, which is executed automatically when the template database is loaded. De-registration can then be handled by the AutoExit procedure (see also Auto Macros 587).

Note: When developing a customized import filter, please consider the following: When a VBA project is reset, such as due to code changes in the debugger, previously registered import filters of this project will no longer be called. The import filter has to be de-registered and re-registered again.

A Custom FPScript Function

This example implements a custom FPScript function that can be used in FlexPro FPScript formulas like a built-in function after registration. It is supported in the wizard of the FPScript Editor and is automatically available in all loaded databases.

Notes

The example shows how to implement this type of function in VBA. The complete VBA source text for the example is in the CustomFunction.fpd project database. However, you can also use any other automation-compatible programming language, such as C# or C++. A variant of the example that has been developed as an add-in in C++, is also included with FlexPro. The project database path name is usually C:
\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%
>\Examples\VBA\CustomFunction\CustomFunction.fpd or
C:>Users>Public>Public Documents>Weisang>FlexPro><%
VERSION_COMMERCIAL%
>>Examples>VBA>CustomFunction>CustomFunction.fpd.

You can implement a custom FPScript function in FPScript as well. See Custom FPScript Functions Tutorial.

Background

The example defines a function with three arguments that adds the first two arguments together and dependent on the third argument. It is constructed from several built-in functions. The example covers the essential elements for implementing a custom FPScript function:

- Defining arguments
- Argument type and structure restrictions
- Using default arguments
- Registering and unregistering the function
- Accessing the arguments after the function call
- Defining custom FPScript constants

Registering a Custom Function

Before you can register a function, you have to define its arguments and other properties. The registration itself then appears at the end of the function definition.

First, the function is added to the CustomFPScriptFunctions collection. You should then assign it a description. This is then automatically displayed when the wizard window is used, for instance.

```
With CustomFPScriptFunctions.Add("MyFunction")
     .Description = "Adds or subtracts two values"
     .Indeterministic = False
     …
End With
```

The Indeterministic property defines whether the function always returns the same result for the same input. If Indeterministic is set to True, formulas that use this function have to be calculated at each update interval. Therefore, this value should be set to True only if absolutely required, such as when there are changing external dependencies.

Once the function has been added to the CustomFPScriptFunctions collection, the required arguments can be added. Only the name of the argument is required, all other settings are optional. However, it is recommended that you provide a description so that a note regarding the argument can be displayed in the wizard window. In addition, the argument data types and structures are usually limited to the exact degree required. The advantage of this procedure is that FlexPro already handles checking for conformance with data type and structure restrictions before the custom FPScript function routine is called. This makes it possible to limit the amount of code necessary for implementing the function and allows FlexPro to handle the output of error messages.

```
With .Parameters.Add("Arg1")
     .Description = "First argument"
     .AllowedTypes = fpParameterTypeNumeric
     .AllowedStructures = fpParameterStructureScalar Or _
        fpParameterStructureDataSeries Or fpParameterStructureSignal
End With
```

In the case above, the first argument is limited to numeric data types and scalar values, data series and signals are permitted as data structures. Data types and structures can be linked for the assignment with the OR operator as shown. If the argument deviates from the specified restrictions when the function is used, an error message will appear.

The default value is used for the last argument of the function. This means that the argument can be omitted. In this case, the default value for this argument can be used internally.

```
With .Parameters.Add("Operation")
      .Description = "Type of operation"
      .AllowedTypes = fpParameterTypeNumeric
      .AllowedStructures = fpParameterStructureScalar
      .DefaultValue = "MYFUNC_OPERATION_ADD"
End With
```

The case above has a unique feature. The type of the Operation parameter is numeric, but text is specified as the default value. The text contains the name of a previously included custom FPScript constant whose value is determined automatically as a default value in this case. The advantage of this type of definition is that the more accessible name of the constant is displayed in the wizard window.

Custom FPScript constants can be inserted as follows:

```
With CustomFPScriptConstants
      .Add "MYFUNC_OPERATION_ADD", "Selects the add operation", 1
      .Add "MYFUNC_OPERATION_MIN", "Selects the subtract operation", 2
End With
```

A new element is simply added to the CustomFPScriptConstants collection by specifying a name, a description and a scalar value. Conflicts with the names of existing built-in or custom constants will result in a relevant error message.

After the function parameters are declared, you can register the function for use. The Register method is called for this purpose. As an argument, it receives a reference to an object that implements the ICustomFunctionCalculate interface.

```
.Register oMyFunction
```

The ICustomFunctionCalculate interface contains the Calculate method, which is called by FlexPro when calculating a formula that uses MyFunction, provided that the parameter restrictions are met.

Notes
Only after registering a function are the defined parameters checked for consistency. If an error occurs during this check, you should review the code that defines the parameters for problems.

Typically, custom FPScript functions are registered in an automatically executed function like AutoOpen or AutoExec so that they are available at a defined time early on.

Unregistering is the opposite of registering a function, so that usually in AutoOpen or AutoClose, the following occurs:

```
Sub AutoClose()
    '   unregister function
    CustomFPScriptFunctions.Item("MyFunction").Delete

    '   unregister constants
    With CustomFPScriptConstants
        .Item("MYFUNC_OPERATION_ADD").Delete
        .Item("MYFUNC_OPERATION_SUB").Delete
    End With
End Sub
```

When the function is deleted from the CustomFPScriptFunctions collection, it is automatically deregistered. Constants are unregistered in the same manner.

Note Unregistering custom FPScript elements is not absolutely necessary, since they are automatically removed once the program is closed. Since, however, just an object reference is supplied when a function is registered, explicitly unregistering it results in controlled termination of the object that executes the function calculation. This makes it possible to avoid unwanted side-effects (such as FlexPro hanging).

Using a Custom Function

When the registered FPScript function MyFunction is called, the Calculate method of the ICustomFunctionCalculate interface is called. The arguments are passed in the form of a Variance array. The implementation example demonstrates how to access arguments. Due to the possible variety of data structures and types, the example has been simplified in some spots.

The array with the arguments SafeArrayOfArguments is one-based, as is typical in VBA. You can therefore access the desired operation by using index three:

```
'   MYFUNC_OPERATION_ADD = 1
'   MYFUNC_OPERATION_SUB = 2
nOperation = SafeArrayOfArguments(3)
```

The third argument has been restricted to scalar values of the numeric type. However, a void value can still be specified for the operation to be executed. Therefore, a check is carried out:

```
'   check operation
If nOperation <> 1 And nOperation <> 2 Then
'   will be propagated as error 0x800a0002 to FlexPro
    Err.Raise 2, "ICustomFunctionCalculate_Calculate"_
               , "Invalid operation value"
End If
```

In the event of an error, an exception occurs. The message "Invalid operation value" then appears together with the specific error code either in the FlexPro Event Log or in an error dialog box.

The remaining part of the function calculation shows how you conduct case differentiations using the first argument with the help of the VB functions IsObject, TypeOf and IsArray and then execute the desired operation.

Notes on using VBA for development

- If after registering a function, its Calculate method is changed, the function has to be unregistered and then reregistered. This is necessary because otherwise the entry point for the function is no longer valid and an error message will be displayed.
- At least when testing a Calculate method, you should make sure that under File > Options on the System Settings tab, the option Update objects in the background is disabled. Otherwise, FlexPro will crash if you use breakpoints in the Calculate method. The reason for this is that problems occur in the VBA environment when calling the Calculate method in the background program thread.

- For custom FPScript functions in VBA, the runtime environment ensures that the Calculate method runs only once at a certain time. This limitation is in place for reasons of safety. For one thing, the VBA runtime environment does not support multiprocessing very well, and for another, the necessary limitation to one running instance in VBA is not possible using the language as the sole medium.

Displaying a Diagram on a Form

This example illustrates a visualization of measured or calculated low-frequency data.

Every second, a new value pair (time plus measured value) is added to a signal. After that, a diagram, which contains this signal as a curve, is updated and also updated in the form. As long as the data set with the acquired values contains less than 100 values, the complete signal is displayed. Otherwise, only the last 100 values are shown.

The example code and the corresponding FlexPro objects are stored in the Visualize.fpd database. The project database path name is usually `C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples\VBA\Visualize\Visualize.fpd` or `C:>Users>Public>Public Documents>Weisang>FlexPro><%VERSION_COMMERCIAL%>>Examples>VBA>Visualize>Visualize.fpd`.

Creating a Form for Diagram Display

The required form only contains an image control that handles the display of the diagram as well as a button for closing the form.

When the form is initialized, the random number generator is initialized.

```
Private Sub UserForm_Initialize()
    '   init random number generator (needed for the sample data)
    Randomize
    nTimerID = SetTimer(0, 0, 1000, AddressOf UpdateProc)
End Sub
Private Sub UserForm_Terminate()
    KillTimer 0, nTimerID
End Sub
```

Inserting and Updating the Diagram

In the UpdateProc procedure,

```
On Error Resume Next
```

is used to prevent the procedure from being aborted if an error occurs. Use of the SetTimer API requires this to make sure that the procedure call always returns to Windows. After that, the procedure verifies whether the target signal for the data is already available in the root folder of the database. If that is not the case, it is created and initialized; otherwise, a new row is added to the signal.

```
If oSignal Is Nothing Then
    Set oSignal = ActiveDatabase.RootFolder.Add("Signal", fpObjectTypeDataSet)
    oSignal.DataStructure = fpDataStructureSignal
    oSignal.DataType(fpDataComponentX)= fpDataTypeCalendarTime
    oSignal.DataType(fpDataComponentY) = fpDataTypeFloat64
    ' we now have a signal with one value
Else ' increase the rows of the signal
    oSignal.NumberOfRows = oSignal.NumberOfRows + 1
End If
```

The new value is then calculated and appended to the existing signal using the Range object (see also Working with Data Sets 574).

```
' you could acquire the data here, in this sample we calculate
' the new value pair
oSignal.Value(fpDataComponentX, , oSignal.NumberOfRows) = Now
oSignal.Value(fpDataComponentY, , oSignal.NumberOfRows) = (fMax - fMin + 1)_
                                                          * Rnd + fMin
```

At the end of the function, the diagram to be shown in the form is updated and the Picture property of the diagram is assigned to the Picture property of the viewing control field in the form.

```
Set oDiag = ActiveDatabase.RootFolder.Object("2D-Diagram.2D")
oDiag.Update
Display.DiagramImage.Picture = oDiag.Picture
```

FlexPro Objects Required for the Visualization

To be able to visualize the data as described above, a diagram and a formula object were created in the root folder of the database. The diagram serves as the basis for the visual presentation in the form. It contains exactly one curve consisting of a signal that provides the LastValuesOfSignal formula. The formula verifies whether the number of values in the signal is larger than the customizable maximum (100 in the example). If this is the case, the formula returns the most recent 100 values of the signal; otherwise, it returns the complete signal.

```
nMax = 100
nCount = NumberOfRows(Signal)
if nCount < nMax then
 return Signal
else
 return Signal[-nMax, -1]
end
```

Processing Events

This example shows how you can capture and respond to events in your Basic programs. The program code and the required FlexPro objects can be found in the database called Events.fpd. The path to the Project Database is usually C:\Users\Public\Documents\Weisang\FlexPro\<%VERSION_COMMERCIAL%>\Examples\VBA\Events\Events.fpd or C:>Users>Public>Public Documents>Weisang>FlexPro><%VERSION_COMMERCIAL%>>Examples>VBA>Events>Events.fpd.This database contains a diagram with two X axes. The X values of the curve are scaled above the lower axis. The X values of the curve are scaled above the lower axis. In the example, the curve represents a route measured in km. The second axis, which is shown at the top of the diagram, assigns names to certain locations along this route. In the example, these are names of cities that can be found on the route. Two data series were created to scale this axis. The Cities data series contains the names of the cities, and the Positions data series contains their locations along the route. To label the axis with this data, the Linear, division by data set axis type was selected on the Axis Scaling tab and the Positions data series was entered as a data set. This determines the points at which axis divisions are to appear. On the Division Labeling tab, the option Division labeling using a data series, signal or function was enabled and Cities was entered as a data set. Since FlexPro 7 you have been able to easily synchronize the axis end values of this second X axis with the first on the Axis Scaling by selecting the option Same as previous axis. This axis cannot be auto-scaled because no curve will be shown above it.

The following example, which was originally developed for FlexPro 6, illustrates how the end values of the second axis can be synchronized with the first using event processing. The following requirements result:

- Event processing must be activated when you open the database and reset to its former state when you close the database.
- When opening the diagram, the second axis is to be set to fixed scaling and the current end values of the first axis are to be adopted.
- When zooming in or out using the cursor, the new end values of the first X axis are to be adopted by the second.

Activating Event Processing

This task can be executed by implementing the two auto macros 587 AutoOpen and AutoClose. The AutoMacros module contains the following code:

```
Private bEnableEvents As Boolean


Sub AutoOpen()
    bEnableEvents = Application.EnableEvents
    Application.EnableEvents = True
End Sub


Sub AutoClose()
    Application.EnableEvents = bEnableEvents
End Sub
```

AutoOpen stores the current status of the application object's EnableEvents property and then event handling is enabled.

The previous state is restored in AutoClose.

Initializing Axis Scaling

This task is executed by an event procedure that handles the BeforeOpenObject event, which is sent by every FlexPro object before opening it. In the VbaProject of `Events.fpd` there is a corresponding object called AnyFpObject which represents all FlexPro objects of the database and therefore receives all events common to all FlexPro objects. The following code is assigned to this class object:

```
Private Sub FpObjectClass_BeforeOpenObject(ByVal Object As Object)
    Dim oDiag As Diagram2D
    If Object.FullName = "\Route.2D" Then
        Set oDiag = Object
        oDiag.XAxes(2).Scaling.ModeStartValue = fpAxisScalingModeFixed
        oDiag.XAxes(2).Scaling.ModeEndValue = fpAxisScalingModeFixed
        oDiag.XAxes(2).Scaling.StartValue = oDiag.XAxes(1).Scaling.CurrentStartValue
        oDiag.XAxes(2).Scaling.EndValue = oDiag.XAxes(1).Scaling.CurrentEndValue
    End If
End Sub
```

The `If` test verifies whether the correct 2D diagram has been selected. The name, database path and the file name extension for the 2D diagrams have to match. The event procedure receives this event when any database object, e.g. another 2D diagram or data set, is opened. Within the `If` test, it is clear that we are dealing with a 2D diagram and the unspecific object can therefore be assigned to a variable of the Diagram2D type. The next steps are easy to understand. At first, the starting value and the end value are set to fixed scaling. After that, the current starting and end values of the first axis are adopted. The properties CurrentStartValue and CurrentEndValue provide the values resulting from auto-scaling and not those that have been entered on the Axis Scaling tab of the axis, which are inactive in this case.

Adopting the Axis End Values After Zoom Operations

Here, a multitude of events have to be handled, since several operations affect the axis end values. You can zoom in between the cursors, zoom in on a rectangular area or scroll along the axis. All these events only apply to cursor objects, i.e. objects that may contain cursors. These are diagrams, worksheets and documents. The

AnyCursorObject class object, which receives all cursor events, represents all cursor objects. It contains the following code:

```
Private Sub AdaptScaling(Object As CursorObject)
    Dim oDiag As Diagram2D
    If Object.FullName = "\Route.2D" Then
        Set oDiag = Object
        oDiag.XAxes(2).Scaling.StartValue = oDiag.XAxes(1).Scaling.CurrentStartValue
        oDiag.XAxes(2).Scaling.EndValue = oDiag.XAxes(1).Scaling.CurrentEndValue
    End If
End Sub

Private Sub CursorObjectClass_AxisScaled(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_AxisScrolled(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_RectangleZoomed(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_ZoomedBetweenCursors(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_ZoomReset(ByVal CursorObject As Object)
    AdaptScaling CursorObject
End Sub

Private Sub CursorObjectClass_ZoomUndone(ByVal CursorObject As Object)
```

```
    AdaptScaling CursorObject
End Sub
```

AdaptScaling is an auxiliary procedure called by the different event procedures. Again, the procedure first verifies that the object to be processed does exist. The new axis end values are then copied. All event procedures with events that have an impact on axis scaling were implemented. And all events that are sent after the corresponding operation were selected so that the axis end values of the first X axis already feature the new values.

8.3 Additional Options

Remote Control of FlexPro via FPAccess

FPAccess Interface

Apart from the Automation Object Model, FlexPro offers the FPAccess interface, which allows controlling programs such as an application for data acquisition or a Visual Basic program to access the project database externally and to remote control FlexPro in a limited way. The FPAccess interface has a much simpler structure than the Automation Object Model and is targeted for efficient transfer of large volumes of data. The advantage of the FPAccess interface is that it is also available as a DLL and thus part of its functionality is available without FlexPro being installed.

The functionality of the FPAccess interface is a superset of that of FlexPro's DDE interface is therefore preferred. You can use the FPAccess interface to:

1. open FlexPro project databases,
2. manage FlexPro project databases in an Explorer dialog box,
3. Create folders in FlexPro project databases,
4. Write formulas and data sets in project databases,
5. check whether an object is present in the project database, and
6. Open, update and print objects in project databases (only in direct communication with FlexPro - see below).

The interface can open FlexPro project database files independently without FlexPro's help, and can write data to these files. If the project database file has already been opened with FlexPro, then a direct communication with FlexPro will occur. DCOM 609 allows this communication to take place across networks as well.

The project database access objects provided by FlexPro support OLE automation and direct access using COM. You can therefore also make best use of the interface with compiler languages such as C++.

Reference

Project Database Access for Acquisition Programs

The FlexPro package contains modules with which you can prepare your acquisition software for direct access to FlexPro project database files. The following programs are currently supported:

- DasyLab version 9 - 14 (2016)
- LabVIEW 2010 - 2016

The modules were tested using the most current acquisition software version number provided above. If you are using a newer software version, most likely the module will work with that version.

All components are based on the FlexPro FPAccess interface. The project database files can therefore also be opened independently, without running FlexPro. If the project database file has already been opened with FlexPro, then a direct communication with FlexPro will occur. This communication can also take place across networks. You can find more details in the documentation covering the FPAccess Interface 607.

Installation

The matching component may have already been supplied with your acquisition program. If this is not the case, then you will have to restart the FlexPro Setup program.

1. For the Setup Type, select Custom.
2. If you purchased a single user license from FlexPro, you are only allowed to install FlexPro on one computer. Therefore, select **only** the FPAccess Interface option and the Extension Modules option further down.
3. Click Next. A dialog appears for each program selected. Here, you have to confirm the software installation folder. The matching component is then copied into this folder. The individual components include documents describing how the components work and how they are set up. You can find these documents in the folder specified after the installation.

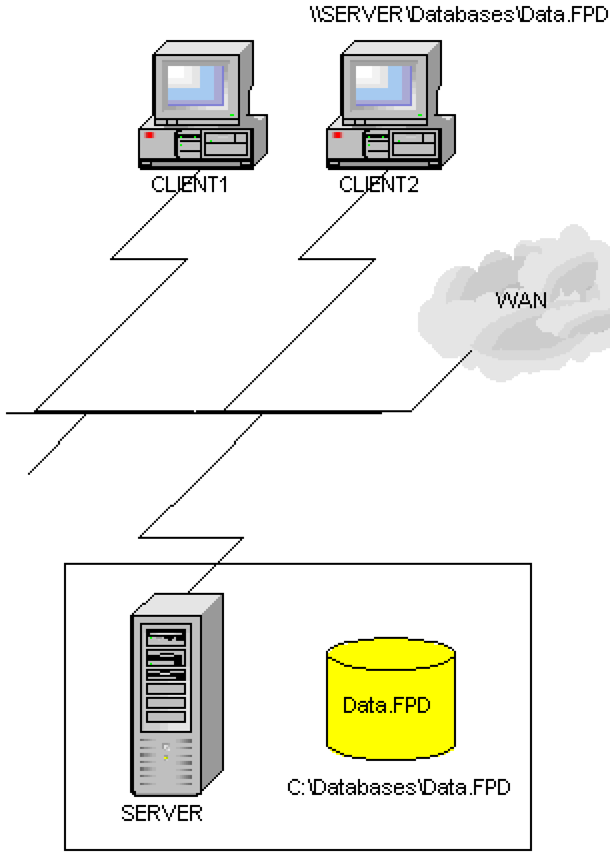
DCOM

DCOM stands for Distributed Component Object Model. DCOM is a protocol that enables software components to communicate with each other reliably, securely and efficiently over a network.

DCOM is part of Windows and is installed during the installation of the operating system.

FlexPro requires DCOM if you wish to use the FPAccess interface or one of the other supported interfaces for data acquisition software 608 based on it.

A typical DCOM application could therefore look like this:



FlexPro runs on one server and has a project database (`C:\Databases\Data.fpd`) open. One or more clients for their part open the same project database through a shared network folder (\\SERVER\Databases\Data.fpd) to write acquired data. The user on the server computer sees how the data is written into the project database and can immediately begin working with the data.

For DCOM to be used, it must first be configured. FlexPro must also be configured to use DCOM.

After following the instructions, configure DCOM on the computer that has FlexPro installed on it.

1. Click on Start.
2. In the text box Start Search enter dcomcnfg and press ENTER to start DCOM configuration.
3. In the User Account Control dialog box that appears, click on Continue.
4. Open the Component Services node and then Computers.
5. Right-click with your mouse on My Computer and then select Properties from the context menu.
6. Switch to the Default Properties tab and make sure that Enable Distributed COM on this computer is selected.
7. Click on OK, to close the dialog box.
8. Open the My Computer node and then DCOM Configuration.
9. Choose the FlexPro Project Database entry and then display its properties.
10. Switch to the General tab and choose the entry None as the Authentication Level.
11. Switch to the Security tab and make sure that the user who needs access to FlexPro from a remote system has the appropriate permissions to do so. To do this, for the launch and access permissions, select the Customize option, click on Edit and add a suitable user group.
12. Switch to the Identity tab. Select either interactive user or a specific user whose user account is to be used for running the application. If you select a specific user, you must specify this user's password.
13. Note: To test the configuration, it makes sense to set the identity to the interactive user first, since you will not be able to access the running FlexPro program otherwise.
14. Click on OK, and then on the File menu, click on Exit to exit the configuration.

- FlexPro does not have to be launched manually on the server computer. Windows launches FlexPro automatically in the background, provided FlexPro is used as the project database server.
- If you are on the client computer and want to access the project database `C:\Project Databases\Data.fpd` that is stored on the server computer called SERVER, you cannot use a network shared root (whole drive) to access the file. In this case, share the directory `C:\Project Databases` on SERVER and access it from the client via `\\SERVER\Project Databases\Data.fpd`. If you have set up the whole drive C: to be shared and want to access the project database with `\\SERVER\C\Project Databases\Data.fpd`, it will cause an access violation. This is a known issue.

Remote Control of FlexPro via DDE

DDE Interface

DDE (**D**ynamic **D**ata **E**xchange) is a dynamic exchange of data between Windows applications. A client application that initiates the exchange is required. A server application is also required to serve the requested information or run the requested actions. FlexPro can function as a server for DDE communication and can run commands that are sent from the client. A typical client would be, for instance, a Visual Basic application that accepts data from a device, saves the information in a file and remotely controls FlexPro for the evaluation and documentation of the data.

Note: FlexPro's more modern Automation interface offers much more comprehensive options for communication between FlexPro and a client application. It is recommended that you use the DDE interface.

Connection Setup, Interaction and Termination

To be able to run FlexPro functions from other Windows programs that support DDE communication as a client, you first have to set up a DDE connection. This is achieved by sending the message WM_DDE_INITIATE, where "FlexPro" or "FlexPro 14" is specified as the application and "SYSTEM" is specified as the topic. If a connection is successfully made, there is a acknowledgement by way of a WM_DDE_ACK message. If you have provided a version number, the connection will only be successful when this particular version of FlexPro is installed.

Any number of commands can then be sent to FlexPro to be run with the help of the message WM_DDE_EXECUTE. It is also possible to send several commands together with a single message. When a command is correctly received, this is confirmed by a WM_DDE_ACK message. There is also a confirmation if the commands received cannot be run correctly.

When the client application has sent all of the desired commands to FlexPro, the DDE communication is ended by sending the WM_DDE_TERMINATE message.

Visual Basic Example

Here is a brief example of how DDE communication used for printing a document can be achieved by means of a client application in Visual Basic:

```
Sub Document_Click ()
Const NONE = 0, LINK_MANUAL = 2        ' Constants
Dim Cmd, Z                              ' Variables

' Combine commands:
Cmd = "[open(""c:\Project Databases\Test.fpd"")]"
Cmd = Cmd & "[openobject(""Document.doc"")]"
Cmd = Cmd & "[update]"
Cmd = Cmd & "[print]"
Cmd = Cmd & "[closeobjects]"
Cmd = Cmd & "[close]"
If Text1.LinkMode = NONE Then
    Z = Shell("FlexPro", 4)                     ' Start FlexPro
    Text1.LinkTopic = "FlexPro 11|System" ' Set LinkTopic
    Text1.LinkItem = ""                 ' LinkItem
    Text1.LinkMode = LINK_MANUAL        ' LinkMode
End If
Text1.LinkExecute Cmd                   ' Execute Commands
End Sub
```

This subroutine can be called, for instance, if the source data on which the document is based has been overwritten by a data acquisition program.

With the commands available, it is possible to open and close project databases, to open, update, print and close an object in the project database, as well as to activate folders.

The commands listed here can, as described above, be sent to FlexPro with a WM_DDE_EXECUTE message, individually or bundled.

Note: In FlexPro it is basically possible to work with several project databases simultaneously. This is not possible in connection with DDE communication, since the project database referred to would have to be specified with every command, except when opening a project database.

9 End-User License Agreement

9.1 FlexPro End-User License Agreement

1. Definitions

Licensor: is the provider of the software and the owner of the rights to this software, Weisang GmbH (Weisang).

Licensee: is the contractual partner of the Licensor.

User: is the natural person who actually uses the Software with the consent of the Licensor and the Licensee.

Software: is the FlexPro computer program developed by Weisang GmbH, the program description and operating instructions as well as other associated written material.

License File: is a file provided by the Licensor and stored on a computer of the Licensee, which enables the execution of the Software on this computer.

License Activation: is the transfer of a License File to a computer of the Licensee. The License File is bound to the computer by electronic signature.

2. Subject of the Agreement

The subject of the contract is the Software developed by Weisang GmbH (Weisang). Weisang makes this Software available to the Licensee as per the material scope of use and time period agreed upon with Weisang.

As agreed, the Licensee will receive the Software either stored on a data carrier or USB stick or as a download via download link or as an e-mail attachment. The Licensee is not entitled to claim a disclosure of the source code.

Weisang draws attention to the fact that despite applying state-of-the-art programming technology, it is not possible to write Software that will work correctly under all circumstances. The subject of this Agreement is only a Software which can be used as intended based on its description and as specified in the user manual.

This Software has not been designed for a use in hazardous environments. Consequently, it is not suited for security or safety related applications in nuclear facilities, in aircraft GPS and communications systems, for air traffic control, weapons systems, or life-support machines.

The software supplied by Weisang is protected by copyright. This Software is a computer program capable of being protected by copyright as specified under 2 Abs.

1 Nr. 1, § 69 a German UrhG. In the relationship between the parties, Weisang is sole owner of any copyright to the Software and of any documentation provided during the negotiations leading to the conclusion and execution of this Agreement.

3. Scope of use

The right of use (hereinafter referred to as "License") for the Software is limited as follows:

3.1. License Models

a) Subscription License

The use of the software is limited to the period agreed in the purchase contract, usually one year, and also includes maintenance during this period. Maintenance includes the following services provided by Weisang:

- Free product support by e-mail or telephone to the extent agreed.
- Migration of a license to another computer.
- Free updates to the next major version number.

If no automatic renewal has been agreed, Weisang will submit an offer to the Licensee 60 days before the end of the term to continue the subscription for the agreed period. Upon acceptance of the offer and payment of the issued invoice, the term of the contract is extended by this period, starting at the end of the previous term.

If automatic renewal has been agreed, Weisang will provide a license for the next period 30 days before the end of the term. The Licensee is entitled to cancel the subscription at any time before the end of the current term, provided that the license for the following period has not yet been Activated by the Licensee. Otherwise, Weisang shall invoice the Licensee the price for the subsequent period in accordance with the currently valid price list. The term of the contract shall be extended by this period, starting at the end of the previous term.

b) Perpetual license

The perpetual license entitles the User to use the major version of FlexPro available at the time of purchase for an unlimited period of time. This license does not include maintenance and the compatibility of FlexPro is only guaranteed for the operating system specified in the system requirements at the time of purchase.

3.2. License Types

a) Single User License

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the Software on a single desktop PC at a single location. As the User, you may also install the Software on a portable computer (notebook), provided that the software is only ever used on one of these computers and only by the same User and not by other persons.

b) Shared License with Dongle

Weisang grants the Licensee the ordinary, non-exclusive right to install the Software on any number of workstation computers, provided that the Software is only used on a single computer at any one time.

c) Network License

Weisang grants the Licensee the ordinary, non-exclusive right to install the Software to any number of workstation computers located in the same local area network (LAN) on which the FlexPro network license is installed. The number of concurrent users of FlexPro may at no time exceed the number of concurrent users for which the network license was issued.

For a site network license, the physical distance of the users of these workstation computers to the location of the company branch office to which the network license is assigned may not exceed eight kilometers / five miles. Excluded are Users in the home office, whose regular workstation, however, must meet this condition, and such Users who have checked out a license for temporary offline use from the server.

In the case of a regional network license, this restriction of the radius of use applies to the region agreed in the purchase contract; usually to a specific country.

The Worldwide Network License is not subject to any such restriction on the radius of use.

3.3 Special Licenses

a) 30-Day Evaluation License

Weisang grants the Licensee the ordinary, non-exclusive and personal right to use the Software on a single computer and in a single location for a period of 30 days from the date of first use. This use shall be limited to evaluation purposes. In particular, the Licensee shall not publish or otherwise use the analyses and Visual Basic programs created with this license of FlexPro until he has purchased a Single User License or a Network License of FlexPro.

b) Reader License

Weisang grants the Licensee the ordinary, non-exclusive and personal right to install and use the Software on a single desktop PC and a portable computer (laptop). The license is restricted to read-only access to analyses, which have been created using a FlexPro Single User License or a Network License. In particular, this FlexPro license does not cover the creation of evaluations or use as a runtime environment for automated evaluations.

c) Runtime License

Weisang grants Licensee the non-exclusive right to use the Software on a single computer as a runtime environment for automated evaluations created with a FlexPro Single User License or Network License. Any interactive use of the Runtime License is prohibited. The use of the FlexPro user interface and its reproduction are not permitted. Any operation of the runtime license for server applications shall be subject to written approval from Weisang. To this end, Weisang will charge additional license fees, the amount of which will depend on the number of individuals using the server.

Any further use is prohibited.

d) Student License

Use is only permitted for registered pupils and students and restricted exclusively to activities related to personal education. Consequently, any use in research and professional environments for commercial or industrial purposes is prohibited.

e) University and Institute License

The license is granted only to schools and universities and is limited to use only at schools and universities and exclusively for teaching purposes and for non-commercial research. Thus, use in projects financed by the industry or other principals is prohibited.

f) License for Non-Profit Organizations

The license only covers members of non-profit organizations. Any use for commercial or industrial purposes is prohibited.

4. Granting of Rights

(1) In return for payment of the agreed fee in accordance with the order confirmation, the Licensee shall receive the non-exclusive, non-transferable and non-sublicensable right to use the Software for the term of the subscription license to the extent corresponding to the agreed license type (Section 3.), the license certificate and the order confirmation.

(2) The contractual use includes the installation, loading, display and running of the installed software, moving the license to another computer and the installation of updates provided by the Licensor. The type and scope of use are otherwise

determined by the agreed license type in accordance with the order confirmation and license certificate.

(3) The Licensee is entitled to make a backup copy of the data carrier provided to him. The Licensee must visibly affix the note "Backup copy" and a copyright notice of the manufacturer to the backup copy made. The backup copy must be deleted at the end of the term.

(4) It is not permitted to copy or reproduce the software, or the documentation, in whole or in part, in its original or modified form or mixed with other software.

(5) It is prohibited to modify or deface the copyright notices, further legal reservations or other characteristics serving to identify the programming contained in the Software.

(6) The Licensee is not entitled to reproduce the software beyond the cases mentioned in paragraphs 1 to 4.

(7) The Licensee is not entitled to transfer the copy of the software provided to him or any backup copy made to third parties. In particular, he is not permitted to sell, lend, rent or otherwise sublicense the software or to publicly reproduce or make the software accessible.

(8) The right to use the Software may only be transferred to a third party with the prior written consent of the Licensor and only under the obligation to comply with these conditions and to observe the scope of use in accordance with the license type. This does not include companies affiliated with the Licensee.

(9) The Licensee is prohibited from modifying or translating the Software or creating works derived from the software without the prior written consent of the Licensor. Notwithstanding the above, the creation of documentation and instructions as well as translations for the Licensee's internal use is permitted. The licensee is not entitled to reverse engineer, decompile or disassemble the software or otherwise attempt to make the source code of the software accessible. Any claim to disclosure of the source code is excluded.

(10) If the Licensee violates any of the above provisions, all rights of use granted with the order confirmation in conjunction with these terms and conditions shall immediately become ineffective and automatically revert to the Licensor. In this case, the Licensee must immediately and completely cease using the Software, delete all copies of the Software installed on its systems and delete any backup copies that may have been made.

5. Remuneration, Due Date and Default

(1) The remuneration for the granting of use, its due date and the granting of the rights of use are set out in the order confirmation.

(2) If the Licensee is in default with the payment of the remuneration, the default interest shall amount to eight percent (8%) above the applicable base interest rate.

6. Term and Termination

(1) The term of the license results from the license model agreed between the parties (Section 3.1) and the agreed license type (Sections 3.2 and 3.3) in conjunction with the order confirmation.

(2) Notwithstanding the above, the license may be terminated by either party without notice for good cause.

Good cause entitling the Licensor to terminate the contract shall be deemed to exist in particular if the Licensee infringes the Licensor's rights of use by using the Software beyond the extent permitted under the order confirmation in conjunction with the license type and these Terms and Conditions and fails to remedy the infringement within a reasonable period of time following a warning from the Licensor.

(3) Notice of termination must be given in text form.

(4) In the event of termination, the Licensee must immediately cease using the Software and remove all installed copies of the program from its computers and permanently delete the license file.

7. Indemnity In Case of a Breach of this Agreement

Weisang draws attention to the Licensee that Licensee shall be liable for any damage resulting from copyright infringement which affects Weisang as a result of the Licensee's breach of the contractual terms.

8. Infringement of Property Rights of Third Parties

(1) In the event that third parties raise claims against the Licensee due to an infringement of their property rights through the Software, and if the use of the Software is adversely affected or prohibited by such claim, Weisang will be liable as follows:

At the Licensee's election and costs Weisang will either modify or replace the Software so that the property right will no longer be infringed or hold the Licensee harmless against license fees requested by the owner of the property right or a third party.

If Weisang does not succeed in doing this under reasonable conditions, Weisang will notify the Licensee and prohibit the use of the Software in whole or in part (in the case of a module) from a specified time going forward. Weisang will then indemnify the Licensee by refunding the purchase price for the license concerned and/or the license for the affected module. Rights to further indemnities do not exist.

Weisang may only be held liable if the Licensee notifies Weisang promptly on of any claim raised by a third party, if Weisang does not acknowledge the alleged property right violation and transfers any form of dispute including out-of-court settlements to Weisang or only leads them after consultation with Weisang. Licensee's necessary court costs and reasonable attorneys' fees for the defense of such claim and incurred after such consultation with Weisang shall be borne by Weisang.

(2) In the event that Licensee discontinues using the Software in order to mitigate possible damages or for another good cause, the Licensee shall be obliged to notify such third party that this end of usage does not imply acknowledgement of the alleged infringement of property rights.

(3) Claims against Weisang shall be excluded to the extent that the Licensee is responsible for such infringement of property rights.

9. Software Protection and Audit

(1) The Licensee is obliged to take suitable measures to protect the software from access by unauthorized third parties, in particular to store all copies of the software in a protected location.

(2) At Weisang's request, the Licensee shall enable Weisang to verify the proper use of the Software, in particular whether the Licensee is using the program qualitatively and quantitatively within the scope of the licenses acquired by it. To this end, the Licensee shall provide Weisang with information upon request, grant Weisang access to relevant documents and records and allow Weisang or an auditing firm designated by Weisang and acceptable to the Licensee to review the hardware and software environment used.

(3) Weisang may carry out the inspection on the Licensee's premises during the Licensee's regular business hours or have it carried out by third parties bound to secrecy. Care must be taken to ensure that the Licensee's business operations are disrupted as little as possible. If the inspection reveals use not in accordance with the contract, the Licensee shall bear the costs of the inspection, otherwise Weisang shall bear the costs.

10. Maintenance and Warranty

(1) The Licensor warrants the maintenance of the contractually agreed quality of the Software. This warranty is valid for five years from the release of a major version. Furthermore, the Licensor warrants that no rights of third parties conflict with the contractual use of the software.

(2) The Licensor shall remedy any material defects and defects of title in the Software within a reasonable period of time. The Licensor also fulfills its obligation to rectify defects by making updates with an automatic installation routine available for download on its homepage and offering the Licensee telephone support or remote maintenance support to solve any installation problems that may arise.

(3) The Licensee is obliged to notify Weisang in writing of any defects in the software as soon as they are discovered. In the case of material defects, this shall be done by describing the time of occurrence of the defects and the more detailed circumstances.

(4) Functional impairments resulting from the hardware and software environment provided by the licensee, incorrect operation, external defective data, computer network malfunctions or other reasons originating from the Licensee's area of risk shall not be considered defects. No warranty is assumed for Software that has been modified by the Licensee, unless the Licensee proves that the modification is not the cause of the reported defect.

11. Liability of the Licensor

(1) The Licensor shall be liable as follows if the Software is used in accordance with the contract:

a) In the event of a slightly negligent breach of an obligation that is essential for achieving the purpose of the contract (cardinal obligation), the amount shall be limited to the damage that is foreseeable and typical for the type of transaction in question;

b) in the event of intent or gross negligence, in the event of injury to life, limb or health, in the event of violation of the provisions of the Product Liability Act, as well as to the extent of a guarantee assumed, without limitation.

(2) Any further liability of the Licensor, in particular liability for indirect damages of the Licensee and its loss of profit, does not exist.

(3) The above limitation of liability also applies to the personal liability of the Licensor's employees, representatives and bodies.

12. Offsetting and Form

(1) Offsetting is only permitted against undisputed or legally established claims of the Licensor.

(2) The text form is sufficient for the written form. All e-mails to the Licensor should be sent to the following e-mail address: info@weisang.com

13. Export and Import Restrictions

The software may be subject to export and import restrictions. In particular, authorization requirements may exist or the use of the software or associated technologies may be subject to restrictions abroad. The Licensee shall comply with the applicable export and import control regulations of the Federal Republic of Germany, the European Union and the United States of America.

14. Applicable Law and Place of Jurisdiction

(1) This contract shall be governed exclusively by German law to the exclusion of the United Nations Convention on Contracts for the International Sale of Goods of April 11, 1980 (UN Sales Convention).

(2) The place of jurisdiction is St. Ingbert, Germany.

Date: April 07, 2025

Weisang GmbH

9.2 Licenses of Open-Source Software Used in FlexPro

ACE and TAO Licenses

Copyright and Licensing Information for ACE(TM), TAO(TM), CIAO(TM), DAnCE(TM), and

CoSMIC(TM)

ACE(TM), TAO(TM), CIAO(TM), DAnCE(TM), and CoSMIC(TM) (henceforth referred to as "DOC software") are copyrighted by Douglas C. Schmidt and his research group at Washington University, University of

California, Irvine, and Vanderbilt University, Copyright (c) 1993-2018, all rights reserved. Since DOC

software is open-source, freely available software, you are free to use, modify, copy, and distribute--

perpetually and irrevocably--the DOC software source code and object code produced from the source, as

well as copy and distribute modified versions of this software. You must, however, include this copyright

statement along with any code built using DOC software that you release. No copyright statement needs to

be provided if you just ship binary executables of your software products.

You can use DOC software in commercial and/or binary software releases and are under no obligation to

redistribute any of your source code that is built using DOC software. Note, however, that you may not

misappropriate the DOC software code, such as copyrighting it yourself or claiming authorship of the DOC

software code, in a way that will prevent DOC software from being distributed freely using an open-source

development model. You needn't inform anyone that you're using DOC software in your software, though we

encourage you to let us know so we can promote your project in the DOC software success stories.

The ACE, TAO, CIAO, DAnCE, and CoSMIC web sites are maintained by the DOC Group at the Institute

for Software Integrated Systems (ISIS) for the development of open-source software as part of the open-

source software community. Submissions are provided by the submitter ``as is'' with no warranties whatsoever, including any warranty of merchantability, noninfringement of third party intellectual property,

or fitness for any particular purpose. In no event shall the submitter be liable for any direct, indirect, special,

exemplary, punitive, or consequential damages, including without limitation, lost profits, even if advised of

the possibility of such damages. Likewise, DOC software is provided as is with no warranties of any kind,

including the warranties of design, merchantability, and fitness for a particular purpose, noninfringement, or

arising from a course of dealing, usage or trade practice. Washington University, UC Irvine, Vanderbilt

University, their employees, and students shall have no liability with respect to the infringement of

copyrights, trade secrets or any patents by DOC software or any part thereof. Moreover, in no event will

Washington University, UC Irvine, or Vanderbilt University, their employees, or students be liable for any

lost revenue or profits or other special, indirect and consequential damages.

DOC software is provided with no support and without any obligation on the part of Washington University,

UC Irvine, Vanderbilt University, their employees, or students to assist in its use, correction, modification, or

enhancement. A number of companies around the world provide commercial support for DOC software,

however. DOC software is Y2K-compliant, as long as the underlying OS platform is Y2K-compliant.

Likewise, DOC software is compliant with the new US daylight savings rule passed by Congress as "The

Energy Policy Act of 2005," which established new daylight savings times (DST) rules for the United States

that expand DST as of March 2007. Since DOC software obtains time/date and calendaring information from operating systems users will not be affected by the new DST rules as long as they upgrade their operating

systems accordingly.

The names ACE(TM), TAO(TM), CIAO(TM), DAnCE(TM), CoSMIC(TM), Washington University, UC Irvine, and

Vanderbilt University, may not be used to endorse or promote products or services derived from this source

without express written permission from Washington University, UC Irvine, or Vanderbilt University. This

license grants no permission to call products or services derived from this source ACE(TM), TAO(TM),

CIAO(TM), DAnCE(TM), or CoSMIC(TM), nor does it grant permission for the name Washington University, UC

Irvine, or Vanderbilt University to appear in their names.

If you have any suggestions, additions, comments, or questions, please let me know.

Douglas C. Schmidt

Back to the ACE home page.

Last modified 10:39:57 CDT 09 October 2020

Apache Arrow License

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You

meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

----

src/arrow/util (some portions): Apache 2.0, and 3-clause BSD

Some portions of this module are derived from code in the Chromium project, copyright (c) Google inc and (c) The Chromium Authors and licensed under the Apache 2.0 License or the under the 3-clause BSD license:

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This project includes code from Daniel Lemire's FrameOfReference project.

https://github.com/lemire/FrameOfReference/blob/6ccaf9e97160f9a3b299e23a8ef739e711ef0c71/src/bpacking.cpp

https://github.com/lemire/FrameOfReference/blob/146948b6058a976bc7767262ad3a2ce201486b93/scripts/turbopacking64.py

Copyright: 2013 Daniel Lemire
Home page: http://lemire.me/en/
Project page: https://github.com/lemire/FrameOfReference
License: Apache License Version 2.0 http://www.apache.org/licenses/LICENSE-2.0

This project includes code from the TensorFlow project

Copyright 2015 The TensorFlow Authors. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This project includes code from the NumPy project.

https://github.com/numpy/numpy/blob/e1f191c46f2eebd6cb892a4bfe14d9dd43a06c4e/numpy/core/src/multiarray/multiarraymodule.c#L2910

https://github.com/numpy/numpy/blob/68fd82271b9ea5a9e50d4e761061dfcca851382a/numpy/core/src/multiarray/datetime.c

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following
  disclaimer in the documentation and/or other materials provided
  with the distribution.

* Neither the name of the NumPy Developers nor the names of any

contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This project includes code from the Boost project

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including

----

This project includes code from the tslib project

This project includes code from the jemalloc project

https://github.com/jemalloc/jemalloc

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice(s),
   this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice(s),
   this list of conditions and the following disclaimer in the documentation
   and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER(S) ``AS IS'' AND ANY EXPRESS
OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE COPYRIGHT HOLDER(S) BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This project includes code from the Go project, BSD 3-clause license + PATENTS
weak patent termination clause
(https://github.com/golang/go/blob/master/PATENTS).

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This project includes code from the hs2client

https://github.com/cloudera/hs2client

The script ci/scripts/util_wait_for_it.sh has the following license

use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

The script r/configure has the following license (MIT)

Copyright (c) 2017, Jeroen Ooms and Jim Hester

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

----

cpp/src/arrow/util/logging.cc, cpp/src/arrow/util/logging.h and
cpp/src/arrow/util/logging-test.cc are adapted from
Ray Project (https://github.com/ray-project/ray) (Apache 2.0).

See the License for the specific language governing permissions and limitations under the License.

The files cpp/src/arrow/vendored/datetime/date.h,
cpp/src/arrow/vendored/datetime/tz.h,
cpp/src/arrow/vendored/datetime/tz_private.h,
cpp/src/arrow/vendored/datetime/ios.h,
cpp/src/arrow/vendored/datetime/ios.mm,
cpp/src/arrow/vendored/datetime/tz.cpp are adapted from
Howard Hinnant's date library (https://github.com/HowardHinnant/date)
It is licensed under MIT license.

The file cpp/src/arrow/util/utf8.h includes code adapted from the page
https://bjoern.hoehrmann.de/utf-8/decoder/dfa/
with the following license (MIT)

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The files in cpp/src/arrow/vendored/xxhash/ have the following license
(BSD 2-Clause License)

xxHash Library
Copyright (c) 2012-2014, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

You can contact the author at :

- xxHash homepage: http://www.xxhash.com

- xxHash source repository : https://github.com/Cyan4973/xxHash

----

The files in cpp/src/arrow/vendored/double-conversion/ have the following license

(BSD 3-Clause License)

notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials provided
with the distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The files in cpp/src/arrow/vendored/uriparser/ have the following license (BSD 3-Clause License)

uriparser - RFC 3986 URI parsing library

Copyright (C) 2007, Weijia Song <songweijia@gmail.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The files under dev/tasks/conda-recipes have the following license

BSD 3-clause license
Copyright (c) 2015-2018, conda-forge
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR

TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF

THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The files in cpp/src/arrow/vendored/utfcpp/ have the following license

Copyright 2006-2018 Nemanja Trifunovic

Permission is hereby granted, free of charge, to any person or organization

obtaining a copy of the software and accompanying documentation covered by

this license (the "Software") to use, reproduce, display, distribute,

execute, and transmit the Software, and to prepare derivative works of the

Software, and to permit third-parties to whom the Software is furnished to

do so, all subject to the following:

The copyright notices in the Software and this entire statement, including

the above license grant, this restriction and the following disclaimer,

must be included in all copies of the Software, in whole or in part, and

all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

This project includes code from Apache Kudu.

 * cpp/cmake_modules/CompilerInfo.cmake is based on Kudu's
cmake_modules/CompilerInfo.cmake

Copyright: 2016 The Apache Software Foundation.
Home page: https://kudu.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

This project includes code from Apache Impala (incubating), formerly
Impala. The Impala code and rights were donated to the ASF as part of the
Incubator process after the initial code imports into Apache Parquet.

Home page: http://impala.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

This project includes code from Apache Aurora.

* dev/release/{release,changelog,release-candidate} are based on the scripts from
  Apache Aurora

Home page: https://aurora.apache.org/
License: http://www.apache.org/licenses/LICENSE-2.0

This project includes code from the Google styleguide.

* cpp/build-support/cpplint.py is based on the scripts from the Google styleguide.

Homepage: https://github.com/google/styleguide
License: 3-clause BSD

This project includes code from Snappy.

* cpp/cmake_modules/{SnappyCMakeLists.txt,SnappyConfig.h} are based on code
  from Google's Snappy project.

Homepage: https://github.com/google/snappy
License: 3-clause BSD

This project includes code from the manylinux project.

* python/manylinux1/scripts/{build_python.sh,python-tag-abi-tag.py,
  requirements.txt} are based on code from the manylinux project.

Homepage: https://github.com/pypa/manylinux
License: The MIT License (MIT)

This project includes code from the cymove project:

* python/pyarrow/includes/common.pxd includes code from the cymove project

The MIT License (MIT)

The projects includes code from the Ursabot project under the dev/archery
directory.

License: BSD 2-Clause

modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL

DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This project include code from mingw-w64.

* cpp/src/arrow/util/cpu-info.cc has a polyfill for mingw-w64 < 5

Copyright (c) 2009 - 2013 by the mingw-w64 project
Homepage: https://mingw-w64.org
License: Zope Public License (ZPL) Version 2.1.

This project include code from Google's Asylo project.

* cpp/src/arrow/result.h is based on status_or.h

Copyright (c) Copyright 2017 Asylo authors
Homepage: https://asylo.dev/
License: Apache 2.0

This project includes code from Google's protobuf project

* cpp/src/arrow/result.h ARROW_ASSIGN_OR_RAISE is based off ASSIGN_OR_RETURN
* cpp/src/arrow/util/bit_stream_utils.h contains code from wire_format_lite.h

Copyright 2008 Google Inc. All rights reserved.
Homepage: https://developers.google.com/protocol-buffers/
License:

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner of the input file used when generating it. This code is not standalone and requires a support library to be linked with it. This support library is itself covered by the above license.

3rdparty dependency LLVM is statically linked in certain binary distributions. Additionally some sections of source code have been derived from sources in LLVM and have been clearly labeled as such. LLVM has the following license:

==
=
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==
=

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing

the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

==
=

Software from third parties included in the LLVM Project:

==
=

The LLVM Project contains third party software which is under different license terms. All such code will be identified clearly using at least one of two mechanisms:

1) It will be in a separate directory tree with its own `LICENSE.txt` or `LICENSE` file at the top containing the specific license and restrictions which apply to that software, or
2) It will contain specific license and restriction terms at the top of every file.

3rdparty dependency gRPC is statically linked in certain binary distributions, like the python wheels. gRPC has the following license:

3rdparty dependency Apache Thrift is statically linked in certain binary distributions, like the python wheels. Apache Thrift has the following license:

3rdparty dependency Apache ORC is statically linked in certain binary distributions, like the python wheels. Apache ORC has the following license:

3rdparty dependency zstd is statically linked in certain binary distributions, like the python wheels. ZSTD has the following license:

BSD License

For Zstandard software

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3rdparty dependency lz4 is statically linked in certain binary
distributions, like the python wheels. lz4 has the following license:

LZ4 Library

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
  list of conditions and the following disclaimer in the documentation and/or
  other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND

ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE

DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

3rdparty dependency Brotli is statically linked in certain binary

distributions, like the python wheels. Brotli has the following license:

3rdparty dependency rapidjson is statically linked in certain binary distributions, like the python wheels. rapidjson and its dependencies have the following licenses:

Tencent is pleased to support the open source community by making RapidJSON available.

If you have downloaded a copy of the RapidJSON binary from Tencent, please note that the RapidJSON binary is licensed under the MIT License.
If you have downloaded a copy of the RapidJSON source code from Tencent, please note that RapidJSON source code is licensed under the MIT License, except for the third-party components listed below which are subject to different license terms. Your integration of RapidJSON into your own projects may require compliance with the MIT License, as well as the other licenses applicable to the third-party components included within RapidJSON. To avoid the problematic JSON license in your own projects, it's sufficient to exclude the bin/jsonchecker/ directory, as it's the only code under the JSON license.
A copy of the MIT License is included in this file.

Other dependencies and licenses:

Open Source Software Licensed Under the BSD License:

The msinttypes r29

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS AND CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR

SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Terms of the MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3rdparty dependency snappy is statically linked in certain binary distributions, like the python wheels. snappy has the following license:

3rdparty dependency gflags is statically linked in certain binary distributions, like the python wheels. gflags has the following license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3rdparty dependency glog is statically linked in certain binary distributions, like the python wheels. glog has the following license:

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A function gettimeofday in utilities.cc is based on

http://www.google.com/codesearch/p?
hl=en#dR3YEbitojA/COPYING&q=GetSystemTimeAsFileTime%20license:bsd

The license of this code is:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name(s) of the above-listed copyright holder(s) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

3rdparty dependency re2 is statically linked in certain binary distributions, like the python wheels. re2 has the following license:

Copyright (c) 2009 The RE2 Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

3rdparty dependency c-ares is statically linked in certain binary distributions, like the python wheels. c-ares has the following license:

c-ares license

3rdparty dependency zlib is redistributed as a dynamically linked shared library in certain binary distributions, like the python wheels. In the future this will likely change to static linkage. zlib has the following license:

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.11, January 15th, 2017

Jean-loup Gailly   Mark Adler
jloup@gzip.org   madler@alumni.caltech.edu

3rdparty dependency openssl is redistributed as a dynamically linked shared library in certain binary distributions, like the python wheels. openssl preceding version 3 has the following license:

LICENSE ISSUES
==============

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

Original SSLeay License

```
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
```

*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]

*/

This project includes code from the rtools-backports project.

* ci/scripts/PKGBUILD and ci/scripts/r_windows_build.sh are based on code
  from the rtools-backports project.

Homepage: https://github.com/r-windows/rtools-backports
License: 3-clause BSD

Some code from pandas has been adapted for the pyarrow codebase. pandas is available under the 3-clause BSD license, which follows:

pandas license
==============

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some bits from DyND, in particular aspects of the build system, have been adapted from libdynd and dynd-python under the terms of the BSD 2-clause license

The BSD 2-Clause License

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Dynamic NDArray Developers list:

* Mark Wiebe
* Continuum Analytics

Some source code from Ibis (https://github.com/cloudera/ibis) has been adapted for PyArrow. Ibis is released under the Apache License, Version 2.0.

----

dev/tasks/homebrew-formulae/apache-arrow.rb has the following license:

BSD 2-Clause License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

----

cpp/src/arrow/vendored/base64.cpp has the following license

ZLIB License

Copyright (C) 2004-2017 René Nyffenegger

This source code is provided 'as-is', without any express or implied
warranty. In no event will the author be held liable for any damages arising
from the use of this software.

This project includes code from Folly.

 * cpp/src/arrow/vendored/ProducerConsumerQueue.h

is based on Folly's

 * folly/Portability.h
 * folly/lang/Align.h
 * folly/ProducerConsumerQueue.h

Home page: https://github.com/facebook/folly
License: http://www.apache.org/licenses/LICENSE-2.0

The file cpp/src/arrow/vendored/musl/strptime.c has the following license

----

The file cpp/cmake_modules/BuildUtils.cmake contains code from

https://gist.github.com/cristianadam/ef920342939a89fae3e8a85ca9459b49

which is made available under the MIT license

The files in cpp/src/arrow/vendored/portable-snippets/ contain code from

https://github.com/nemequ/portable-snippets

and have the following copyright notice:

The files in cpp/src/arrow/vendored/fast_float/ contain code from

https://github.com/lemire/fast_float

which is made available under the Apache License 2.0.

The file python/pyarrow/vendored/docscrape.py contains code from

https://github.com/numpy/numpydoc/

which is made available under the BSD 2-clause license.

The file python/pyarrow/vendored/version.py contains code from

https://github.com/pypa/packaging/

which is made available under both the Apache license v2.0 and the BSD 2-clause license.

The files in cpp/src/arrow/vendored/pcg contain code from

https://github.com/imneme/pcg-cpp

and have the following copyright notice:

be copied, modified, or distributed except according to those terms.

Distributed on an "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, either
express or implied. See your chosen license for details.

----

r/R/dplyr-count-tally.R (some portions)

Some portions of this file are derived from code from

https://github.com/tidyverse/dplyr/

which is made available under the MIT license

The file src/arrow/util/io_util.cc contains code from the CPython project
which is made available under the Python Software Foundation License Version 2.

3rdparty dependency opentelemetry-cpp is statically linked in certain binary
distributions. opentelemetry-cpp is made available under the Apache License 2.0.

----

ci/conan/ is based on code from Conan Package and Dependency Manager.

copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3rdparty dependency UCX is redistributed as a dynamically linked shared
library in certain binary distributions. UCX has the following license:

The file dev/tasks/r/github.packages.yml contains code from

https://github.com/ursa-labs/arrow-r-nightly

which is made available under the Apache License 2.0.

.github/actions/sync-nightlies/action.yml (some portions)

Some portions of this file are derived from code from

https://github.com/JoshPiper/rsync-docker

which is made available under the MIT license

.github/actions/sync-nightlies/action.yml (some portions)

Some portions of this file are derived from code from

https://github.com/burnett01/rsync-deployments

which is made available under the MIT license

NOTICE.txt:

Apache Arrow
Copyright 2016-2019 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

This product includes software from the SFrame project (BSD, 3-clause).
* Copyright (C) 2015 Dato, Inc.
* Copyright (c) 2009 Carnegie Mellon University.

This product includes software from the Feather project (Apache 2.0)
https://github.com/wesm/feather

This product includes software from the DyND project (BSD 2-clause)
https://github.com/libdynd

This product includes software from the LLVM project
 * distributed under the University of Illinois Open Source

This product includes software from the google-lint project
 * Copyright (c) 2009 Google Inc. All rights reserved.

This product includes software from the mman-win32 project
 * Copyright https://code.google.com/p/mman-win32/
 * Licensed under the MIT License;

This product includes software from the LevelDB project
 * Copyright (c) 2011 The LevelDB Authors. All rights reserved.
 * Use of this source code is governed by a BSD-style license that can be
 * Moved from Kudu http://github.com/cloudera/kudu

This product includes software from the CMake project
 * Copyright 2001-2009 Kitware, Inc.
 * Copyright 2012-2014 Continuum Analytics, Inc.
 * All rights reserved.

This product includes software from https://github.com/matthew-brett/multibuild (BSD 2-clause)
 * Copyright (c) 2013-2016, Matt Terry and Matthew Brett; all rights reserved.

This product includes software from the Ibis project (Apache 2.0)
 * Copyright (c) 2015 Cloudera, Inc.
 * https://github.com/cloudera/ibis

This product includes software from Dremio (Apache 2.0)
 * Copyright (C) 2017-2018 Dremio Corporation
 * https://github.com/dremio/dremio-oss

This product includes software from Google Guava (Apache 2.0)
 * Copyright (C) 2007 The Guava Authors
 * https://github.com/google/guava

This product include software from CMake (BSD 3-Clause)
 * CMake - Cross Platform Makefile Generator
 * Copyright 2000-2019 Kitware, Inc. and Contributors

The web site includes files generated by Jekyll.

This product includes code from Apache Kudu, which includes the following in its NOTICE file:

Apache Kudu
Copyright 2016 The Apache Software Foundation

This product includes software developed at
The Apache Software Foundation (http://www.apache.org/).

Portions of this software were developed at
Cloudera, Inc (http://www.cloudera.com/).

This product includes code from Apache ORC, which includes the following in its NOTICE file:

Apache ORC
Copyright 2013-2019 The Apache Software Foundation

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).

This product includes software developed by Hewlett-Packard:
(c) Copyright [2014-2015] Hewlett-Packard Development Company, L.P

Apache Portable Runtime Apache License V 2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with

the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the

same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Boost License

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including

brotli License

bzip2 License

3. Altered source versions must be plainly marked as such, and must
   not be misrepresented as being the original software.

4. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior written
   permission.

Julian Seward, jseward@acm.org
bzip2/libbzip2 version 1.0.8 of 13 July 2019

CDF License

CDF Copyright

Space Physics Data Facility

NASA/Goddard Space Flight Center

This software may be copied or redistributed as long as it is not sold for profit, but it can be incorporated into any other substantive product with or without modifications for profit or non¬profit. If the software is modified, it must include the following notices:

The software is not the original (for protection of the original author's reputations from any

problems introduced by others)

Change history (e.g. date, functionality, etc.)

This copyright notice must be reproduced on each copy made. This software is provided as is

without any express or implied warranties whatsoever.

CrashRpt (New BSD) License

Copyright (c) 2003, The CrashRpt Project Authors.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification,

are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY

EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES

OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED

TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,

STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT

OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Curl License

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2023, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

krb5.c Notice:

*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
*
* 3. Neither the name of the Institute nor the names of its contributors
*   may be used to endorse or promote products derived from this software
*   without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE. */

inet_ntop.c and inet_pton.c Notice:

```
/*
 * Copyright (C) 1996-2022  Internet Software Consortium.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM
 * DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL
 * INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT,
 * INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER
RESULTING
 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * SPDX-License-Identifier: ISC
 */
/*
 * Original code by Paul Vixie. "curlified" by Gisle Vanem.
 */
```

DEWESOFT DWDataReader EULA

END USER LICENSE AGREEMENT (EULA) FOR DWDATAREADER SOFTWARE

This End-User License Agreement (hereinafter: EULA) is a legal agreement between you (either an individual or a

single entity) and DEWESoft d.o.o. for the DWDataReader that accompanies this EULA, which includes computer

software and may include associated media, printed materials, "online" or electronic documentation.

By accepting this agreement or by downloading DWDataReader software offered by or on behalf of DEWESoft,

d.o.o. (hereinafter: "DWDataReader") or by clicking "Save", you agree to the following terms, notwithstanding

anything to the contrary in this agreement. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, SAVE, COPY, OR

USE THE DWDATAREADER.

DWDataReader is a Dynamic library for reading *.d7d data files distributed by DEWESoft, d.o.o.

Reservation of right and ownership

Subject to your compliance with these terms and conditions, DEWESoft, d.o.o. grants you a royalty-free, non-

exclusive usage of the DWDataReader. You may use and distribute the DWDatReader on its own or as a part of

your package for the purpose of data analysis.

DEWESoft, d.o.o. reserves all rights in the DWDataReader not expressly granted to you here. The DWDataReader

is protected by copyright and other intellectual property laws and treaties. DEWESoft, d.o.o. or its suppliers own the

title, copyright, and other intellectual property rights in the DWDataReader.

All title and intellectual property rights in and to the content which may be accessed through use of the software is

the property of the respective content owner and may be protected by applicable copyright or other intellectual

property laws and treaties. This EULA grants you no rights to use such content.

Limitations on reverse engineering, decompilation and disassembly

The source code and structure of the DWDataReader are trade secrets. You may not reverse engineer, decompile or

disassemble the DWDataReader Software in whole or in part. You will not use the DWDataReader for illegal

purposes.

DWDataReader may not be sold or otherwise transferred for value, or used for any purpose other than reading *.d7d

data files distributed by DEWESoft, d.o.o. for data analysis, demonstration, testing or evaluation purposes.

Disclaimer of warranties

DEWESoft, d.o.o. and its suppliers provide the DWDataReader Software and support services (if any) AS IS AND

WITH ALL FAULTS, and hereby disclaim all other warranties and conditions, whether express, implied or

statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of

fitness for a particular purpose, of reliability or availability, of accuracy or completeness of responses, of results, of

workmanlike effort, of lack of viruses, and of lack of negligence, all with regard to the DWDataReader Software,

and the provision of or failure to provide support or other services, information, software, and related content through the DWDataReader Software or otherwise arising out of the use of the DWDataReader Software. ALSO,

THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, AND

CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE

DWDATAREADER SOFTWARE.

Dewesoft, d.o.o. disclaims any responsibility for harm resulting from the DWDataReader or usage of the

DWDataReader. Dewesoft, d.o.o. does not guarantee that DWDataReader software will function or appear as

offered or hoped, or be complete, accurate, or free from bugs, errors, viruses, or other harmful content.

No advice or information, whether oral or written, obtained from Dewesoft, d.o.o. or elsewhere will create any

warranty or condition not expressly stated in this agreement.

Exclusion of incidental, consequential and certain other damages

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL DEWESOFT

OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, PUNITIVE, INDIRECT, OR

CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR

LOSS OF PROFITS OR CONFIDENTIAL OR OTHER INFORMATION, FOR BUSINESS INTERRUPTION,

FOR PERSONAL INJURY, FOR LOSS OF PRIVACY, FOR FAILURE TO MEET ANY DUTY INCLUDING OF

GOOD FAITH OR OF REASONABLE CARE, FOR NEGLIGENCE, AND FOR ANY OTHER PECUNIARY OR

OTHER LOSS WHATSOEVER) ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR

INABILITY TO USE THE DWDATAREADER SOFTWARE, THE PROVISION OF OR FAILURE TO

PROVIDE SUPPORT OR OTHER SERVICES, INFORMATON, DWDATAREADER SOFTWARE, AND

RELATED CONTENT THROUGH THE DWDATAREADER SOFTWARE OR OTHERWISE ARISING OUT

OF THE USE OF THE DWDATAREADER SOFTWARE, OR OTHERWISE UNDER OR IN CONNECTION

WITH ANY PROVISION OF THIS EULA, EVEN IN THE EVENT OF THE FAULT, TORT (INCLUDING

NEGLIGENCE), MISREPRESENTATION, STRICT LIABILITY, BREACH OF CONTRACT OR BREACH OF

WARRANTY OF DEWESOFT OR ANY SUPPLIER, AND EVEN IF DEWESOFT OR ANY SUPPLIER HAS

BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Applicable law

This EULA is governed by the laws of Republic of Slovenia and, in respect of any dispute which may arise

hereunder, you consent to the jurisdiction of the federal and provincial courts sitting in Ljubljana, Slovenia.

Dewesoft, d.o.o. may make changes to these terms from time to time. When these changes are made, Dewesoft,

d.o.o. will make a new copy of the terms available at http://www.dewesoft.com. You understand and agree that if

you use the DWDataReader Software after the date on which the terms have changed, Dewesoft, d.o.o. will treat

your use as acceptance of the updated terms. You agree that Dewesoft, d.o.o. may provide you with notices,

including those regarding changes to the terms, by postings on http://www.dewesoft.com.

This EULA refers to all versions of DWDataReader and its updates, made by Dewesoft, d.o.o. and presents

Dewesoft's complete and exclusive understanding with you regarding your use of the DWDataReader Software as

an end user.

Contact

For any legal or technical questions regarding usage of DWDataReader software please contact:

support@dewesoft.com.

Dewesoft, d.o.o.

Trbovlje, 10. 3. 2014

HDF Group HDF 5 Library License

Copyright Notice and License Terms for
HDF5 (Hierarchical Data Format 5) Software Library and Utilities
----

Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.

3. Neither the name of The HDF Group, the name of the University, nor the name of any Contributor may be used to endorse or promote products derived from this software without specific prior written permission from The HDF Group, the University, or the Contributor, respectively.

the University of Illinois, Fortner Software, Unidata Program Center (netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip), and Digital Equipment Corporation (DEC).

Portions of HDF5 were developed with support from the Lawrence Berkeley National Laboratory (LBNL) and the United States Department of Energy under Prime Contract No. DE-AC02-05CH11231.

Portions of HDF5 were developed with support from Lawrence Livermore National Laboratory and the United States Department of Energy under Prime Contract No. DE-AC52-07NA27344.

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL).
The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

ICU License

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING DATA FILES, AND/OR SOFTWARE, YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

Permission is hereby granted, free of charge, to any person obtaining a copy of data files and any associated documentation (the "Data Files") or software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that either (a) this copyright and permission notice appear with all copies of the Data Files or Software, or (b) this copyright and permission notice appear in associated Documentation.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.

IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Third-Party Software Licenses

This section contains third-party software notices and/or additional terms for licensed third-party software components included within ICU libraries.

ICU License - ICU 1.8.1 to ICU 57.1

COPYRIGHT AND PERMISSION NOTICE

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above

----

Chinese/Japanese Word Break Dictionary Data (cjdict.txt)

```
# BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
# LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
# NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
#
#  The word list in cjdict.txt are generated by combining three word lists
# listed below with further processing for compound word breaking. The
# frequency is generated with an iterative training against Google web
# corpora.
#
#  * Libtabe (Chinese)
#    - https://sourceforge.net/project/?group_id=1519
#    - Its license terms and conditions are shown below.
#
#  * IPADIC (Japanese)
#    - http://chasen.aist-nara.ac.jp/chasen/distribution.html
#    - Its license terms and conditions are shown below.
#
#  ---------COPYING.libtabe ---- BEGIN--------------------
#
#  /*
#   * Copyright (c) 1999 TaBE Project.
#   * Copyright (c) 1999 Pai-Hsiang Hsiao.
#   * All rights reserved.
#   *
#   * Redistribution and use in source and binary forms, with or without
#   * modification, are permitted provided that the following conditions
#   * are met:
```

```
#  *
#  * . Redistributions of source code must retain the above copyright
#  *   notice, this list of conditions and the following disclaimer.
#  * . Redistributions in binary form must reproduce the above copyright
#  *   notice, this list of conditions and the following disclaimer in
#  *   the documentation and/or other materials provided with the
#  *   distribution.
#  * . Neither the name of the TaBE Project nor the names of its
#  *   contributors may be used to endorse or promote products derived
#  *   from this software without specific prior written permission.
#  *
#  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#  * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#  * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#  * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#  * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#  * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#  * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#  * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#  * OF THE POSSIBILITY OF SUCH DAMAGE.
#  */
#
# /*
#  * Copyright (c) 1999 Computer Systems and Communication Lab,
#  *                  Institute of Information Science, Academia
#     *                  Sinica. All rights reserved.
```

```
#  *
#  * Redistribution and use in source and binary forms, with or without
#  * modification, are permitted provided that the following conditions
#  * are met:
#  *
#  * . Redistributions of source code must retain the above copyright
#  *   notice, this list of conditions and the following disclaimer.
#  * . Redistributions in binary form must reproduce the above copyright
#  *   notice, this list of conditions and the following disclaimer in
#  *   the documentation and/or other materials provided with the
#  *   distribution.
#  * . Neither the name of the Computer Systems and Communication Lab
#  *   nor the names of its contributors may be used to endorse or
#  *   promote products derived from this software without specific
#  *   prior written permission.
#  *
#  * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
#  * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
#  * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
#  * FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
#  * REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
#  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
#  * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
#  * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
#  * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
#  * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
#  * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
#  * OF THE POSSIBILITY OF SUCH DAMAGE.
#  */
```

```
#
# Copyright 1996 Chih-Hao Tsai @ Beckman Institute,
#     University of Illinois
# c-tsai4@uiuc.edu  http://casper.beckman.uiuc.edu/~c-tsai4
#
# ---------------COPYING.libtabe-----END-------------------------------
#
#
# ---------------COPYING.ipadic-----BEGIN------------------------------
#
# Copyright 2000, 2001, 2002, 2003 Nara Institute of Science
# and Technology.  All Rights Reserved.
#
# Use, reproduction, and distribution of this software is permitted.
# Any copy of this software, whether in its original form or modified,
# must include both the above copyright notice and the following
# paragraphs.
#
# Nara Institute of Science and Technology (NAIST),
# the copyright holders, disclaims all warranties with regard to this
# software, including all implied warranties of merchantability and
# fitness, in no event shall NAIST be liable for
# any special, indirect or consequential damages or any damages
# whatsoever resulting from loss of use, data or profits, whether in an
# action of contract, negligence or other tortuous action, arising out
# of or in connection with the use or performance of this software.
#
# A large portion of the dictionary entries
# originate from ICOT Free Software.  The following conditions for ICOT
```

```
# Free Software applies to the current dictionary as well.
#
# Each User may also freely distribute the Program, whether in its
# original form or modified, to any third party or parties, PROVIDED
# that the provisions of Section 3 ("NO WARRANTY") will ALWAYS appear
# on, or be attached to, the Program, which is distributed substantially
# in the same form as set out herein and that such intended
# distribution, if actually made, will neither violate or otherwise
# contravene any of the laws and regulations of the countries having
# jurisdiction over the User or the intended distribution itself.
#
# NO WARRANTY
#
# The program was produced on an experimental basis in the course of the
# research and development conducted during the project and is provided
# to users as so produced on an experimental basis.  Accordingly, the
# program is provided without any warranty whatsoever, whether express,
# implied, statutory or otherwise.  The term "warranty" used herein
# includes, but is not limited to, any warranty of the quality,
# performance, merchantability and fitness for a particular purpose of
# the program and the nonexistence of any infringement or violation of
# any right of any third party.
#
# Each user of the program will agree and understand, and be deemed to
# have agreed and understood, that there is no warranty whatsoever for
# the program and, accordingly, the entire risk arising from or
# otherwise connected with the program is assumed by the user.
#
# Therefore, neither ICOT, the copyright holder, or any other
```

```
# organization that participated in or was otherwise related to the
# development of the program and their respective officials, directors,
# officers and other employees shall be held liable for any and all
# damages, including, without limitation, general, special, incidental
# and consequential damages, arising out of or otherwise in connection
# with the use or inability to use the program or any product, material
# or result produced or otherwise obtained by using the program,
# regardless of whether they have been advised of, or otherwise had
# knowledge of, the possibility of such damages at any time during the
# project or thereafter.  Each user will be deemed to have agreed to the
# foregoing by his or her commencement of use of the program.  The term
# "use" as used herein includes, but is not limited to, the use,
# modification, copying and distribution of the program and the
# production of secondary products from the program.
#
# In the case where the program, whether in its original form or
# modified, was distributed or delivered to or received by a user from
# any person, organization or entity other than ICOT, unless it makes or
# grants independently of ICOT any specific warranty to the user in
# writing, such person, organization or entity, will also be exempted
# from and not be held liable to the user for any such damages as noted
# above as far as the program is concerned.
#
# ---------------COPYING.ipadic-----END----------------------------------
```

Lao Word Break Dictionary Data (laodict.txt)

```
# Copyright (C) 2016 and later: Unicode, Inc. and others.
# License & terms of use: http://www.unicode.org/copyright.html
# Copyright (c) 2015 International Business Machines Corporation
# and others. All Rights Reserved.
#
# Project: https://github.com/rober42539/lao-dictionary
# Dictionary: https://github.com/rober42539/lao-dictionary/laodict.txt
# License: https://github.com/rober42539/lao-dictionary/LICENSE.txt
#         (copied below)
#
#          This file is derived from the above dictionary version of Nov 22, 2020
#  ----------------------------------------------------------------------
#  Copyright (C) 2013 Brian Eugene Wilson, Robert Martin Campbell.
#  All rights reserved.
#
#  Redistribution and use in source and binary forms, with or without
#  modification, are permitted provided that the following conditions are met:
#
#  Redistributions of source code must retain the above copyright notice, this
#  list of conditions and the following disclaimer. Redistributions in binary
#  form must reproduce the above copyright notice, this list of conditions and
#  the following disclaimer in the documentation and/or other materials
#  provided with the distribution.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
```

Burmese Word Break Dictionary Data (burmesedict.txt)

disclaimer in the documentation and/or other materials provided
with the distribution.
#
Neither the name Myanmar Karen Word Lists, nor the names of its
contributors may be used to endorse or promote products derived
from this software without specific prior written permission.
#
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED
TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
--

Time Zone Database

ICU uses the public domain data and code derived from Time Zone Database for its time zone support. The ownership of the TZ database is explained in BCP 175: Procedure for Maintaining the Time Zone

Database section 7.

```
# 7.  Database Ownership
#
#    The TZ database itself is not an IETF Contribution or an IETF
#    document.  Rather it is a pre-existing and regularly updated work
#    that is in the public domain, and is intended to remain in the
#    public domain.  Therefore, BCPs 78 [RFC5378] and 79 [RFC3979] do
#    not apply to the TZ Database or contributions that individuals make
#    to it.  Should any claims be made and substantiated against the TZ
#    Database, the organization that is providing the IANA
#    Considerations defined in this RFC, under the memorandum of
#    understanding with the IETF, currently ICANN, may act in accordance
#    with all competent court orders.  No ownership claims will be made
#    by ICANN or the IETF Trust on the database or the code.  Any person
#    making a contribution to the database or code waives all rights to
#    future claims in that contribution or in the TZ Database.
```

Google double-conversion

Copyright 2006-2011, the V8 project authors. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

File: aclocal.m4 (only for ICU4C)

Section: pkg.m4 - Macros to locate and utilise pkg-config.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

As a special exception to the GNU General Public License, if you distribute this file as part of a program that contains a configuration script generated by Autoconf, you may include it under the same distribution terms that you use for the rest of that program.

(The condition for the exception is fulfilled because ICU4C includes a configuration script generated by Autoconf, namely the `configure` script.)

File: config.guess (only for ICU4C)

(The condition for the exception is fulfilled because
ICU4C includes a configuration script generated by Autoconf,
namely the `configure` script.)

File: install-sh (only for ICU4C)

JSON for Modern C++

MIT License

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

libaec License

Copyright 2024 Mathis Rosenhauer, Moritz Hanke, Joerg Behrens, Luis Kornblueh
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

libf2c Library License

/**

Copyright 1990 - 1997 by AT&T, Lucent Technologies and Bellcore.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of AT&T, Bell Laboratories, Lucent or Bellcore or any of their entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

AT&T, Lucent and Bellcore disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall AT&T, Lucent or Bellcore be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use,

data or profits, whether in an action of contract, negligence or
other tortious action, arising out of or in connection with the
use or performance of this software.
***/

libjpeg-turbo Licenses

libjpeg-turbo Licenses
=====================

libjpeg-turbo is covered by three compatible BSD-style open source licenses:

- The IJG (Independent JPEG Group) License, which is listed in
  README.ijg

  This license applies to the libjpeg API library and associated programs
  (any code inherited from libjpeg, and any modifications to that code.)

- The Modified (3-clause) BSD License, which is listed below

  This license covers the TurboJPEG API library and associated programs, as
  well as the build system.

- The [zlib License](https://opensource.org/licenses/Zlib)

  This license is a subset of the other two, and it covers the libjpeg-turbo
  SIMD extensions.

Complying with the libjpeg-turbo Licenses
==

This section provides a roll-up of the libjpeg-turbo licensing terms, to the best of our understanding.

1. If you are distributing a modified version of the libjpeg-turbo source, then:

    1. You cannot alter or remove any existing copyright or license notices from the source.

        Origin
        - Clause 1 of the IJG License
        - Clause 1 of the Modified BSD License
        - Clauses 1 and 3 of the zlib License

    2. You must add your own copyright notice to the header of each source file you modified, so others can tell that you modified that file (if there is not an existing copyright header in that file, then you can simply add a notice stating that you modified the file.)

        Origin
        - Clause 1 of the IJG License
        - Clause 2 of the zlib License

    3. You must include the IJG README file, and you must not alter any of the copyright or license text in that file.

    Origin
    - Clause 1 of the IJG License

2. If you are distributing only libjpeg-turbo binaries without the source, or
   if you are distributing an application that statically links with
   libjpeg-turbo, then:

   1. Your product documentation must include a message stating:

      This software is based in part on the work of the Independent JPEG
      Group.

      Origin
      - Clause 2 of the IJG license

   2. If your binary distribution includes or uses the TurboJPEG API, then
      your product documentation must include the text of the Modified BSD
      License (see below.)

      Origin
      - Clause 2 of the Modified BSD License

3. You cannot use the name of the IJG or The libjpeg-turbo Project or the
   contributors thereof in advertising, publicity, etc.

   Origin
   - IJG License
   - Clause 3 of the Modified BSD License

4. The IJG and The libjpeg-turbo Project do not warrant libjpeg-turbo to be free of defects, nor do we accept any liability for undesirable consequences resulting from your use of the software.

Origin
- IJG License
- Modified BSD License
- zlib License

The Modified (3-clause) BSD License
===================================

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the libjpeg-turbo Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Why Three Licenses?
===================

The zlib License could have been used instead of the Modified (3-clause) BSD License, and since the IJG License effectively subsumes the distribution conditions of the zlib License, this would have effectively placed libjpeg-turbo binary distributions under the IJG License. However, the IJG License specifically refers to the Independent JPEG Group and does not extend attribution and endorsement protections to other entities. Thus, it was desirable to choose a license that granted us the same protections for new code that were granted to the IJG for code derived from their software.

LIBMATIO MAT File IO Library License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

libmseed - miniSEED Library - License

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of

this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all

other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

LIBPNG License

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE
===

PNG Reference Library License version 2
----

non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

----

libpng versions 1.0.7, July 1, 2000, through 1.6.35, July 15, 2018 are Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson, are derived from libpng-1.0.6, and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors:

Simon-Pierre Cadieux
Eric S. Raymond
Mans Rullgard
Cosmin Truta
Gilles Vollant
James Yu
Mandar Sahastrabuddhe
Google Inc.
Vadim Barkov

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Some files in the "contrib" directory and some configure-generated files that are distributed with libpng have other copyright owners, and are released under other open source licenses.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998-2000 Glenn Randers-Pehrson, are derived from libpng-0.96, and are distributed according to the same disclaimer and

license as libpng-0.96, with the following individuals added to the
list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996-1997 Andreas Dilger, are derived from libpng-0.88,
and are distributed according to the same disclaimer and license as
libpng-0.88, with the following individuals added to the list of
Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Some files in the "scripts" directory have other copyright owners,
but are released under this license.

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"
is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

libxlswriter Library License

Libxlsxwriter is released under a FreeBSD license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF

LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

Libxlsxwriter includes the `queue.h` and `tree.h` macros from FreeBSD. It also includes and, unless overridden, uses the optional libraries `minizip`, `tmpfileplus` and `md5`. It also includes the `emyg_dtoa` library but doesn't use it by default. These components have the following licenses:

Queue.h from FreeBSD:

Copyright (c) 1991, 1993
The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
4. Neither the name of the University nor the names of its contributors

may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

Tree.h from FreeBSD:

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The `minizip` files used in the libxlsxwriter source tree are taken from the `zlib` ` contrib/minizip` directory. [Zlib](http://www.zlib.net) has the following License/Copyright:

(C) 1995-2013 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not

claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

The `minizip` files have the following additional copyright declarations:

Copyright (C) 1998-2010 Gilles Vollant
(minizip) (http://www.winimage.com/zLibDll/minizip.html)

Modifications for Zip64 support
Copyright (C) 2009-2010 Mathias Svensson (http://result42.com)

Note, it is possible to compile libxlsxwriter without statically linking the
`minizip` files and instead dynamically linking to `lminizip`, see
@ref gsg_minizip.

[Tmpfileplus](http://www.di-mgt.com.au/c_function_to_create_temp_file.html)
has the following license:

This Source Code Form is subject to the terms of the Mozilla Public
License, v. 2.0. If a copy of the MPL was not distributed with this
file, You can obtain one at http://mozilla.org/MPL/2.0/.

Copyright (c) 2012-16 David Ireland, DI Management Services Pty Ltd
<http://www.di-mgt.com.au/contact/>.

See the [Mozilla Public License, v. 2.0](http://mozilla.org/MPL/2.0/).

Note, it is possible to compile libxlsxwriter using the standard library `tmpfile()` function instead of `tmpfileplus`, see @ref gsg_tmpdir.

The [Milo Yip DTOA library](https://github.com/miloyip/dtoa-benchmark) for converting doubles to strings. It has the following license:

Copyright (C) 2015 Doug Currie
based on dtoa_milo.h
Copyright (C) 2014 Milo Yip

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

This Milo Yip DTOA library (emyg_dtoa) is used to avoid issues where the standard sprintf() dtoa function changes output based on locale settings. It is also 40-50% faster than the standard dtoa for raw numeric data. The use of this library is optional. If you wish to use it you can pass `USE_DTOA_LIBRARY=1` to make when compiling.

[Openwall MD5](https://openwall.info/wiki/people/solar/software/public-domain-source-code/md5)
has the following licence:

This software was written by Alexander Peslyak in 2001. No copyright is claimed, and the software is hereby placed in the public domain.
In case this attempt to disclaim copyright and place the software in the public domain is deemed null and void, then the software is Copyright (c) 2001 Alexander Peslyak and it is hereby released to the general public under the following terms:

Redistribution and use in source and binary forms, with or without modification, are permitted.

There's ABSOLUTELY NO WARRANTY, express or implied.

(This is a heavily cut-down "BSD license".)

Note, the MD5 library is used to avoid including duplicate image files in the
xlsx file. If you don't want to use this code, and the additional licence, you
can use OpenSSL's MD5 functions instead by passing `USE_OPENSSL_MD5=1` to
make. If this functionality isn't required it is possible to compile
libxlsxwriter without image deduplication by passing `USE_NO_MD5=1` to make.

LZ4 Library License

LZ4 Library
Copyright (c) 2011-2020, Yann Collet
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this
  list of conditions and the following disclaimer in the documentation and/or
  other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;

LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON

ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS

SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Microsoft Detours MIT Open Source License

Copyright (c) Microsoft Corporation

All rights reserved.

MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED *AS IS*, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

MiniZip License

MiniZip - Copyright (c) 1998-2010 - by Gilles Vollant - version 1.1 64 bits from Mathias Svensson

Introduction

MiniZip 1.1 is built from MiniZip 1.0 by Gilles Vollant (http://www.winimage.com/zLibDll/minizip.html)

When adding ZIP64 support into minizip it would result into risk of breaking compatibility with minizip 1.0.

All possible work was done for compatibility.

Background

When adding ZIP64 support Mathias Svensson found that Even Rouault have added ZIP64

support for unzip.c into minizip for a open source project called gdal (http://www.gdal.org/)

That was used as a starting point. And after that ZIP64 support was added to zip.c some refactoring and code cleanup was also done.

Changed from MiniZip 1.0 to MiniZip 1.1

----

* Added ZIP64 support for unzip (by Even Rouault)
* Added ZIP64 support for zip (by Mathias Svensson)
* Reverted some changed that Even Rouault did.
* Bunch of patches received from Gulles Vollant that he received for MiniZip from various users.
* Added unzip patch for BZIP Compression method (patch create by Daniel Borca)
* Added BZIP Compress method for zip
* Did some refactoring and code cleanup

Credits

Gilles Vollant - Original MiniZip author

Even Rouault - ZIP64 unzip Support

Daniel Borca - BZip Compression method support in unzip

Mathias Svensson - ZIP64 zip support

Mathias Svensson - BZip Compression method support in zip

Resources

ZipLayout http://result42.com/projects/ZipFileLayout

Command line tool for Windows that shows the layout and information of the headers in a zip archive.

Used when debugging and validating the creation of zip files using MiniZip64

ZIP App Note http://www.pkware.com/documents/casestudies/APPNOTE.TXT
Zip File specification

Notes.
* To be able to use BZip compression method in zip64.c or unzip64.c the BZIP2 lib is needed and HAVE_BZIP2 need to be defined.

License

Condition of use and distribution are the same than zlib :

Minpack License

Minpack Licensing

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the University of Chicago, as Operator of Argonne National

Laboratory.

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. WARRANTY DISCLAIMER. THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, AND THEIR EMPLOYEES: (1) DISCLAIM ANY WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, (2) DO NOT ASSUME ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS, OR USEFULNESS OF THE SOFTWARE, (3) DO NOT REPRESENT THAT USE OF THE SOFTWARE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS, (4) DO NOT WARRANT THAT THE SOFTWARE WILL FUNCTION UNINTERRUPTED, THAT IT IS ERROR-FREE OR THAT ANY ERRORS WILL BE CORRECTED.

5. LIMITATION OF LIABILITY. IN NO EVENT WILL THE COPYRIGHT HOLDER, THE UNITED STATES, THE UNITED STATES DEPARTMENT OF ENERGY, OR THEIR EMPLOYEES: BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR PUNITIVE DAMAGES OF ANY KIND OR NATURE, INCLUDING BUT NOT LIMITED TO LOSS OF PROFITS OR LOSS OF DATA, FOR ANY REASON WHATSOEVER, WHETHER SUCH LIABILITY IS ASSERTED ON THE BASIS OF CONTRACT, TORT (INCLUDING NEGLIGENCE OR STRICT LIABILITY), OR OTHERWISE, EVEN IF ANY OF SAID PARTIES HAS BEEN WARNED OF THE

POSSIBILITY OF SUCH LOSS OR DAMAGES.

NetCDF Library License

Copyright 2018 Unidata

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Netlib License

For those codes indicated with a Math a la Carte copyright, the same rules apply, except without the full force of the Caltech legal team.

When citing this software we request that you also mention the names of the people who wrote the software you are using.

NI TDM Library License

NATIONAL INSTRUMENTS SOFTWARE LICENSE AGREEMENT

FOR TDM DLL

INSTALLATION NOTICE: THIS IS A CONTRACT. BEFORE YOU DOWNLOAD THE TDM DLL AND/OR COMPLETE THE INSTALLATION PROCESS, CAREFULLY READ THIS AGREEMENT. BY DOWNLOADING THE TDM DLL AND/OR CLICKING THE APPLICABLE BUTTON TO COMPLETE THE DOWNLOAD PROCESS, YOU CONSENT TO THE TERMS OF

THIS AGREEMENT AND YOU AGREE TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT WISH TO BECOME A PARTY TO THIS AGREEMENT AND BE BOUND BY ALL OF ITS TERMS AND CONDITIONS, CLICK THE APPROPRIATE BUTTON TO CANCEL THE DOWNLOAD PROCESS AND DO NOT USE THE TDM DLL.

1. Definitions. As used in this Agreement, the following terms have the following meanings:

A. "You." Means you the individual using the TDM DLL (the "TDM DLL") as well as your
employer if you are using the TDM DLL within the scope of your employment.

B. "NI." Means National Instruments Corporation, a company organized under the laws of
the State of Delaware, U.S.A.

C. "TDM DLL." Means the computer libraries provided with this Agreement and which are
being downloaded together with all accompanying documentation, utilities, and driver
interface software

2. Grant of License; Obligations. NI grants you a limited, non-exclusive right to download, use
and distribute the TDM DLL pursuant to the terms and conditions of this Agreement. You
may use the TDM DLL in your product in order for the product to access (i.e., read and write)
TDM and/or TDMS files. If your application reads TDM files using the TDM DLL, you must
support writing TDM files as well. If your application reads TDMS files using the TDM DLL,

you must support writing TDMS files as well. You may distribute the TDM DLL as part of your

product with such limitations as set forth below.

3. Restrictions. You may not: (i) reverse engineer, decompile, or disassemble the TDM DLL

(except to the extent such foregoing restriction is expressly prohibited by applicable law); (ii)

sub-license, lease, or rent the TDM DLL; (iii) modify, or create derivatives of the TDM DLL;

and (iv) directly or indirectly, export, re-export, download, or ship the TDM DLL in violation of

the laws and regulations of the U.S.A. and the laws and regulations of the applicable

jurisdiction in which you use or are downloading the TDM DLL. Under no circumstance is

“floating” or shared use permitted under this Agreement. You agree not to use the TDM DLL

to make or distribute your own or a third party's application, a principal purpose of which, as

reasonably determined by NI, is to perform (by itself or in combination with other products)

the same or similar functions as (or are otherwise intended to replace or supplant any

component of) the TDM DLL or any other software of NI.

4. Transfer/Distribution. Your license is for your own use only. Therefore, the license is non-

transferable and you may not distribute or otherwise provide the TDM DLL to any third party

except as part of your own product which you provide to third parties. Such third parties may

not redistribute the TDM DLL and you agree to include such a prohibition in your form of license agreement. In distributing the TDM DLL as part of your product, you must include the

copyright notice: "Portions used under license from National Instruments Corporation;

copyright (c) 2007 National Instruments Corporation.

5. Multiple Versions (CD-ROM Media). If the TDM DLL is distributed on a CD-ROM containing

multiple versions of the TDM DLL, you may only use one version of the TDM DLL.

6. Copyright. The TDM DLL is owned by NI or its suppliers and is protected by applicable

copyright laws and international treaty provisions. Therefore, you must treat the TDM DLL like

any other copyrighted material. You may, however, copy the TDM DLL solely for backup or

archival purposes. All rights not expressly granted to you in this Agreement are reserved to

NI.

7. Patent and Trademark Notice For patents covering National Instruments products, refer to

the appropriate location: Help»Patents in your software, the patents.txt file on your CD,

DVD, or ni.com/patents. ComponentWorks, CVI, FieldPoint, IMAQ, Lookout, LabVIEW,

LabWindows/ CVI, Measurement Studio, NI-488.2, NI-CAN, NI-DAQ, NI-FBUS, NI-VISA, NI-

VXI and TestStand are the trademarks of National Instruments Corporation. DIAdem and

DASYLab are the trademarks of National Instruments (Ireland) Limited. Further, all other

product and company names used herein are (or may be) trademarks or trade names of their respective companies.

8. No Warranties. YOU ASSUME THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE TDM DLL. NI WILL NOT UPDATE THE TDM DLL NOR WILL NI SUPPORT THE TDM DLL. THE TDM DLL IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, AND NO WARRANTIES, EITHER EXPRESSED OR IMPLIED ARE MADE WITH RESPECT TO THE TDM DLL, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT, OR ANY OTHER WARRANTIES THAT MAY ARISE FROM USAGE OF TRADE OR COURSE OF DEALING. NI DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF OR THE RESULTS OF THE USE OF THE TDM DLL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE AND DOES NOT WARRANT THAT THE OPERATION OF THE TDM DLL WILL BE UNINTERRUPTED OR ERROR FREE. NI EXPRESSLY DISCLAIMS ANY WARRANTIES NOT STATED HEREIN.

9. Limitation On Liability. To the maximum extent permitted by applicable law, in no event shall NI and its licensors, distributors, and suppliers (including its and their directors, officers, employees, and agents) be liable for any damages, including, but not limited to, any special, direct, indirect, incidental, exemplary, or consequential damages, expenses, lost profits, lost savings, business interruption, lost business information, or any other damages arising out of the use or inability to use the TDM DLL or related hardware even if NI or its licensors, distributors, and suppliers has been advised of the possibility of such damages. You

acknowledge that the applicable fees and prices reflect this allocation of risk. Because some

states/jurisdictions do not allow the exclusion or limitation of liability for consequential or

incidental damages, the above limitation may not apply. If the foregoing limitation of liability is

not enforceable because the TDM DLL or the hardware is determined by a court of

competent jurisdiction in a final, non-appealable judgment to be defective and to have directly

caused bodily injury, death, or property damage, in no event shall NI's liability for property

damage exceed the greater of $5000 or the license fee paid for the TDM DLL.

10. INDEMNIFICATION. YOU SHALL DEFEND, INDEMNIFY, AND HOLD NI HARMLESS FROM AND AGAINST ALL LIABILITY, DAMAGES, AND COSTS (INCLUDING COURT COSTS AND REASONABLE FEES OF ATTORNEYS AND OTHER PROFESSIONALS) ARISING OUT OF OR RESULTING FROM YOUR DISTRIBUTION OR USE OF THE TDM DLL; EVEN IF DUE IN PART TO NI'S CONCURRENT NEGLIGENCE, OTHER FAULT, OR STRICT LIABILITY.

11. Warning. (1) NI PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN. (2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS,

MALFUNCTIONS OR FAILURES ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED “SYSTEM FAILURES”). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NI’S TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NI PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NI, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NI PRODUCTS WHENEVER NI PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

12. U.S. Government Restricted Rights. If you are an agency, department, or other entity of the

United States Government (“Government”), the use, duplication, reproduction, release,

modification, disclosure or transfer of the TDM DLL, or any related documentation of any

kind, including technical data or manuals, is restricted in accordance with Federal Acquisition

Regulation 12.212 for civilian agencies and Defense Federal Acquisition Regulation

Supplement 227.7202 for military agencies. The TDM DLL is commercial computer software and the related documentation is commercial computer software documentation. The use of

the TDM DLL and related documentation is further restricted in accordance with the terms of

this Agreement, or any modification hereto. The Contractor/Manufacturer is National

Instruments Corporation, 11500 North Mopac Expressway, Austin, Texas, U.S.A., 78759-

3504.

13. Compliance. You agree to make all applicable records available for review by NI during your

normal business hours so as to permit NI (upon reasonable notice to you) to verify your

compliance with the terms and conditions of this Agreement. Further, if you are a business or

other entity, you agree that upon the request of NI or NI’s authorized representative, you will

promptly document and certify in writing to NI that your and your employees’ use of the TDM

DLL complies with the terms and conditions of this Agreement.

14. Termination. NI may terminate this Agreement at any time by delivering notice to you. You

may terminate this Agreement by destroying or erasing all copies of the TDM DLL. Further,

this Agreement shall automatically terminate upon failure by you to comply with its terms.

Upon termination of this Agreement, the license granted to you in Section 2 above will

immediately terminate and you agree to destroy or erase all copies of the TDM DLL that you

have made. However, the destruction requirement does not apply to copies distributed prior
to the effective date of termination.

15. General.

A. This Agreement is governed by the laws of the State of Texas, U.S.A., exclusive of any
provisions of the United Nations Convention on the International Sale of Goods, and
without regard to principles of conflicts of law and (ii) the non-exclusive venue for all
actions under this Agreement shall be in the courts located in Travis County, Texas,
U.S.A. and the parties agree to submit to the jurisdiction of such courts.

B. This Agreement constitutes the complete agreement between you and NI regarding the
TDM DLL and it supersedes any oral or written proposals, prior agreements, purchase
orders or any other communication between you and NI relating to the subject matter of
this Agreement.

C. If any action is brought by either party to this Agreement against the other regarding the
subject matter hereof, the prevailing party shall be entitled to recover, in addition to any
relief granted, reasonable attorney fees and court costs. If any provision of this
Agreement is held invalid, the offending clause will be modified so as to be enforceable
and, as modified, shall be fully enforced, and the remainder of this Agreement will
continue in full force and effect. If you are downloading the TDM DLL, you represent and

warrant that you are not located in or under the control of any country which the export
laws and regulations of such country or of the United States prohibit the exportation of
the TDM DLL to. Product and company names used herein are trademarks or trade names of their respective companies.

NumPy License

NumPy license

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the NumPy Developers nor the names of any contributors may be used to endorse or promote products derived

from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OpenSSL License

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the

Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

OpenStreetMap ODC Open Database License (ODbL) V 1.0

ODC Open Database License (ODbL)

Preamble

The Open Database License (ODbL) is a license agreement intended to allow users to freely share, modify, and use this Database while maintaining this same freedom for others. Many databases are covered by copyright, and therefore this document licenses these rights. Some jurisdictions, mainly in the European Union, have specific rights that cover databases, and so the ODbL addresses these rights, too. Finally, the ODbL is also an agreement in contract for users of this Database to act in certain ways in return for accessing this Database.

Databases can contain a wide variety of types of content (images, audiovisual material, and sounds all in the same database, for example), and so the ODbL only governs the rights over the Database, and not the contents of the Database individually. Licensors should use the ODbL together with another license for the contents, if the contents have a single set of rights that uniformly covers all of the contents. If the contents have multiple sets of different rights, Licensors should describe what rights govern what contents together in the individual record or in some other way that clarifies what rights apply.

Sometimes the contents of a database, or the database itself, can be covered by other rights not addressed here (such as private contracts, trade mark over the name, or privacy rights / data protection rights over information in the contents), and so you are advised that you may have to consult other documents or clear other rights before doing activities not covered by this License.

The Licensor (as defined below)

and

You (as defined below)

agree as follows:

1.0 Definitions of Capitalised Words

"Collective Database" - Means this Database in unmodified form as part of a collection of independent databases in themselves that together are assembled into a collective whole. A work that constitutes a Collective Database will not be considered a Derivative Database.

"Convey" - As a verb, means Using the Database, a Derivative Database, or the Database as part of a Collective Database in any way that enables a Person to make or receive copies of the Database or a Derivative Database. Conveying does not include interaction with a user through a computer network, or creating and Using a Produced Work, where no transfer of a copy of the Database or a Derivative Database occurs.
"Contents" - The contents of this Database, which includes the information, independent works, or other material collected into the Database. For example, the contents of the Database could be factual data or works such as images, audiovisual material, text, or sounds.

"Database" - A collection of material (the Contents) arranged in a systematic or methodical way and individually accessible by electronic or other means offered under the terms of this License.

"Database Directive" - Means Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended or succeeded.

"Database Right" - Means rights resulting from the Chapter III ("sui generis") rights in the Database Directive (as amended and as transposed by member states), which includes the Extraction and Re-utilisation of the whole or a Substantial part of the Contents, as well as any similar rights available in the relevant jurisdiction under Section 10.4.

"Derivative Database" - Means a database based upon the Database, and includes any translation, adaptation, arrangement, modification, or any other alteration of the Database or of a Substantial part of the Contents. This includes, but is not limited to, Extracting or Re-utilising the whole or a Substantial part of the Contents in a new Database.

"Extraction" - Means the permanent or temporary transfer of all or a Substantial part of the Contents to another medium by any means or in any form.

"License" - Means this license agreement and is both a license of rights such as copyright and Database Rights and an agreement in contract.

"Licensor" - Means the Person that offers the Database under the terms of this License.

"Person" - Means a natural or legal person or a body of persons corporate or incorporate.

"Produced Work" - a work (such as an image, audiovisual material, text, or sounds) resulting from using the whole or a Substantial part of the

Contents (via a search or other query) from this Database, a Derivative Database, or this Database as part of a Collective Database.

"Publicly" - means to Persons other than You or under Your control by either more than 50% ownership or by the power to direct their activities (such as contracting with an independent consultant).

"Re-utilisation" - means any form of making available to the public all or a Substantial part of the Contents by the distribution of copies, by renting, by online or other forms of transmission.

"Substantial" - Means substantial in terms of quantity or quality or a combination of both. The repeated and systematic Extraction or Re-utilisation of insubstantial parts of the Contents may amount to the Extraction or Re-utilisation of a Substantial part of the Contents.

"Use" - As a verb, means doing any act that is restricted by copyright or Database Rights whether in the original medium or any other; and includes without limitation distributing, copying, publicly performing, publicly displaying, and preparing derivative works of the Database, as well as modifying the Database as may be technically necessary to use it in a different mode or format.

"You" - Means a Person exercising rights under this License who has not previously violated the terms of this License with respect to the Database, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

Words in the singular include the plural and vice versa.

2.0 What this License covers

2.1. Legal effect of this document. This License is:

a. A license of applicable copyright and neighbouring rights;

b. A license of the Database Right; and

c. An agreement in contract between You and the Licensor.

2.2 Legal rights covered. This License covers the legal rights in the Database, including:

a. Copyright. Any copyright or neighbouring rights in the Database. The copyright licensed includes any individual elements of the Database, but does not cover the copyright over the Contents independent of this Database. See Section 2.4 for details. Copyright law varies between jurisdictions, but is likely to cover: the Database model or schema, which is the structure, arrangement, and organisation of the Database, and can also include the Database tables and table indexes; the data entry and output sheets; and the Field names of Contents stored in the Database;

b. Database Rights. Database Rights only extend to the Extraction and Re-utilisation of the whole or a Substantial part of the Contents. Database Rights can apply even when there is no copyright over the Database. Database Rights can also apply when the Contents are removed from the Database and are selected and arranged in a way that would not infringe any applicable copyright; and

c. Contract. This is an agreement between You and the Licensor for access to the Database. In return you agree to certain conditions of use on this access as outlined in this License.

2.3 Rights not covered.

a. This License does not apply to computer programs used in the making or operation of the Database;

b. This License does not cover any patents over the Contents or the Database; and

c. This License does not cover any trademarks associated with the Database.

2.4 Relationship to Contents in the Database. The individual items of the Contents contained in this Database may be covered by other rights, including copyright, patent, data protection, privacy, or personality rights, and this License does not cover any rights (other than Database Rights or in contract) in individual Contents contained in the Database. For example, if used on a Database of images (the Contents), this License would not apply to copyright over individual images, which could have their own separate licenses, or one single license covering all of the rights over the images.

3.0 Rights granted

3.1 Subject to the terms and conditions of this License, the Licensor grants to You a worldwide, royalty-free, non-exclusive, terminable (but only under Section 9) license to Use the Database for the duration of any applicable copyright and Database Rights. These rights explicitly include commercial use, and do not exclude any field of endeavour. To the extent possible in the relevant jurisdiction, these rights may be exercised in all media and formats whether now known or created in the future.

The rights granted cover, for example:

a. Extraction and Re-utilisation of the whole or a Substantial part of the Contents;

b. Creation of Derivative Databases;

c. Creation of Collective Databases;

d. Creation of temporary or permanent reproductions by any means and in any form, in whole or in part, including of any Derivative Databases or as a part of Collective Databases; and

e. Distribution, communication, display, lending, making available, or performance to the public by any means and in any form, in whole or in part, including of any Derivative Database or as a part of Collective Databases.

3.2 Compulsory license schemes. For the avoidance of doubt:

a. Non-waivable compulsory license schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

b. Waivable compulsory license schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

c. Voluntary license schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

3.3 The right to release the Database under different terms, or to stop distributing or making available the Database, is reserved. Note that this Database may be multiple-licensed, and so You may have the choice of using alternative licenses for this Database. Subject to Section 10.4, all other rights not expressly granted by Licensor are reserved.

4.0 Conditions of Use

4.1 The rights granted in Section 3 above are expressly made subject to Your complying with the following conditions of use. These are important conditions of this License, and if You fail to follow them, You will be in material breach of its terms.

4.2 Notices. If You Publicly Convey this Database, any Derivative Database, or the Database as part of a Collective Database, then You must:

a. Do so only under the terms of this License or another license permitted under Section 4.4;

b. Include a copy of this License (or, as applicable, a license permitted under Section 4.4) or its Uniform Resource Identifier (URI) with the Database or Derivative Database, including both in the Database or Derivative Database and in any relevant documentation; and

c. Keep intact any copyright or Database Right notices and notices that refer to this License.

d. If it is not possible to put the required notices in a particular file due to its structure, then You must include the notices in a location (such as a relevant directory) where users would be likely to look for it.

4.3 Notice for using output (Contents). Creating and Using a Produced Work does not require the notice in Section 4.2. However, if you Publicly Use a Produced Work, You must include a notice associated with the Produced Work reasonably calculated to make any Person that uses, views, accesses, interacts with, or is otherwise exposed to the Produced Work aware that Content was obtained from the Database, Derivative Database, or the Database as part of a Collective Database, and that it is available under this License.

a. Example notice. The following text will satisfy notice under Section 4.3:

> Contains information from DATABASE NAME, which is made available here under the Open Database License (ODbL).

DATABASE NAME should be replaced with the name of the Database and a hyperlink to the URI of the Database. "Open Database License" should contain a hyperlink to the URI of the text of this License. If hyperlinks are not possible, You should include the plain text of the required URI's with the above notice.

4.4 Share alike.

a. Any Derivative Database that You Publicly Use must be only under the terms of:

i. This License;

ii. A later version of this License similar in spirit to this License; or

iii. A compatible license.

If You license the Derivative Database under one of the licenses mentioned in (iii), You must comply with the terms of that license.

b. For the avoidance of doubt, Extraction or Re-utilisation of the whole or a Substantial part of the Contents into a new database is a Derivative Database and must comply with Section 4.4.

c. Derivative Databases and Produced Works. A Derivative Database is Publicly Used and so must comply with Section 4.4. if a Produced Work created from the Derivative Database is Publicly Used.

d. Share Alike and additional Contents. For the avoidance of doubt, You must not add Contents to Derivative Databases under Section 4.4 a that are incompatible with the rights granted under this License.

e. Compatible licenses. Licensors may authorise a proxy to determine compatible licenses under Section 4.4 a iii. If they do so, the authorised proxy's public statement of acceptance of a compatible license grants You permission to use the compatible license.

4.5 Limits of Share Alike. The requirements of Section 4.4 do not apply in the following:

a. For the avoidance of doubt, You are not required to license Collective Databases under this License if You incorporate this Database or a Derivative Database in the collection, but this License still applies to this Database or a Derivative Database as a part of the Collective Database;

b. Using this Database, a Derivative Database, or this Database as part of a Collective Database to create a Produced Work does not create a Derivative Database for purposes of Section 4.4; and

c. Use of a Derivative Database internally within an organisation is not to the public and therefore does not fall under the requirements of Section 4.4.

4.6 Access to Derivative Databases. If You Publicly Use a Derivative Database or a Produced Work from a Derivative Database, You must also offer to recipients of the Derivative Database or Produced Work a copy in a machine readable form of:

a. The entire Derivative Database; or

b. A file containing all of the alterations made to the Database or the method of making the alterations to the Database (such as an algorithm), including any additional Contents, that make up all the differences between the Database and the Derivative Database.

The Derivative Database (under a.) or alteration file (under b.) must be available at no more than a reasonable production cost for physical distributions and free of charge if distributed over the internet.

4.7 Technological measures and additional terms

a. This License does not allow You to impose (except subject to Section 4.7 b.) any terms or any technological measures on the Database, a Derivative Database, or the whole or a Substantial part of the Contents that alter or restrict the terms of this License, or any rights granted under it, or have the effect or intent of restricting the ability of any person to exercise those rights.

b. Parallel distribution. You may impose terms or technological measures on the Database, a Derivative Database, or the whole or a Substantial part of the Contents (a "Restricted Database") in contravention of Section 4.74 a. only if You also make a copy of the Database or a Derivative Database available to the recipient of the Restricted Database:

i. That is available without additional fee;

ii. That is available in a medium that does not alter or restrict the terms of this License, or any rights granted under it, or have the effect or intent of restricting the ability of any person to exercise those rights (an "Unrestricted Database"); and

iii. The Unrestricted Database is at least as accessible to the recipient as a practical matter as the Restricted Database.

c. For the avoidance of doubt, You may place this Database or a Derivative Database in an authenticated environment, behind a password, or within a similar access control scheme provided that You do not alter or restrict the terms of this License or any rights granted under it or have the effect or intent of restricting the ability of any person to exercise those rights.

4.8 Licensing of others. You may not sublicense the Database. Each time You communicate the Database, the whole or Substantial part of the Contents, or any Derivative Database to anyone else in any way, the

Licensor offers to the recipient a license to the Database on the same terms and conditions as this License. You are not responsible for enforcing compliance by third parties with this License, but You may enforce any rights that You have over a Derivative Database. You are solely responsible for any modifications of a Derivative Database made by You or another Person at Your direction. You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License.

5.0 Moral rights

5.1 Moral rights. This section covers moral rights, including any rights to be identified as the author of the Database or to object to treatment that would otherwise prejudice the author's honour and reputation, or any other derogatory treatment:

a. For jurisdictions allowing waiver of moral rights, Licensor waives all moral rights that Licensor may have in the Database to the fullest extent possible by the law of the relevant jurisdiction under Section 10.4;

b. If waiver of moral rights under Section 5.1 a in the relevant jurisdiction is not possible, Licensor agrees not to assert any moral rights over the Database and waives all claims in moral rights to the fullest extent possible by the law of the relevant jurisdiction under Section 10.4; and

c. For jurisdictions not allowing waiver or an agreement not to assert moral rights under Section 5.1 a and b, the author may retain their moral rights over certain aspects of the Database.

Please note that some jurisdictions do not allow for the waiver of moral rights, and so moral rights may still subsist over the Database in some jurisdictions.

6.0 Fair dealing, Database exceptions, and other rights not affected

6.1 This License does not affect any rights that You or anyone else may independently have under any applicable law to make any use of this Database, including without limitation:

a. Exceptions to the Database Right including: Extraction of Contents from non-electronic Databases for private purposes, Extraction for purposes of illustration for teaching or scientific research, and Extraction or Re-utilisation for public security or an administrative or judicial procedure.

b. Fair dealing, fair use, or any other legally recognised limitation or exception to infringement of copyright or other applicable laws.

6.2 This License does not affect any rights of lawful users to Extract and Re-utilise insubstantial parts of the Contents, evaluated quantitatively or qualitatively, for any purposes whatsoever, including creating a Derivative Database (subject to other rights over the Contents, see Section 2.4). The repeated and systematic Extraction or Re-utilisation of insubstantial parts of the Contents may however amount to the Extraction or Re-utilisation of a Substantial part of the Contents.

7.0 Warranties and Disclaimer

7.1 The Database is licensed by the Licensor "as is" and without any warranty of any kind, either express, implied, or arising by statute, custom, course of dealing, or trade usage. Licensor specifically disclaims any and all implied warranties or conditions of title, non-infringement, accuracy or completeness, the presence or absence of errors, fitness for a particular purpose, merchantability, or otherwise. Some jurisdictions do not allow the exclusion of implied warranties, so this exclusion may not apply to You.

8.0 Limitation of liability

8.1 Subject to any liability that may not be excluded or limited by law, the Licensor is not liable for, and expressly excludes, all liability for loss or damage however and whenever caused to anyone by any use under this License, whether by You or by anyone else, and whether caused by any fault on the part of the Licensor or not. This exclusion of liability includes, but is not limited to, any special, incidental, consequential, punitive, or exemplary damages such as loss of revenue, data, anticipated profits, and lost business. This exclusion applies even if the Licensor has been advised of the possibility of such damages.

8.2 If liability may not be excluded by law, it is limited to actual and direct financial loss to the extent it is caused by proved negligence on the part of the Licensor.

9.0 Termination of Your rights under this License

9.1 Any breach by You of the terms and conditions of this License automatically terminates this License with immediate effect and without notice to You. For the avoidance of doubt, Persons who have received the Database, the whole or a Substantial part of the Contents, Derivative Databases, or the Database as part of a Collective Database from You under this License will not have their licenses terminated provided their use is in full compliance with this License or a license granted under Section 4.8 of this License. Sections 1, 2, 7, 8, 9 and 10 will survive any termination of this License.

9.2 If You are not in breach of the terms of this License, the Licensor will not terminate Your rights under it.

9.3 Unless terminated under Section 9.1, this License is granted to You for the duration of applicable rights in the Database.

9.4 Reinstatement of rights. If you cease any breach of the terms and conditions of this License, then your full rights under this License will be reinstated:

a. Provisionally and subject to permanent termination until the 60th day after cessation of breach;

b. Permanently on the 60th day after cessation of breach unless otherwise reasonably notified by the Licensor; or

c. Permanently if reasonably notified by the Licensor of the violation, this is the first time You have received notice of violation of this License from the Licensor, and You cure the violation prior to 30 days after your receipt of the notice.

Persons subject to permanent termination of rights are not eligible to be a recipient and receive a license under Section 4.8.

9.5 Notwithstanding the above, Licensor reserves the right to release the Database under different license terms or to stop distributing or making available the Database. Releasing the Database under different license terms or stopping the distribution of the Database will not withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

10.0 General

10.1 If any provision of this License is held to be invalid or unenforceable, that must not affect the validity or enforceability of the remainder of the terms and conditions of this License and each remaining provision of this License shall be valid and enforced to the fullest extent permitted by law.

10.2 This License is the entire agreement between the parties with respect to the rights granted here over the Database. It replaces any earlier understandings, agreements or representations with respect to the Database.

10.3 If You are in breach of the terms of this License, You will not be entitled to rely on the terms of this License or to complain of any breach by the Licensor.

10.4 Choice of law. This License takes effect in and will be governed by the laws of the relevant jurisdiction in which the License terms are sought to be enforced. If the standard suite of rights granted under applicable copyright law and Database Rights in the relevant jurisdiction includes additional rights not granted under this License, these additional rights are granted in this License in order to meet the terms of this License.

pqdiflib License

pqdiflib

This project provides class libraries for reading and writing PQDIF files using C++.

Disclaimer

This open source repository contains material that may be included in, or referenced by, an unapproved draft of a proposed IEEE Standard. All material in this repository is subject to change. The material in this repository is presented "as is" and with all faults. Use of the material is at the sole risk of the user. IEEE specifically disclaims all warranties and representations with respect to all material contained in this repository and shall not be liable, under any theory, for any use of the material. Unapproved drafts of proposed IEEE standards must not be utilized for any conformance/compliance purposes. See the [LICENSE](../LICENSE) file distributed with this work for copyright and licensing information.

License

Copyright 2023 PQDIF Authors

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

See the LICENSE file distributed with this work for copyright and licensing information, the AUTHORS file for a list of copyright holders, and the CONTRIBUTORS file for the list of contributors.

SPDX-License-Identifier: Apache-2.0

Python License Agreement

A. HISTORY OF THE SOFTWARE

==========================

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see https://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see https://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF.

All Python releases are Open Source (see https://opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases.

Release	Derived from	Year	Owner	GPL-compatible? (1)

0.9.0 thru 1.2		1991-1995	CWI	yes
1.3 thru 1.5.2	1.2	1995-1999	CNRI	yes
1.6	1.5.2	2000	CNRI	no
2.0	1.6	2000	BeOpen.com	no
1.6.1	1.6	2001	CNRI	yes (2)
2.1	2.0+1.6.1	2001	PSF	no
2.0.1	2.0+1.6.1	2001	PSF	yes
2.1.1	2.1+2.0.1	2001	PSF	yes
2.1.2	2.1.1	2002	PSF	yes
2.1.3	2.1.2	2002	PSF	yes
2.2 and above	2.1.1	2001-now	PSF	yes

Footnotes:

(1) GPL-compatible doesn't mean that we're distributing Python under the GPL. All Python licenses, unlike the GPL, let you distribute a modified version without making your changes open source. The GPL-compatible licenses make it possible to combine Python with other software that is released under the GPL; the others don't.

(2) According to Richard Stallman, 1.6.1 is not GPL-compatible, because its license has a choice of law clause. According to CNRI, however, Stallman's lawyer has told CNRI's lawyer that 1.6.1 is "not incompatible" with the GPL.

Thanks to the many outside volunteers who have worked under Guido's direction to make these releases possible.

B. TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON
===

Python software and documentation are licensed under the
Python Software Foundation License Version 2.

Starting with Python 3.8.6, examples, recipes, and other code in
the documentation are dual licensed under the PSF License Version 2
and the Zero-Clause BSD license.

Some software incorporated into Python is under different licenses.
The licenses are listed with code falling under that license.

PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2
----

1. This LICENSE AGREEMENT is between the Python Software Foundation
("PSF"), and the Individual or Organization ("Licensee") accessing and
otherwise using this software ("Python") in source or binary form and
its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF hereby
grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce,
analyze, test, perform and/or display publicly, prepare derivative works,
distribute, and otherwise use Python alone or in any derivative version,
provided, however, that PSF's License Agreement and PSF's notice of copyright,
i.e., "Copyright (c) 2001-2024 Python Software Foundation; All Rights Reserved"

are retained in Python alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the Individual or Organization ("Licensee") accessing and otherwise using this software in source or binary form and its associated documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License Agreement, BeOpen hereby grants Licensee a non-exclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use the Software alone or in any derivative version, provided, however, that the BeOpen Python License is retained in the Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an "AS IS" basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT

INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all respects by the law of the State of California, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between BeOpen and Licensee. This License Agreement does not grant permission to use BeOpen trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party. As an exception, the "BeOpen Python" logos available at http://www.pythonlabs.com/logos.html may be used according to the permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

CNRI LICENSE AGREEMENT FOR PYTHON 1.6.1

----

1. This LICENSE AGREEMENT is between the Corporation for National Research Initiatives, having an office at 1895 Preston White Drive, Reston, VA 20191 ("CNRI"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 1.6.1 software in source or binary form and its associated documentation.

2. Subject to the terms and conditions of this License Agreement, CNRI hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 1.6.1 alone or in any derivative version, provided, however, that CNRI's License Agreement and CNRI's notice of copyright, i.e., "Copyright (c) 1995-2001 Corporation for National Research Initiatives; All Rights Reserved" are retained in Python 1.6.1 alone or in any derivative version prepared by Licensee. Alternately, in lieu of CNRI's License Agreement, Licensee may substitute the following text (omitting the quotes): "Python 1.6.1 is made available subject to the terms and conditions in CNRI's License Agreement. This Agreement together with Python 1.6.1 may be located on the internet using the following unique, persistent identifier (known as a handle): 1895.22/1013. This Agreement may also be obtained from a proxy server on the internet using the following URL: http://hdl.handle.net/1895.22/1013".

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6.1 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 1.6.1.

4. CNRI is making Python 1.6.1 available to Licensee on an "AS IS" basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6.1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 1.6.1 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 1.6.1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by the federal intellectual property law of the United States, including without limitation the federal copyright law, and, to the extent such U.S. federal law does not apply, by the law of the Commonwealth of Virginia, excluding Virginia's conflict of law provisions. Notwithstanding the foregoing, with regard to derivative works based on Python 1.6.1 that incorporate non-separable material that was previously distributed under the GNU General Public License (GPL), the law of the Commonwealth of Virginia shall govern this License Agreement only as to issues arising under or with respect to Paragraphs 4, 5, and 7 of this License Agreement. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or

trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the "ACCEPT" button where indicated, or by copying, installing or otherwise using Python 1.6.1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stichting Mathematisch Centrum or CWI not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES

WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

ZERO-CLAUSE BSD LICENSE FOR CODE IN THE PYTHON DOCUMENTATION

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH
REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT,
INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

snappy License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

===

Some of the benchmark data in testdata/ is licensed differently:

- fireworks.jpeg is Copyright 2013 Steinar H. Gunderson, and is licensed under the Creative Commons Attribution 3.0 license

(CC-BY-3.0). See https://creativecommons.org/licenses/by/3.0/
for more information.

- kppkn.gtb is taken from the Gaviota chess tablebase set, and
is licensed under the MIT License. See
https://sites.google.com/site/gaviotachessengine/Home/endgame-tablebases-1
for more information.

- paper-100k.pdf is an excerpt (bytes 92160 to 194560) from the paper
"Combinatorial Modeling of Chromatin Features Quantitatively Predicts DNA
Replication Timing in _Drosophila_" by Federico Comoglio and Renato Paro,
which is licensed under the CC-BY license. See
http://www.ploscompbiol.org/static/license for more ifnormation.

- alice29.txt, asyoulik.txt, plrabn12.txt and lcet10.txt are from Project
Gutenberg. The first three have expired copyrights and are in the public
domain; the latter does not have expired copyright, but is still in the
public domain according to the license information
(http://www.gutenberg.org/ebooks/53).

Somat libsie SIE Reader Library

LICENSE
=======

All files in this software distribution not otherwise marked are under
the following license:

This library is free software; you can redistribute it and/or
modify it under the terms of version 2.1 of the GNU Lesser General
Public License as published by the Free Software Foundation.

This library is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA

The full license is available in the COPYING file.

The files in the "tarballs" directory are third-party software covered
by the licenses containted within their respective archives:

* Check is licensed under the GNU Lesser General Public License,
  Version 2.1, or (at your option) any later version.

"include/sie_uthash.h" is copyright Troy Hanson, and is licensed under
a three-clause BSD license:

Copyright (c) 2003-2014, Troy D. Hanson
http://troydhanson.github.com/uthash/
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the

Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by

modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A

FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software

Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

SQLite License

License Information
===================

SQLite Is Public Domain

The SQLite source code, including all of the files in the directories listed in the bullets below are [Public Domain](https://sqlite.org/copyright.html).

The authors have submitted written affidavits releasing their work to the public for any use. Every byte of the public-domain code can be traced back to the original authors. The files of this repository that are public domain include the following:

* All of the primary SQLite source code files found in the [src/ directory](https://sqlite.org/src/tree/src?type=tree&expand)
* All of the test cases and testing code in the [test/ directory](https://sqlite.org/src/tree/test?type=tree&expand)
* All of the SQLite extension source code and test cases in the [ext/ directory](https://sqlite.org/src/tree/ext?type=tree&expand)
* All code that ends up in the "sqlite3.c" and "sqlite3.h" build products that actually implement the SQLite RDBMS.
* All of the code used to compile the [command-line interface](https://sqlite.org/cli.html)
* All of the code used to build various utility programs such as "sqldiff", "sqlite3_rsync", and "sqlite3_analyzer".

The public domain source files usually contain a header comment similar to the following to make it clear that the software is public domain.

> ~~~

The author disclaims copyright to this source code. In place of a legal notice, here is a blessing:

* May you do good and not evil.
* May you find forgiveness for yourself and forgive others.

 *   May you share freely, never taking more than you give.
~~~

Almost every file you find in this source repository will be
public domain. But there are a small number of exceptions:

Non-Public-Domain Code Included With This Source Repository AS A Convenience
----

This repository contains a (relatively) small amount of non-public-domain
code used to help implement the configuration and build logic. In other
words, there are some non-public-domain files used to implement:

> ~~~
./configure && make
~~~

In all cases, the non-public-domain files included with this
repository have generous BSD-style licenses. So anyone is free to
use any of the code in this source repository for any purpose, though
attribution may be required to reuse or republish the configure and
build scripts. None of the non-public-domain code ever actually reaches
the build products, such as "sqlite3.c", however, so no attribution is
required to use SQLite itself. The non-public-domain code consists of
scripts used to help compile SQLite. The non-public-domain code is
technically not part of SQLite. The non-public-domain code is
included in this repository as a convenience to developers, so that those
who want to build SQLite do not need to go download a bunch of
third-party build scripts in order to compile SQLite.

Non-public-domain code included in this respository includes:

* The ["autosetup"](http://msteveb.github.io/autosetup/) configuration system that is contained (mostly) the autosetup/ directory, but also includes the "./configure" script at the top-level of this archive. Autosetup has a separate BSD-style license. See the [autosetup/LICENSE](http://msteveb.github.io/autosetup/license/) for details.

* There are BSD-style licenses on some of the configuration software found in the legacy autoconf/ directory and its subdirectories.

The following unix shell command is can be run from the top-level of this source repository in order to remove all non-public-domain code:

> ~~~
rm -rf configure autosetup autoconf
~~~

If you unpack this source repository and then run the command above, what is left will be 100% public domain.
~~~

TDM DLL License

TDM DLL is used under license from National Instruments Corporation;

copyright (c) 2017 National Instruments Corporation

Extraction and redistribution of the TDM DLL included in the FlexPro package is prohibited by the copyright owner.

TinyEXIF License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TinyXML2 License

This software is provided 'as-is', without any express or implied

warranty. In no event will the authors be held liable for any

damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

Trio License

Trio is a package with portable string functions. Including printf() clones and others.

Copyright (C) 1998-2001 by Bjorn Reese and Daniel Stenberg.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF

MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE AUTHORS AND CONTRIBUTORS ACCEPT NO RESPONSIBILITY IN ANY CONCEIVABLE MANNER.

Trio is intended to be an integral part of another application, so we have not done anything to create a proper installation.

Compile with 'make' (edit the Makefile if you want a release build)

Test the package with 'make test'

Install by copying trio.h, triop.h, and libtrio.a (and man/man?/* if you want documentation) to the appropriate directories.

Catch some usage examples in example.c

Send feedback and patches to the mailing list, subscription and other information is found here:

http://lists.sourceforge.net/lists/listinfo/ctrio-talk

Enjoy!

Trio web page

http://daniel.haxx.se/trio/

Winsparkle License

for use in the OpenSSL Toolkit (http://www.openssl.org/).

Xerces Apache License V 2.0

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the

origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only

on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,

zlib Library License

Copyright notice:

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

ZSTD License

ZSTD is dual licensed under BSD and GPLv2.

LICENSE:

BSD License

For Zstandard software

License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying

the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING

OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

Index

- " -

- 1 -

- 2 -

- 3 -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- J -

- K -

- L -

- M -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -

- W -

- Z -

Dated 1/27/2026, subject to errors and changes.

WEISANG