



Manual FBox Builder (Basic Version)

0	Content	
0.1	Document History	0-3
0.2	About this manual	0-3
0.3	Brands and trademarks	0-3
1	Overview	
1.1	Main Software Components.....	1-1
1.1.1	Saia PG5® FBox Library Manager.....	1-1
1.1.2	Message Window.....	1-2
1.1.3	Saia PG5® FBox Workspace.....	1-2
2	Saia PG5® FBox Builder Hierarchy	
2.1	The Project Level.....	2-1
2.2	The Library Level.....	2-1
2.3	The Family Level.....	2-3
3	Saia PG5® FBox Basics	
3.1	Saia PG5® FBox Development Process	3-1
3.2	Create a Project.....	3-1
3.2.1	Project Tips	3-2
3.3	Create a Library	3-3
3.3.1	Library Properties.....	3-4
3.4	Create a Family.....	3-5
3.4.1	Family Properties	3-7
3.5	FUPLA Page(s) Import.....	3-8
3.5.1	Program the functionality	3-9
3.5.2	Export the FUPLA Page	3-9
3.5.3	Import the Page(s) in the Saia PG5® FBox Builder	3-10
3.5.4	Check the Resulting Saia PG5® FBox.....	3-12
3.5.5	Saia PG5® FBox Workspace.....	3-12
3.5.6	Create the Saia PG5® FBox Help Topic	3-13
3.6	Generate Output Files.....	3-16
3.7	Documentation Generator.....	3-18
3.8	Add-Tools to Saia PG5® FBox Builder	3-20
4	Saia PG5® FBox Advanced	
4.1	Library Information	4-1
4.1.1	Licensed Saia PG5® FBox Library Install Package	4-3
4.2	Multi-Languages Handling	4-5
4.2.1	Language Editor.....	4-6
4.3	Backup/Restore Mechanism	4-8
4.4	Library Maintenance Handling	4-11
4.4.1	Active	4-11
4.4.2	Renamed.....	4-11
4.4.3	Removed.....	4-11
4.4.4	Replaced.....	4-11
4.4.5	Must be replaced.....	4-12
4.4.6	Obsolete.....	4-12
4.5	Debugging an Saia PG5® FBox	4-12

A	Appendix	
A.1	Icons	A-1
A.2	Contact	A-2



0.1 Document History

Date	Version	Changes	Remarks
2004-09-08	pEN2	-	Preliminary Edition
2004-09-09	EN02	-	Published Edition
2006-02-15	EN06	-	Major modifications and update to Saia PG5® version 1.4
2013-10-16	EN07	-	New logo and new company name

0.2 About this manual

See the section in the appendix in relation to some of the terms, abbreviations and the references used in this manual.

0.3 Brands and trademarks

Saia PCD® and Saia PG5® are registered trademarks of Saia-Burgess Controls AG.

Windows 95/98, Windows NT, Word, Excel, PowerPoint, FrontPage and Microsoft Internet Explorer are registered trademarks of The Microsoft Corporation.

Netscape Navigator is a registered trademark of The Netscape Communications

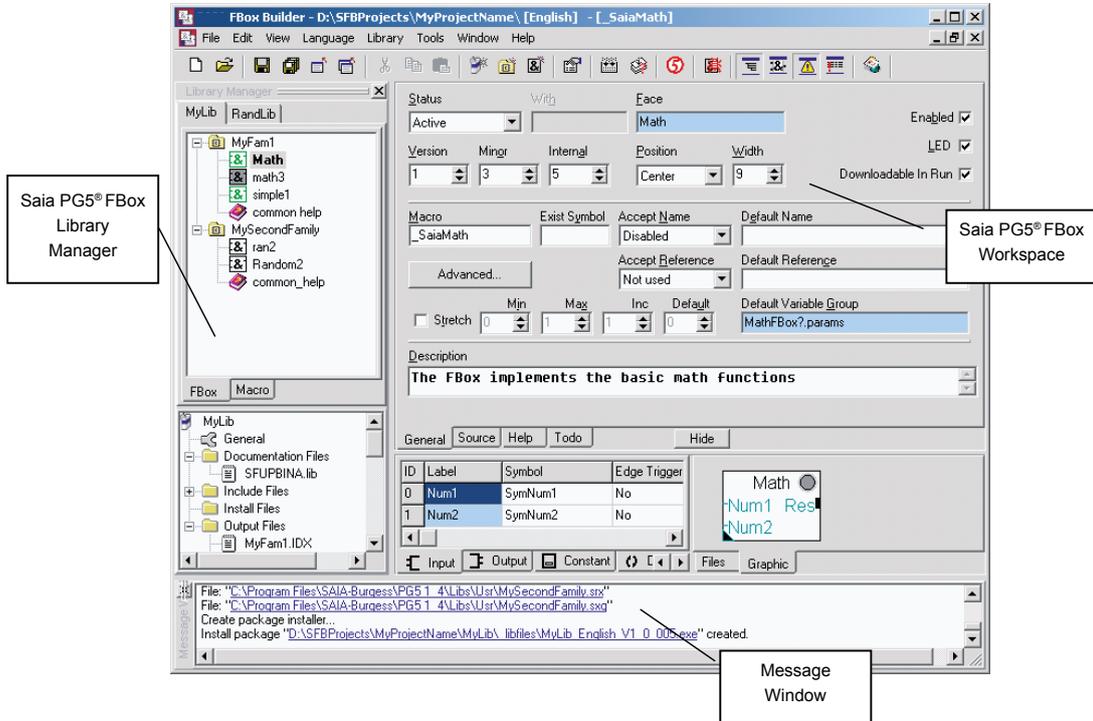
Technical modifications are based on the current state-of-the-art technology.

Saia-Burgess Controls AG, 2004. © All rights reserved.

Published in Switzerland

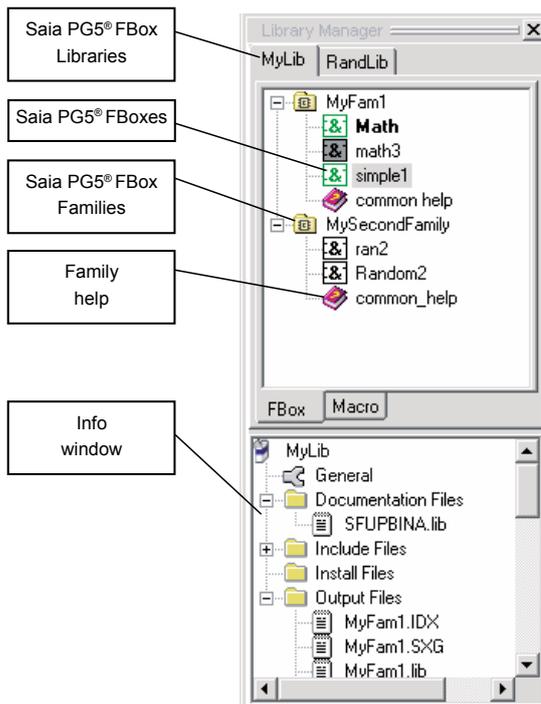
1 Overview

1.1 Main Software Components



1.1.1 Saia PG5® FBox Library Manager

The beginning of every operation starts here: create a library, create a family, create/import/modify an FBox.



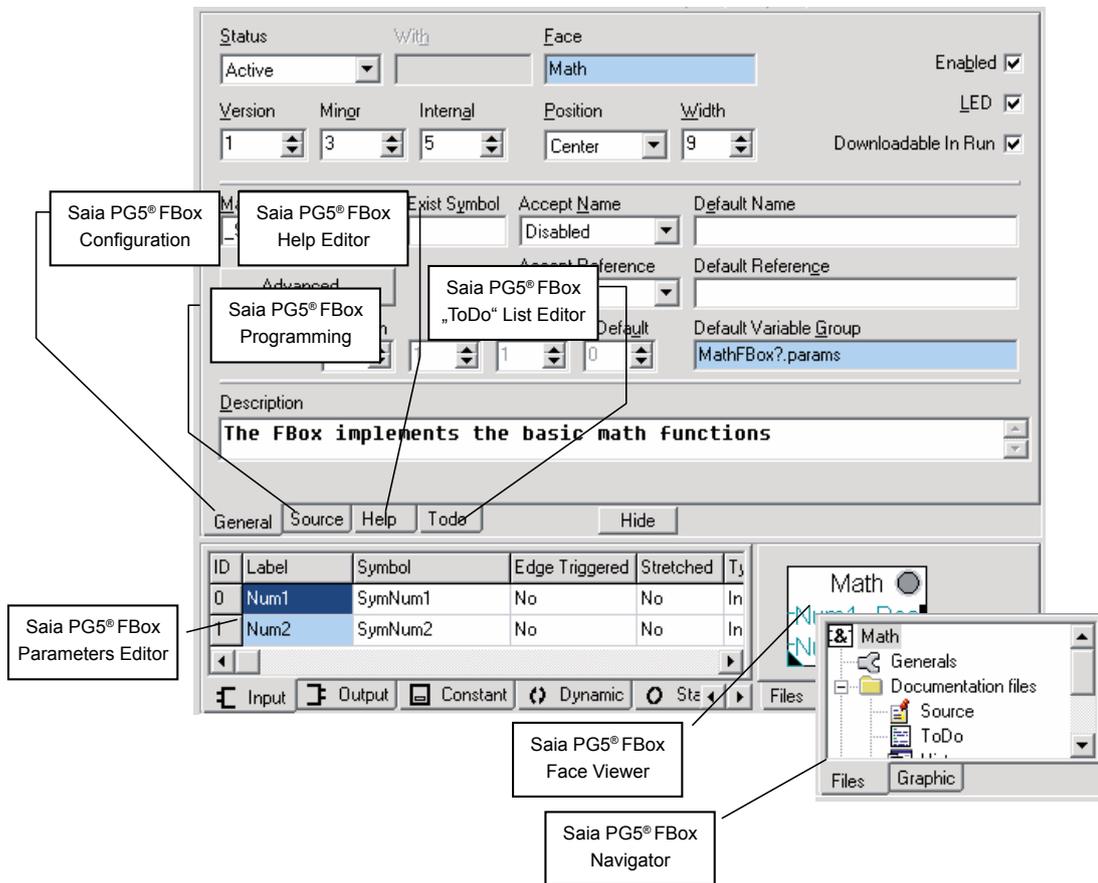
1.1.2 Message Window

Compilation Errors and Warnings will be displayed in the Message Window. In the case of an error, you will find the mistake in double clicking the message.



1.1.3 Saia PG5® FBox Workspace

The Workspace belongs to one FBox and allows their full configuration and programming.



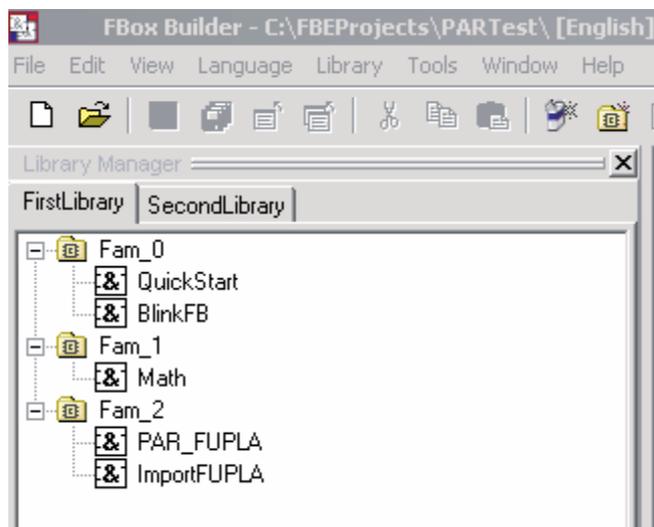
2 Saia PG5® FBox Builder Hierarchy

2.1 The Project Level

The FBox Builder is organised in projects. The project level is the start point of any activity with the FBox Builder. Basically a project is the only level that belongs to a certain user. The user can organise freely his project without any constraints regarding FUPLA.

A project defines a working directory where all the data needed by the FBox Builder are stored. An output directory must also be given by the user to tell where the files generated by the compilation must be placed.

The project contains one or more FBox Libraries, which contain one or more FBox Families, which themselves finally contain one or more FBoxes.



As you can see in the title bar of the above picture, the language of the FBox Library belongs to the project level.

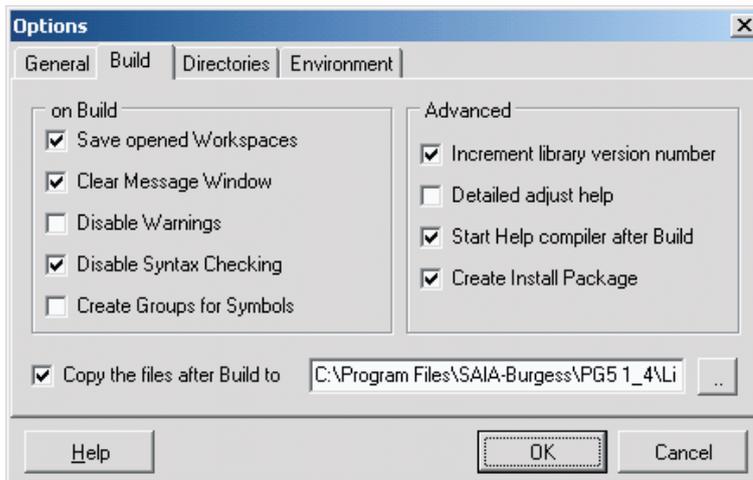
2.2 The Library Level

The FBox Library contains FBox Families. The Library level of the FBox Builder is the same as the Library level defined in FUPLA. The FUPLA libraries are the following:

- Standard Library
- Application Library
- Ladder Library
- User Library

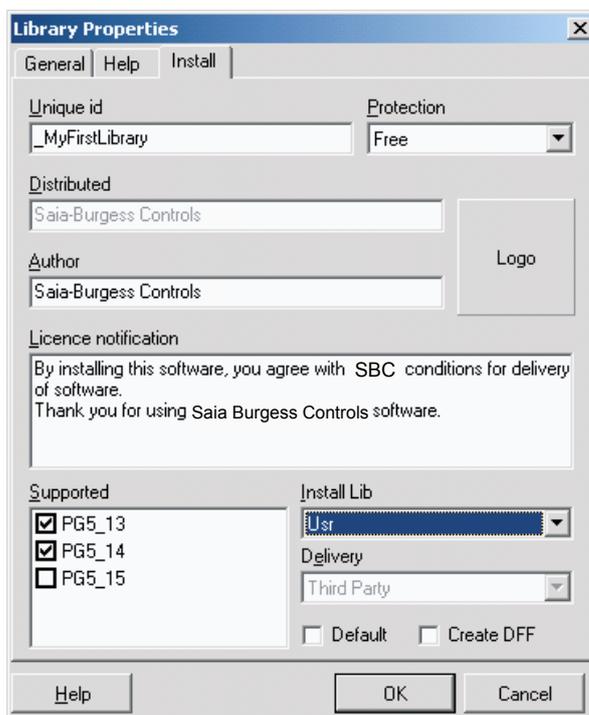
When a library is developed, the user can move the file to the FUPLA library in the "...\\libs\\usr" directory and the library will appear in the FBox Selector of FUPLA.

For example in the menu “View → Options” there is the possibility to choose and copy the Library files in the Saia PG5® library directory of your choice.



2

Or you are able to create an install package if you want to distribute your library, in that case the target directory is defined in the Library properties dialog.



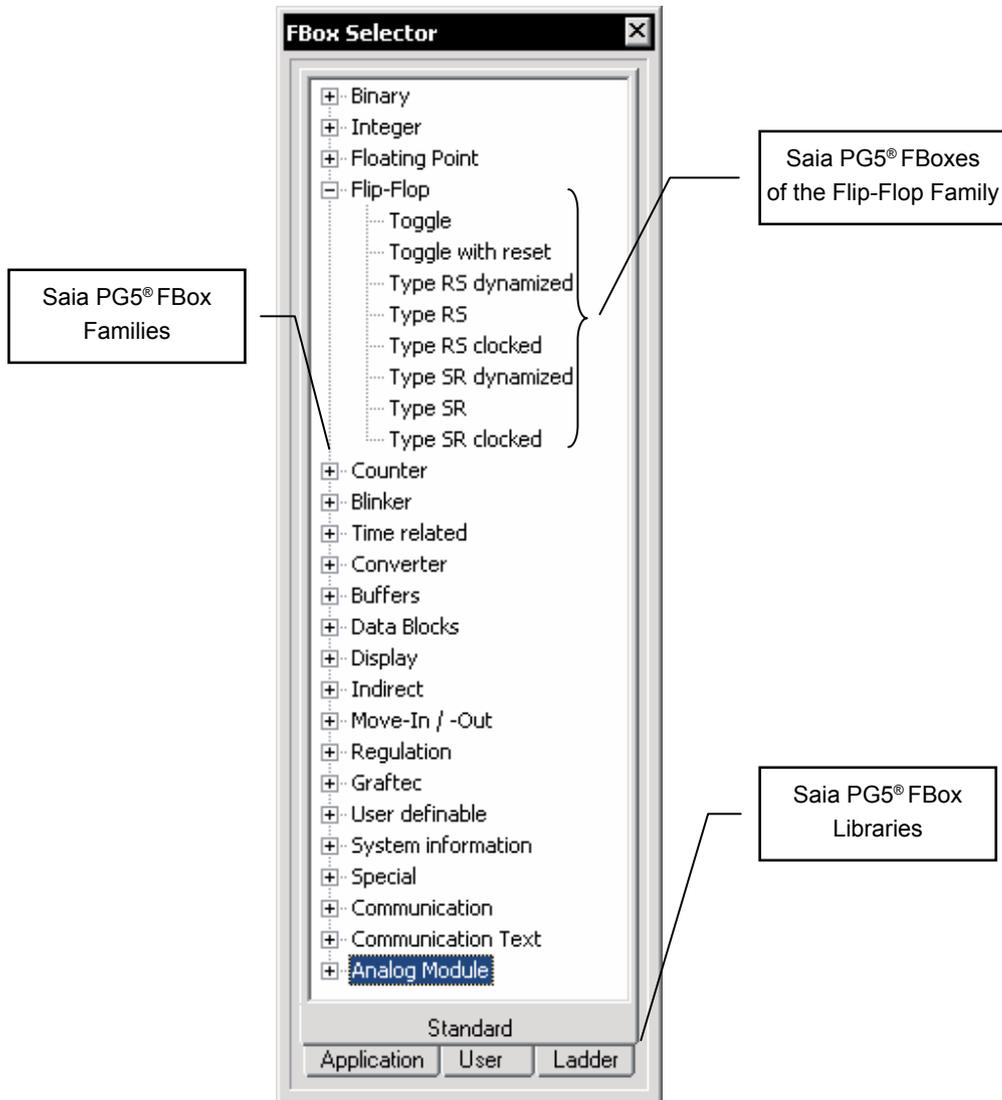
The compiler works at the Library level; so when you compile, you compile the whole FBox Library.

2.3 The Family Level

The FBox Family contains FBoxes. The family level of the FBox Builder is the same as the Library level defined in FUPLA.

2

For example, this is the family list of the FUPLA Standard library:



3 Saia PG5® FBox Basics

In this chapter you will see with an example how to make a simple FBox with the FBox Builder. The “Adjust and View” variables and the language handling will be treated in the next chapter.

3.1 Saia PG5® FBox Development Process

To develop an FBox Library, there are some steps to follow. This paragraph roughly describes the main steps to create an FBox Library.

With FBox Builder Basic Version, the first step is to build (a) FUPLA page(s) with the wanted functionality using the pre-defined FBox Builder tags. This FUPLA page(s) must be tested in FUPLA.

Then the second step is to export this page in FUPLA and then to import it in the FBox Builder.

At that stage, your FBox is functional. As third step you can create a “Help File” using the “Help Generator” of the FBox Builder. This help file is based on the comment you will enter for each FBox parameter, the help text you have entered for the Library and the Family and finally on the text, picture and link you add in the “Help” tab that belongs to the FBox.

It is now time as a fourth step to test your Library, to do so, you have to build it and copy the FBox files to the FUPLA directory “USR”. If you check in the option dialog “Copy the files after build to:” and “Start Help compiler after build” all the necessary files will be copied automatically to the right directory. Finally open FUPLA, create a small test program with your new FBoxes and check them out.

The final step is the deployment of your Library. If you check the option “Create Install Package”, after a successful build an “EXE” file will be created. This file is created based on the Library Information you gave in the “Library properties” dialog in the FBox Builder. This file will simply copy the FBox files in the directory chosen.

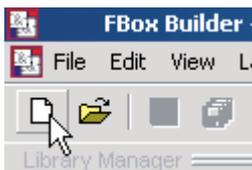
3.2 Create a Project

The project is the highest level of the FBox Builder. A project can contain several FBox Libraries, FBox Families and single FBoxes in different languages.

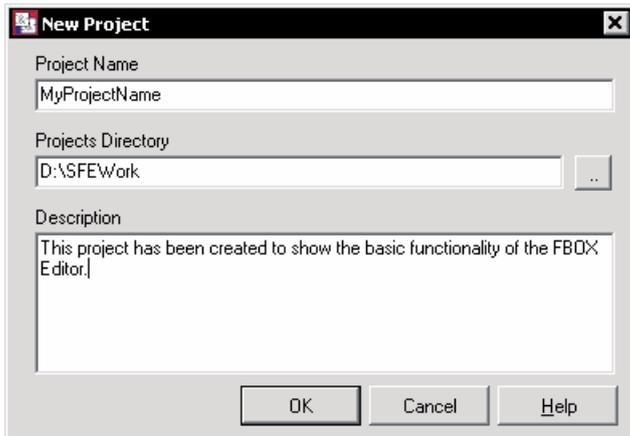
- To create a project you have to choose the following menu:



- Or you can directly click on the following tool bar button:



- The following dialog appears:



- **Project Name:** This entry is used by the FBox Builder as a name for the root directory of all the data which belong to this project
- **Projects Directory:** This entry is used by the FBox Builder as the root directory for the projects. For each new project you are creating a new root directory can be entered; this gives you another chance to organise your FBoxes
- **Description:** This entry is only used internally, it will never appear in a help file or what ever output files.

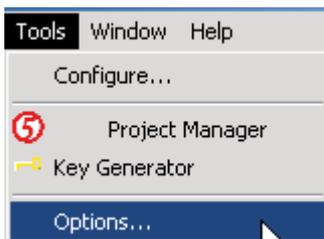
When you have filled up all the entries, click on the OK button to accept or on Cancel to abort the current task.

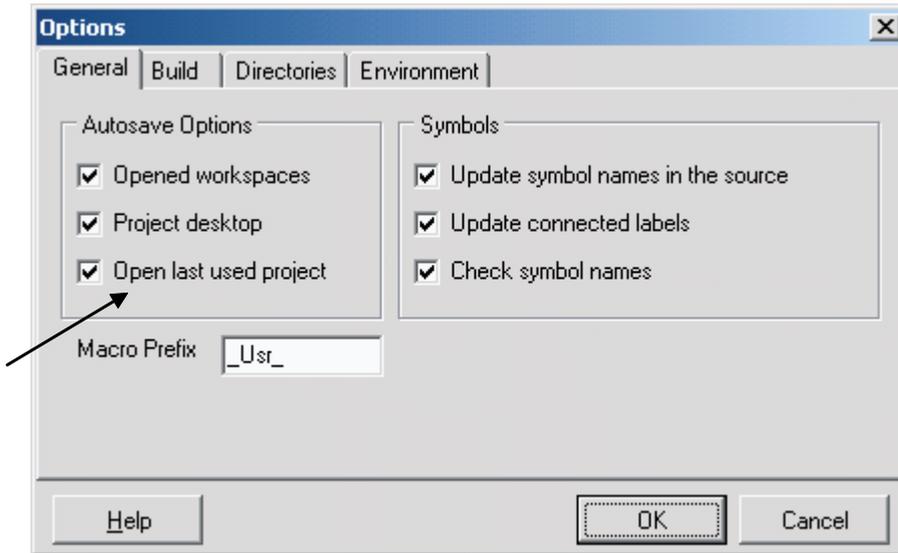


All the languages must be added when creating a project and should never be deleted. The Restore mechanism is not able to restore more languages than the “Actual” project languages number.

3.2.1 Project Tips

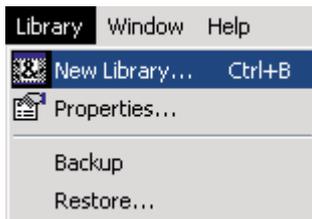
- It is possible to choose in the option menu to open always the last used project:





3.3 Create a Library

If you are currently creating a new project, the following dialog will automatically appear when you have clicked the OK button of the “New Project Dialog”. But if you want to add a new library to an existing project, you have to use the following menu:



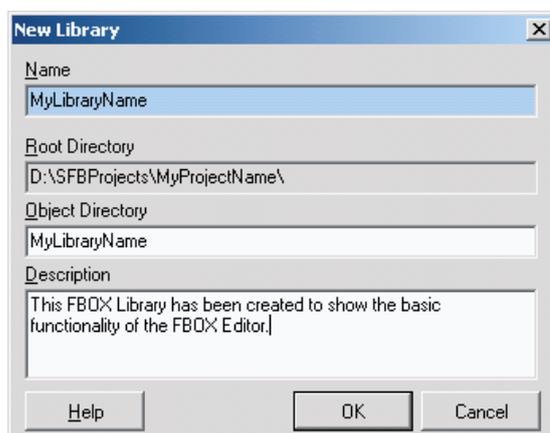
Or you can use the tool bar button:



Or in right clicking on an existing library tab of the Library Manager, you get the following popup menu:



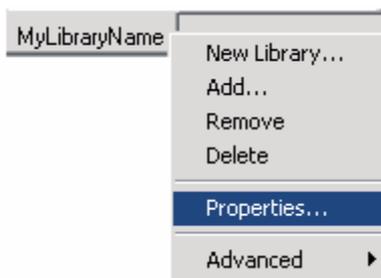
You will get the following dialog:



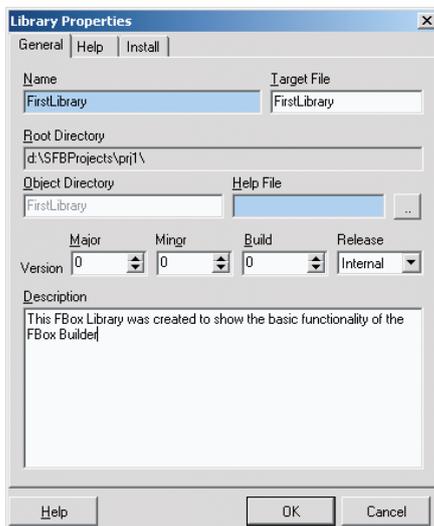
- **Name:** the FBox Builder will use this entry as file name when generating the Library related files
- **Root Directory:** The library you are creating will be placed under the project directory you just have created
- **Object Directory:** This entry is used by the FBox Builder as root directory name for the Library. By default the library directory has the same name as the library
- **Description:** This entry is used to describe the library. It is only used internally; it will not be part of the help file generated by the FBox Builder.

3.3.1 Library Properties

All informations that belong to the library can be edited in the “Library Properties” dialog. You can obtain this dialog by right clicking on the Library tab and choose the “Properties...” menu:



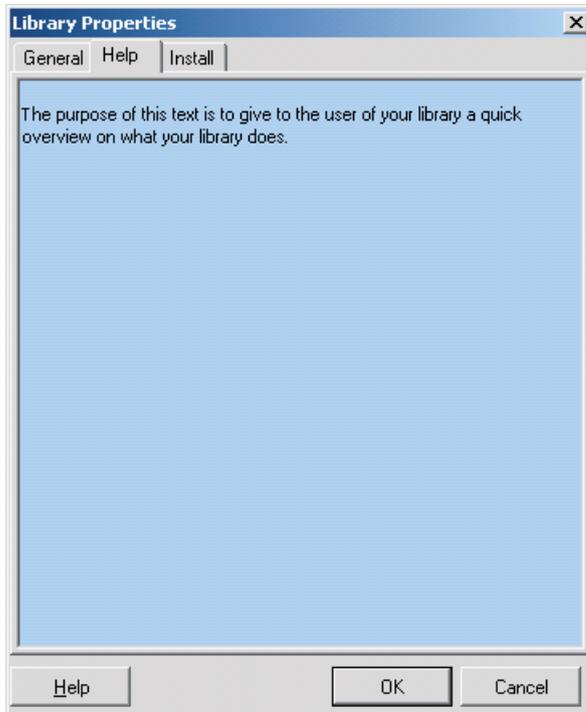
You will get this dialog:



- **Name:** This entry is the Library name previously entered. The library name can be modified; the library tab of the “Library Manager” will be adapted
- **Target File:** Specify here the library filenames (.LIN)
- **Root Directory:** The location of the current library. This field is read-only
- **Version:** This entry is used to manage the modification of the library. Each time you have made changes in your library, you should increment the version. In this way you are able to keep track of your modifications in the release notes generated by the FBox Builder

- **Release:** Internal means that the version will begin with ,'\$' sign. Beta means that the version will begin with ,b' sign. Official means that the version will begin with ,v' sign
- **Description:** This is the library description previously entered.

We'll now have a look at the "Help" tab of this dialog:



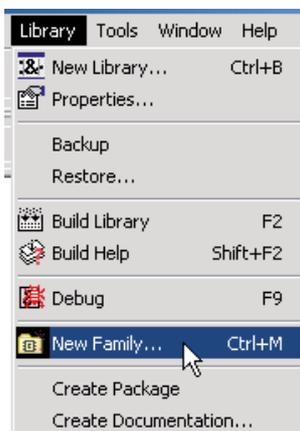
3

This tab is very simple; the text you'll write will be displayed as "Library Overview" in the help file generated by the "Help Generator" of the FBox Builder.

In this chapter, we will not treat "Install" tab. Just for your information, the "Install" tab contains all the data needed by the "Library Installer" of the FBox Builder. This feature will be explained later in the "FBox Advanced" chapter.

3.4 Create a Family

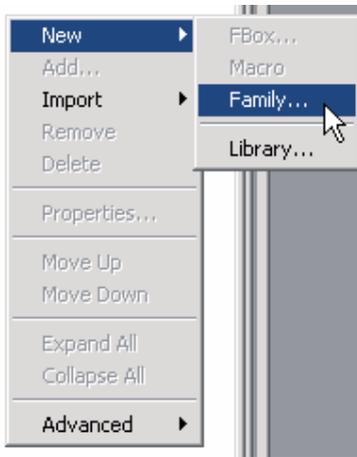
If you are currently creating a brand new project, the following dialog will automatically appear when you clicked on the OK button of the "New Library" Dialog. But If you want add a new family to an existing project, you have to use the following menu:



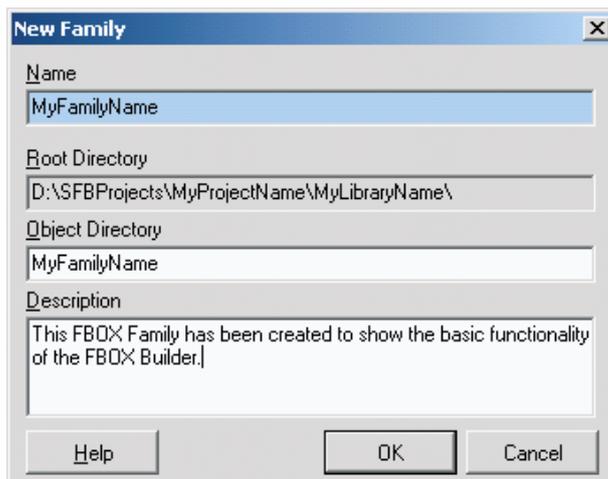
Or you can use the tool bar button:



Or in right clicking on an existing family folder of the Library Manager, you get the following popup menu:

**3**

You will get the following dialog:



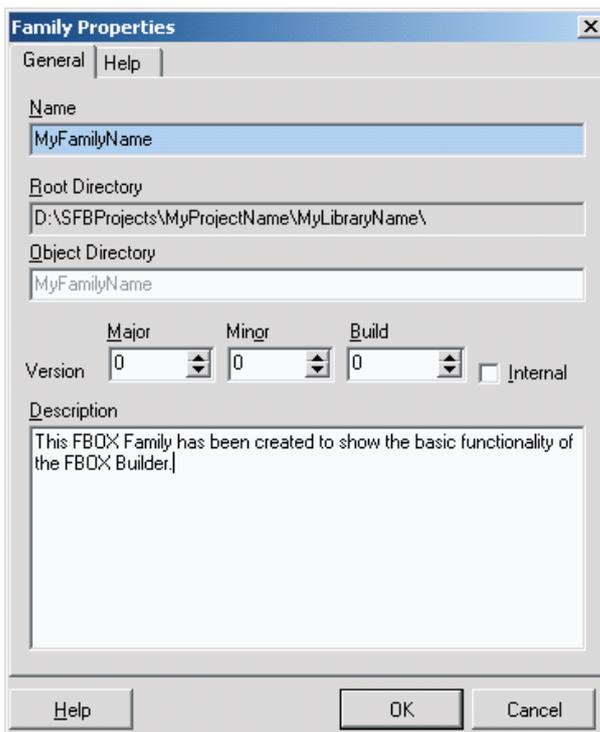
- **Name:** the FBox Builder will use this entry as file name when generating the Family related files
- **Root Directory:** The Family you are creating will be place under the Library directory you just have created
- **Object Directory:** This entry is used by the FBox Builder as root directory name for the Family. By default the Family directory has the same name as the Family
- **Description:** This entry is used to describe the Family. It is only used internally; it will not be part of the help file generated by the FBox Builder.

3.4.1 Family Properties

All information that belongs to the family can be edited in the “Family Properties” dialog. You can obtain this dialog by right clicking on the Family folder and choose the “Properties...” menu:



You will get this dialog:

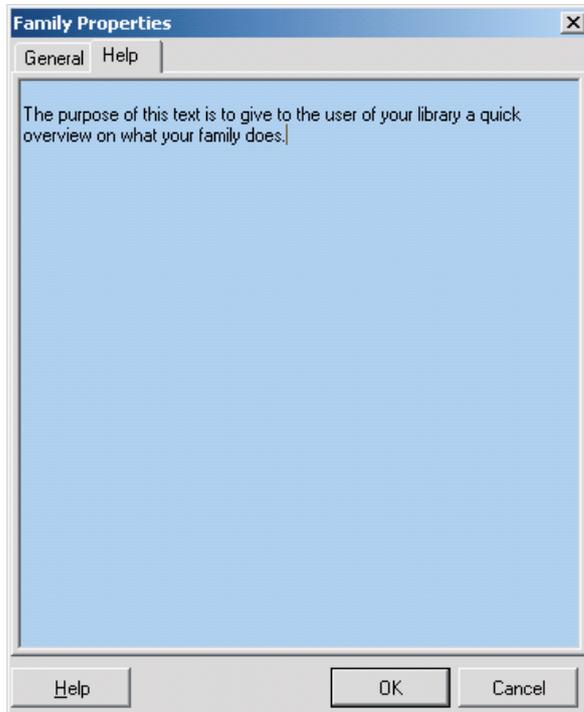


- **Name:** This entry is Family name previously entered. The Family name can be modified; the family folder name of the “Library Manager” will be adapted but not the family directory itself.
- **Root Directory:** The location of the current family. This field is read-only.
- **Version:** This entry is used to manage the modification of the family. Each time you have to make changes in your family, you should increment the version. This way you can keep track of your modifications in the release notes generated by the FBox Builder.
- **Internal:** If this check box is checked, the “Library Installer” of the FBox

Builder will display during the installation the “\$” character before the version number

- **Description:** This is the family description previously entered.

We’ll now have a look at the “Help” tab of this dialog:

**3**

This tab is very simple; the text you’ll write will be displayed as “Family Overview” in the help file generated by the “Help Generator” of the FBox Builder.

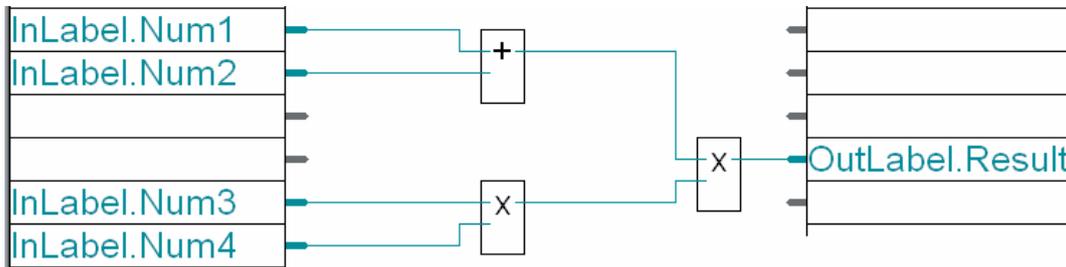
3.5 FUPLA Page(s) Import

The FBox Builder gives you the chance to develop your own FBox without writing a single line of instruction list code. The whole functionality of your FBox can be based on one or several FUPLA page(s). The example described in this paragraph is based on the “Math” FBox developed all along this manual.

You have to keep in mind that it is not meant to be a “full program compressor”; it is not reasonable to take an existing program of twenty pages and directly import it in the FBox Builder. To work correctly it should never exceed 3 pages and the maximum nested level should not exceed 5. By nested levels, it’s meant to import a page, make another page with the resulting FBox and re-import it and so on...

3.5.1 Program the functionality

Open your FUPLA and program the following page:



3

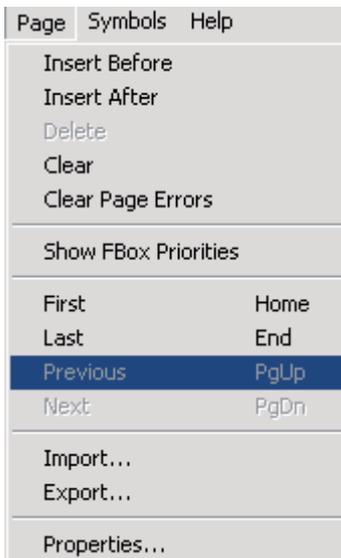
You see on this page the groups “InLabel” and “OutLabel” which are used here as tags. The FBox Builder will read these tags during the import process to place these parameters at the right place in the “Parameter Editor”.

Other tags are defined for the FUPLA page(s) import features:

- “InLabel” tag for FBox inputs variables
- “OutLabel” tag for FBox outputs variables
- “StaLabel” tag for FBox static variables
- “ConstLabel” tag for FBox constant variable
- Not tagged means dynamic variables!

3.5.2 Export the FUPLA Page

After compilation and test of this page, the functionality is correct it’s possible to export it. Select the “Export...” menu of FUPLA.



FUPLA asks for a name of this file and gives the opportunity to export one or several pages. For our example, select only the current page. FUPLA now creates a “FXP” file that contains all the data needed to re-create the page. The FBox Builder uses as input data the following file:

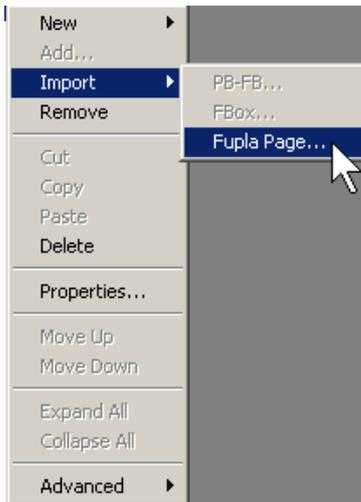
- “FXP” file which is the result of the export page

In the “Import” process of the FBox Builder starts FUPLA to compile the FXP file. The result of this compilation is mainly a “FBD” file that the FBox Builder needs to define the behaviour of the imported page.

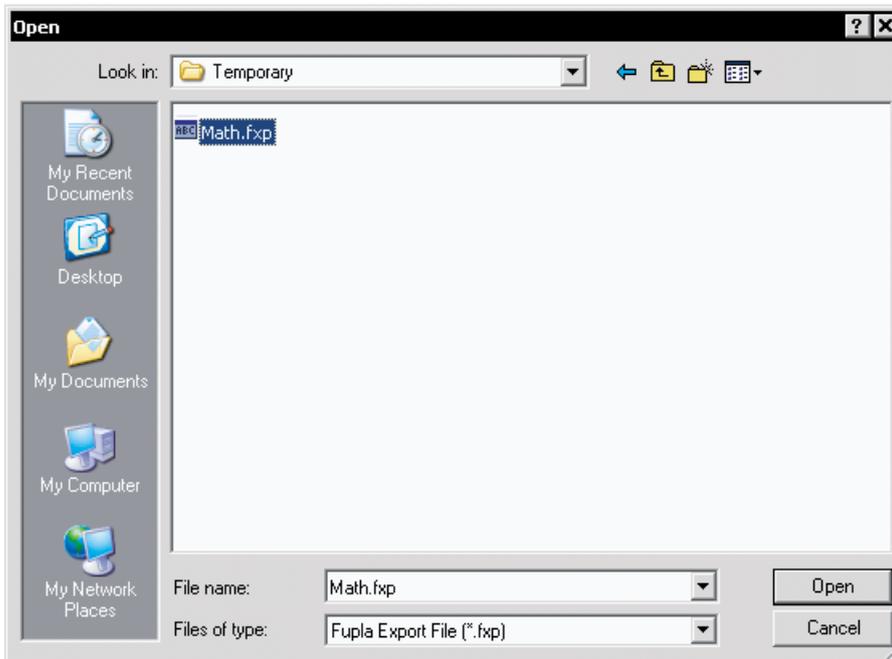
3.5.3 Import the Page(s) in the Saia PG5® FBox Builder

To create an FBox based on exported FUPLA page(s) is quite straightforward. The only thing you have to do is to click right on the family where you want to create this new FBox and choose the menu “Import → ZIP” as follow:

3

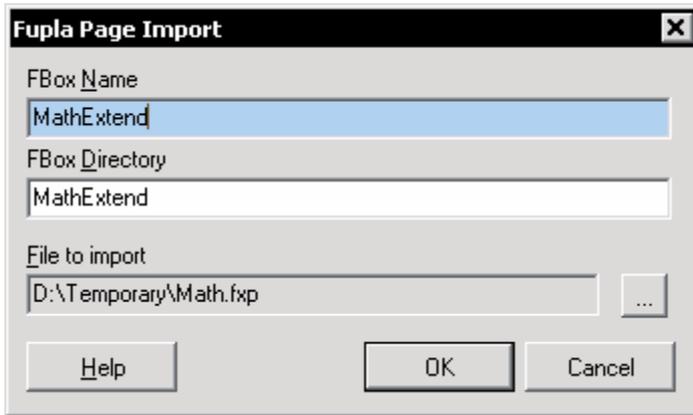


You will get the following “New FBox” dialog; you just have to browse for your created “FXP” file.



Click on the Open button to validate your entries or on Cancel to abort the process you began.

Give a name for your new FBox for example “MathExtend”; the FBox Builder will create a workspace for the FBox.

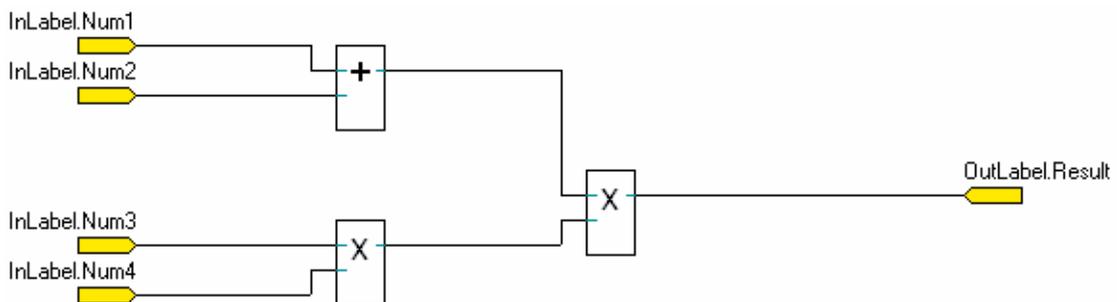


3

- **Saia PG5® FBox Name:** The name of the FBox is very important, it will be used as default to be displayed on the FBox Face, the “Help Generator” will use it as topic keyword and finally the FBox Builder will use this name as default directory name to store the FBox data.
- **Directory:** The FBox you are creating will be placed under the Family directory you just have created.
- **File to import:** This is the path of the FXP import file.

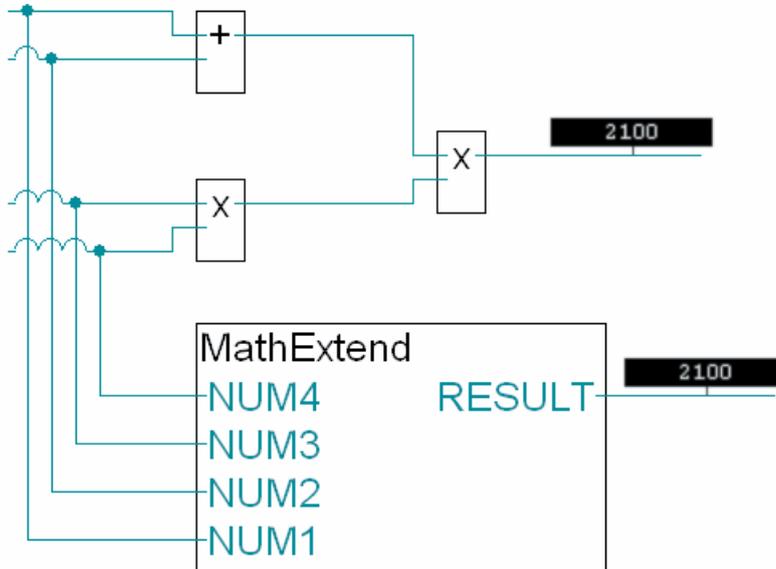
If you have a look at the “Parameter Editor”, you will see the inputs and the output we have tagged are at the right place and you will see a whole bench of variables of three letters in the static and adjust tab. It is now your work to move the variables from one tab to another so as needed.

You can see the scheme of the imported page(s) under the Schematic tab of the workspace; our example looks like this:



3.5.4 Check the Resulting Saia PG5® FBox

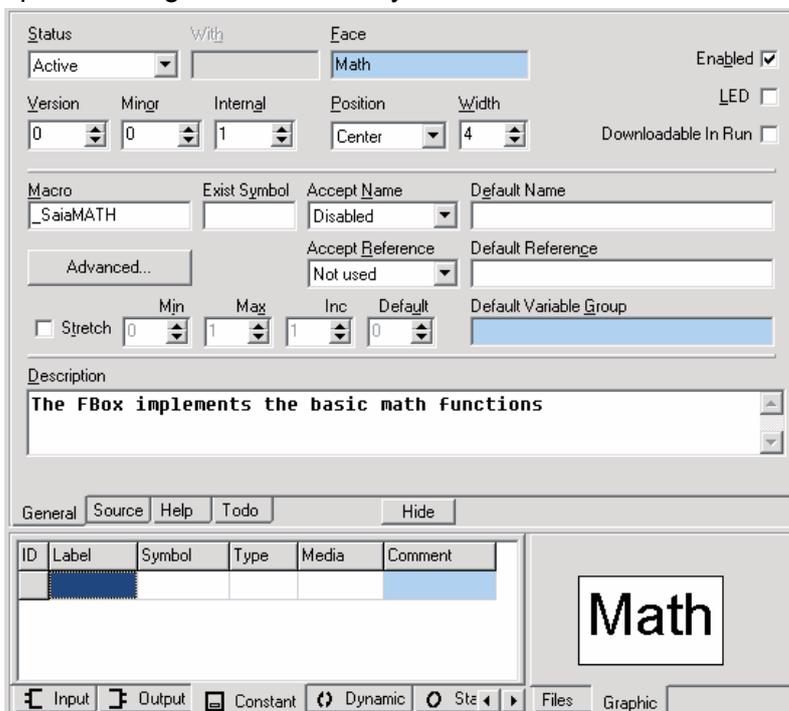
To be sure the import functionality has worked correctly, compile your library and place the “ZIP” FBox on the same page you have exported, with the same entry values you must get the same results as shown in the picture below:



3

3.5.5 Saia PG5® FBox Workspace

The workspace is basically the tool to program and configure an FBox. The workspace belongs to one and only one FBox.



The upper part contains the following tabs:

- **General:** where you define the graphical interface of the FBox
- **Source:** where you edit the code (functionality) of the FBox
- **Help:** where you edit the Help topic that belongs to that FBox
- **TODO:** where the list of task which have to be done on the FBox.

The lower part contains on the left side the “Parameter Editor” where you define the interface and the data needed by the FBox. The right part of the workspace is called the “FBox Navigator” and has the following tabs:

- **Files:** where you see the FBox attached files and where you can navigate into the workspace in double clicking on the node you want
- **Graphic:** where you see the graphical face of the FBox you are building.

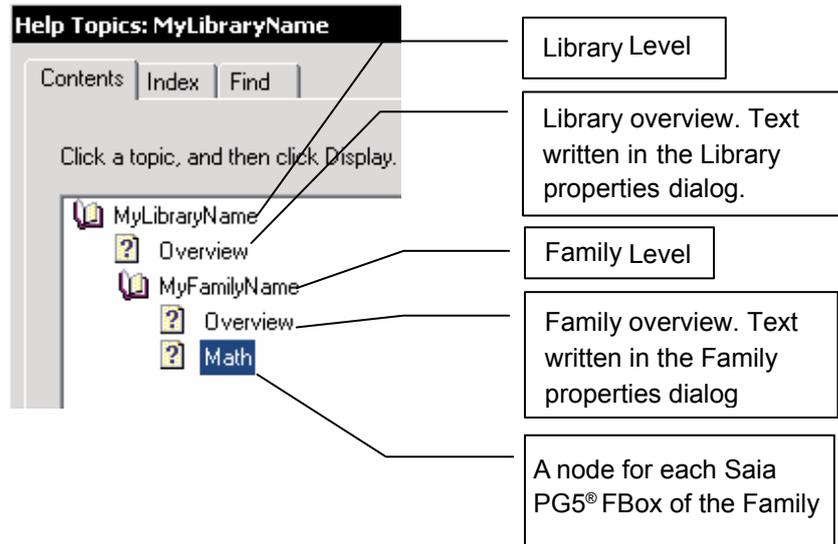


The first thing to do is to check the macro name of our FBox because this name must be unique. A good habit is to use your company name as prefix of the macro name or at least an abbreviation like for example “SBC”; so the macro name of our FBox will be “SBCMath”

3.5.6 Create the Saia PG5® FBox Help Topic

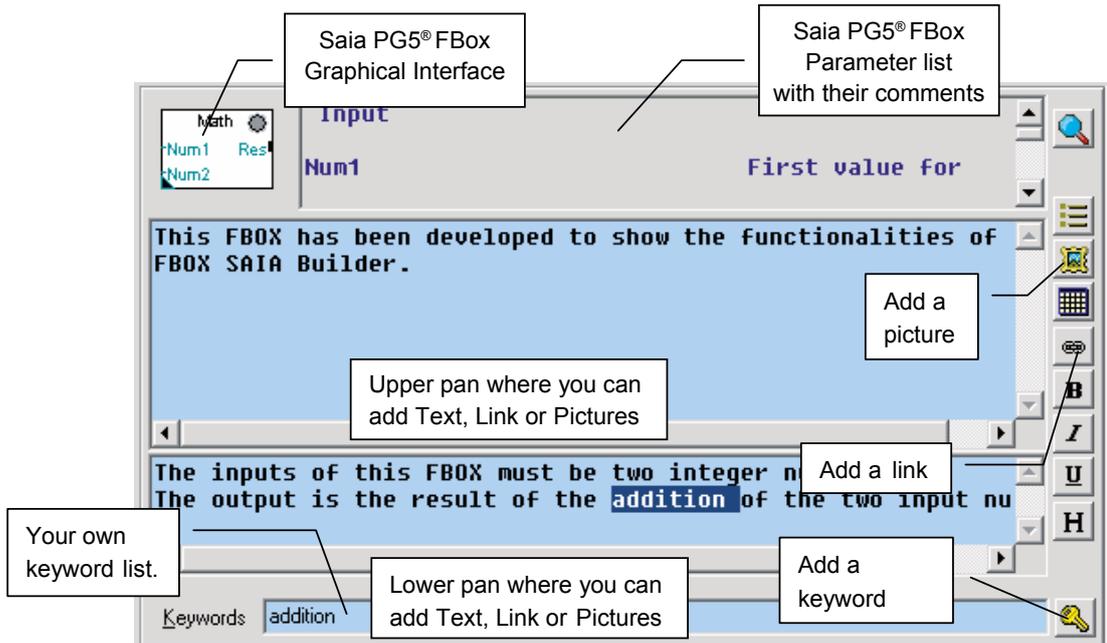
As we have seen in the previous paragraph, the FBox Builder is able to generate a help file that belongs to the library level.

The help file has a structure that is pre-defined by the FBox Builder, which looks like this:



The Library and the Family level are very simple, basically the only hand you have on them is the text you have entered.

For the FBox level a structure is pre-defined but you have quite a lot of possibilities to customize your help. Click on the “Help”  tab of the workspace:



Add a new Keyword

The default keyword is the Macro Name and the FBox Name, one of them are used by FUPLA to call the right help topic according to the current FBox.

But you can add some other keywords that you will use inside the help to make your own link to it. To add a new keyword, you type directly in the keyword edit field or you select a word in the help editor and then click on the  button.

Add a new Link

There exist links which are automatically generated, like the link between the FBox and the Family or between the Family and the Library.

But you are able to add your own links between what ever you want. To do so, you have to write text you want to display as link, select it, click on the  button and finally write the topic ID you want to link to.

The topics ID are defined as follow: IDH_Keyword

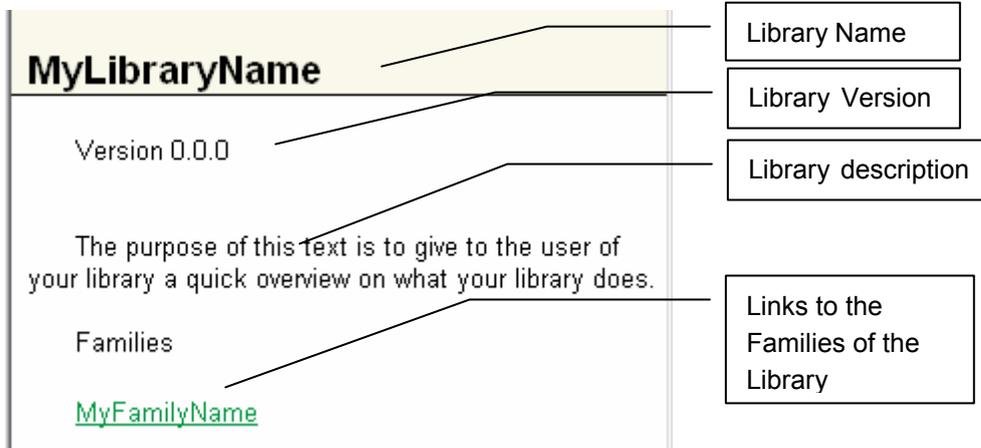
For example the link to our library looks like this: IDH_MyLibraryName

Add a New Picture

It's possible to add a picture you want to show in the Upper or lower tab, simply place the cursor where you want insert the picture and click on the  button and then browse to your picture.

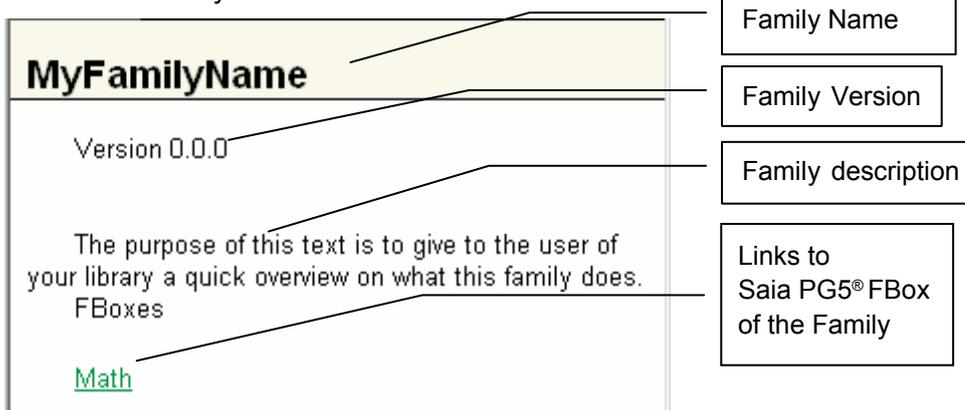
Help Files Pre-defined structure

Each level of the Help has also a pre-defined structure. In the following picture, you will see the look of each level:



3

2. The Family overview:



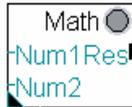
3. The Saia PG5® FBox description:

MathFamily: [MyFamilyName](#)

Name: Math

Macro: _SaiaMath

Version: 1.3.5

**Description**

This FBOX has been developed to show the functionalities of the SAIA FBOX Builder.

The inputs of this FBOX must be two integer numbers.

The output is the result of the addition of the two input numbers.

first	second	third
1.1	1.2	1.3
addition	substraction	
+	-	

Input

Num1 First value for our math operation

Num2 Second value for our math operation

Output

Res Math operation result

3

3.6 Generate Output Files

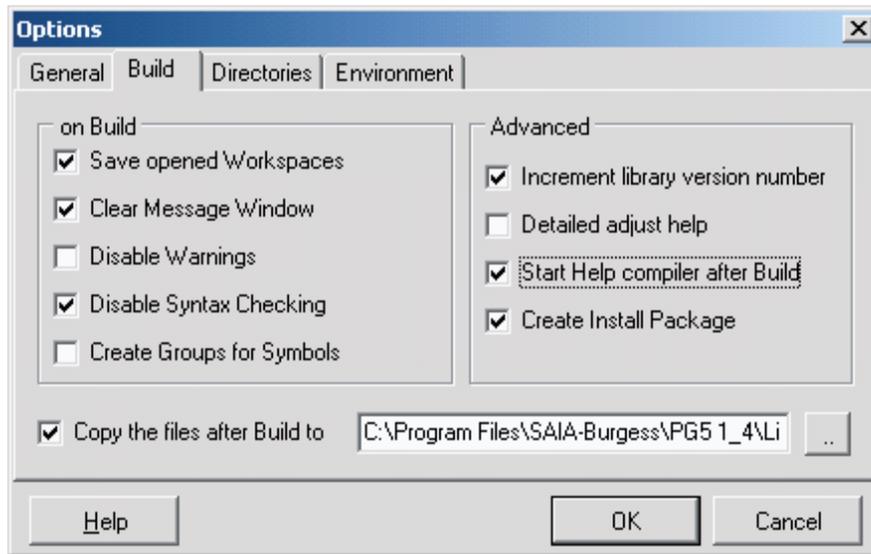
Now everything is ready, the FBox functionality is defined, the help files also. We are able to generate all the files needed by FUPLA for fully run our FBox.

There are two different processes that have to be started, the build of the FBox Library itself and the build of the help file.

You will start the first process by clicking this tool bar button: 

Then you will start the second process by clicking this tool bar button: 

But you can also select the option “Start Help compiler after Build” (View → Option)

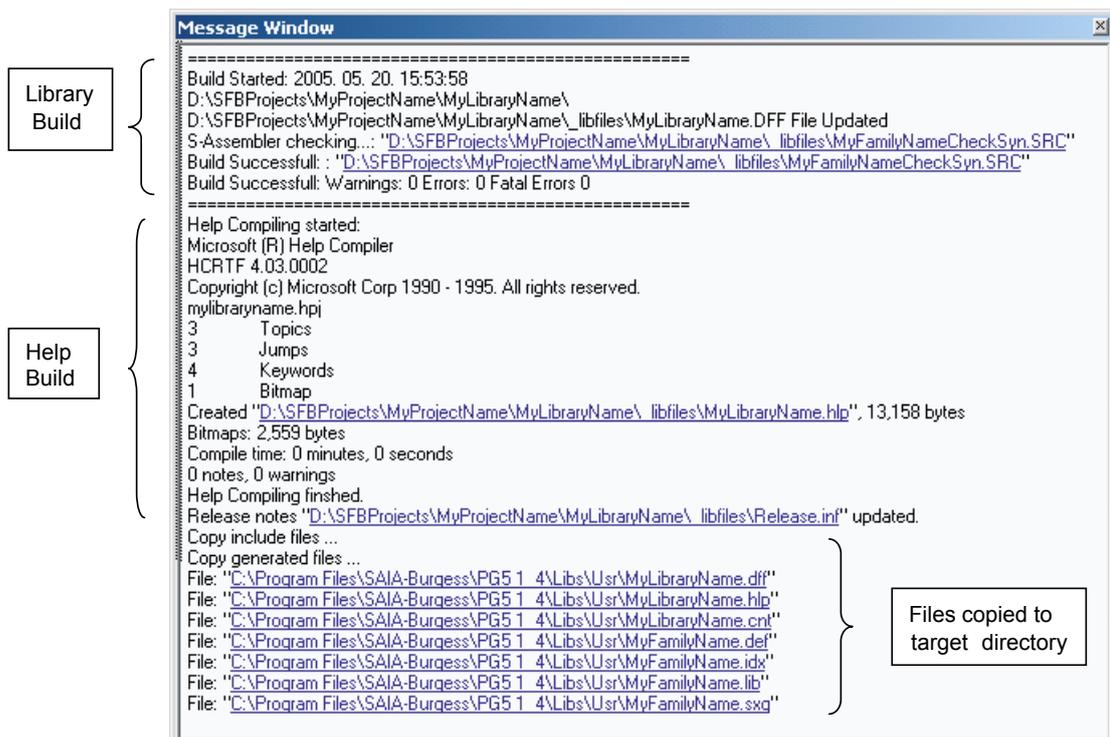


3

Before generating the files, you should select also the “Copy the files after Build to” and select the “Usr” directory of the Saia PG5® library folder. If this option is not selected, you’ll have to copy by hand all the generated files in this directory to be able to use your FBox in FUPLA.

If the option “Disable Syntax Checking” is checked, it means that only the FBox Builder checks the FBox definition; otherwise the Saia PG5® assembler checks each FBox.

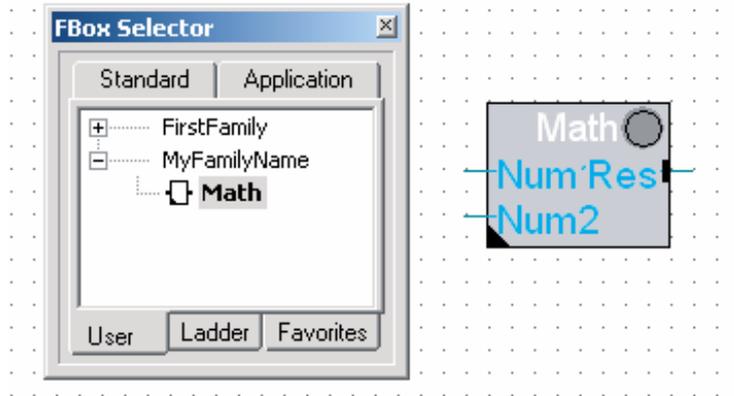
So after the Build process is terminated, you get the description of what happened during this process in the message window. You should get the following messages:





To open a file listed in the Message window just double click on it.

Finally open FUPLA, select the “User” library, the “MyFamilyName” family and the “Math” FBox, you get the following result:

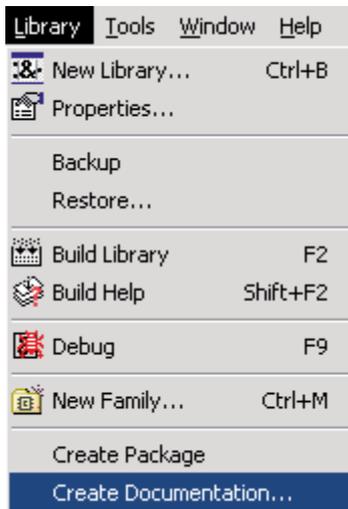


3

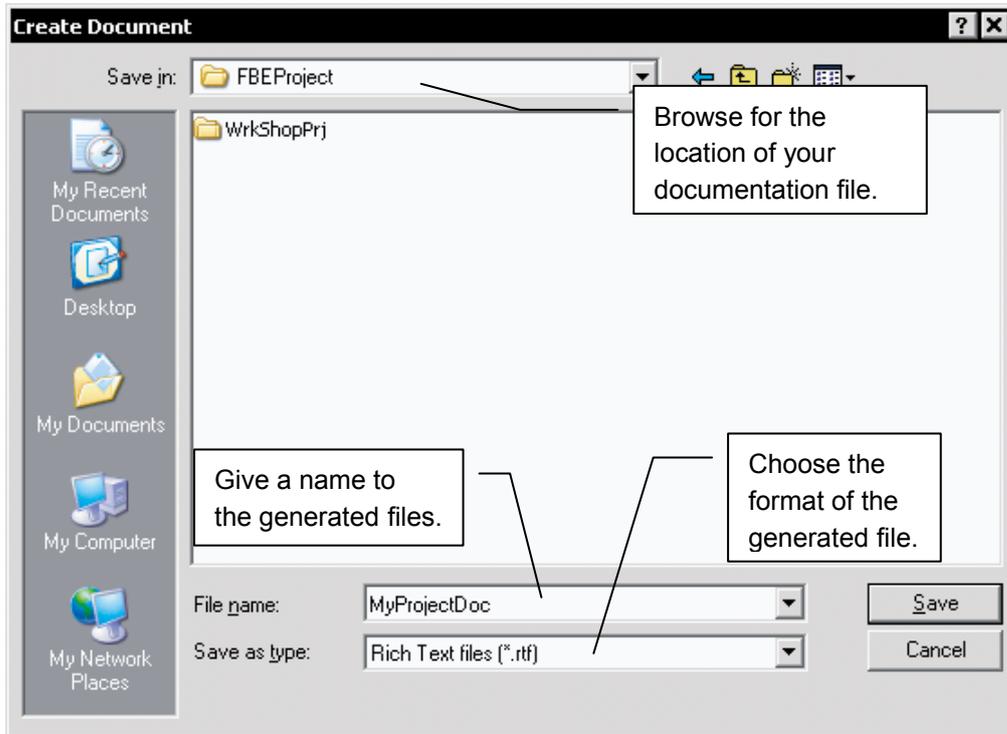
3.7 Documentation Generator

The FBox Builder allows you to generate the documentation that belongs to your project. You have the choice between generating your documentation in RTF or in PDF format.

To generate your documentation, you just have to select the “Create Documentation...” sub-menu in the main “Library” menu.

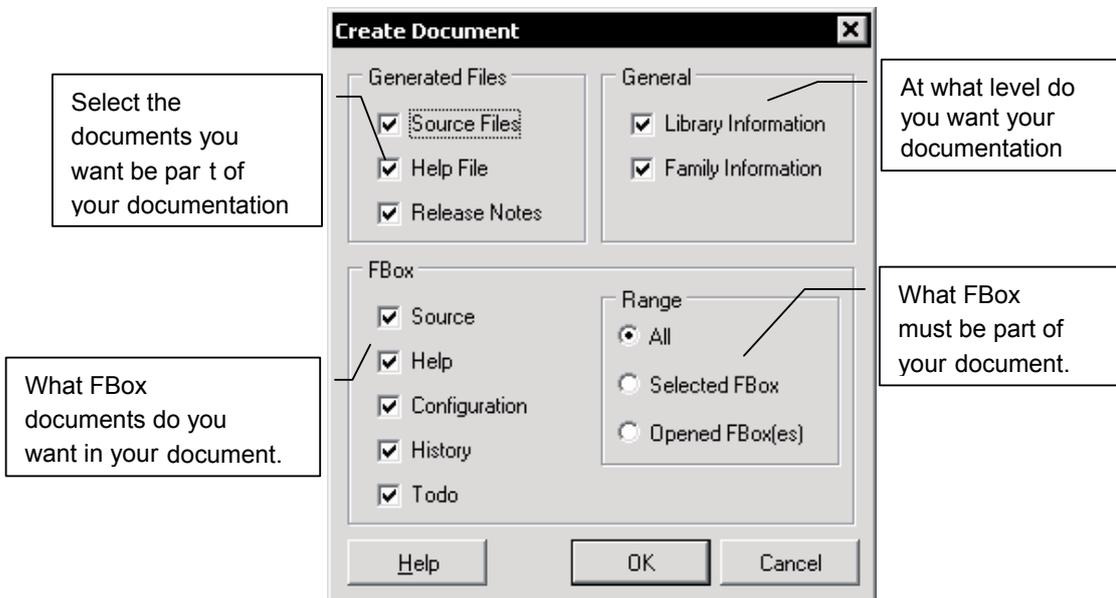


You get the following dialog:



3

After having clicked on the save button, the following dialog popped up:



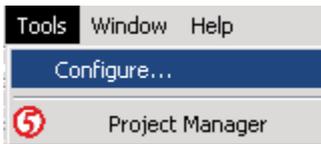
3.8 Add-Tools to Saia PG5® FBox Builder

FBox Builder gives you the opportunity to add your own tool in its environment. For example you can add the Text Editor of your choice and for example configure the extension that belongs to it.

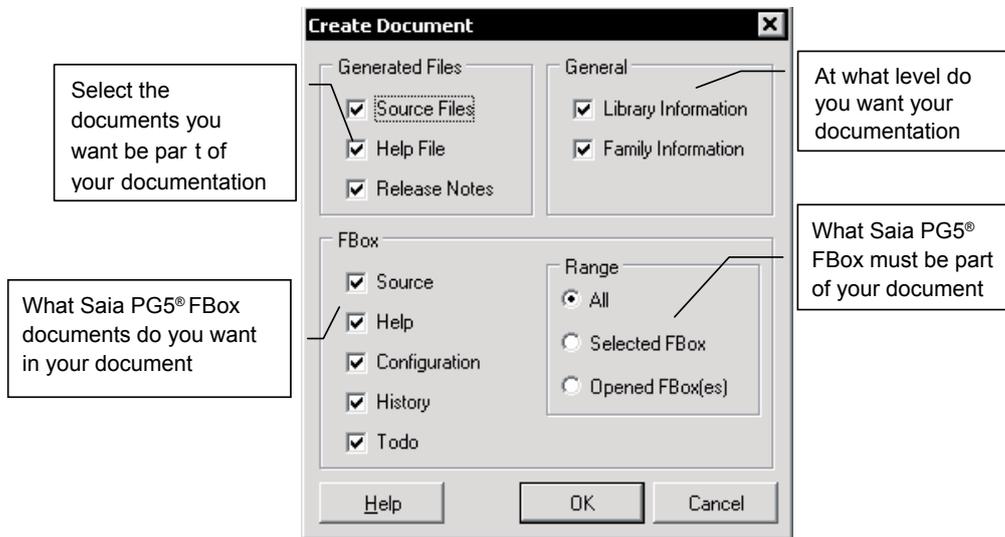
As example, we will add S-Edit to our environment:

1. Call the configuration dialog in clicking on the following menu:

3



You get the following dialog window:



2. Now double click inside the FBox Builder on a "SRC" file, S-Edit will be called to display this file.

To add more than one extension to an add-on tool, you just have to add it more than once with different extensions but with the "Add Tool to the Tools menu" checkbox unchecked.

4 Saia PG5® FBox Advanced

In this chapter you will see the advanced features of the FBox Builder.

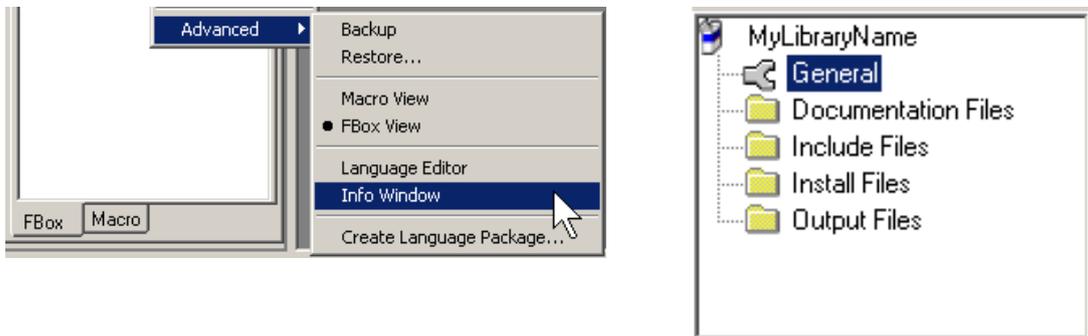
To avoid symbols naming clash with existing FBox libraries, you can contact Saia Burgess Controls to get a list of registered symbols prefixes that you must not use; or if you plan to develop and distribute in a large scale your own library, you can apply at the Saia Burgess Controls to register your own prefix.

Saia Burgess Controls: support@saia-pcd.com

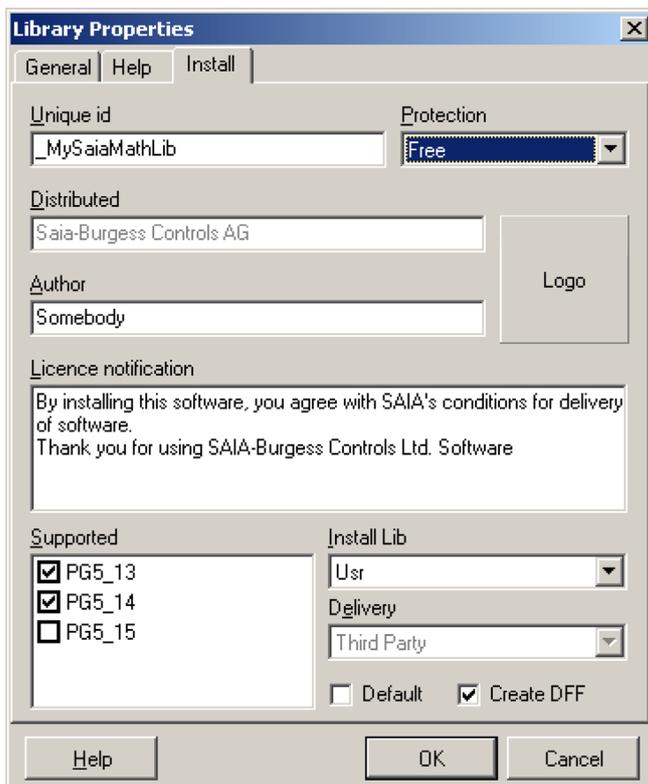
4.1 Library Information

The FBox Builder ease the distribution of FBox Library, the “Library Installer” gives you the chance to make an install package for the user of your libraries.

The Installer program needs some information to customize the installation process. You have to enter this information in the “Library Properties” dialog. To get this window, right click to the “Library manager” for displaying the “Info Window”. Then double click on “General”.

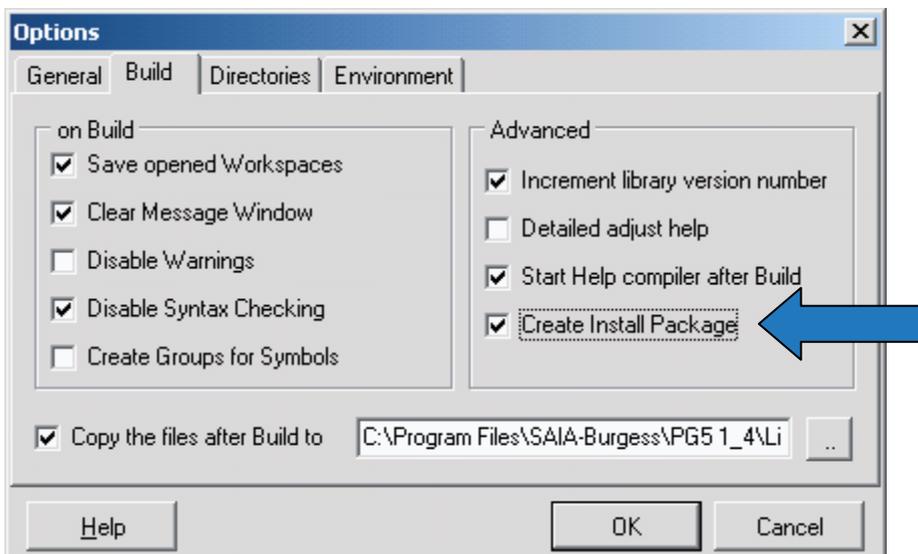
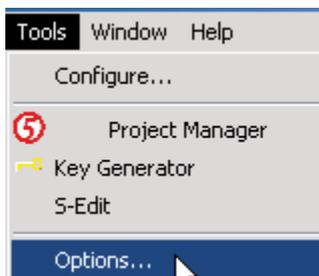


You get the following dialog, and select the “Install” tab: 



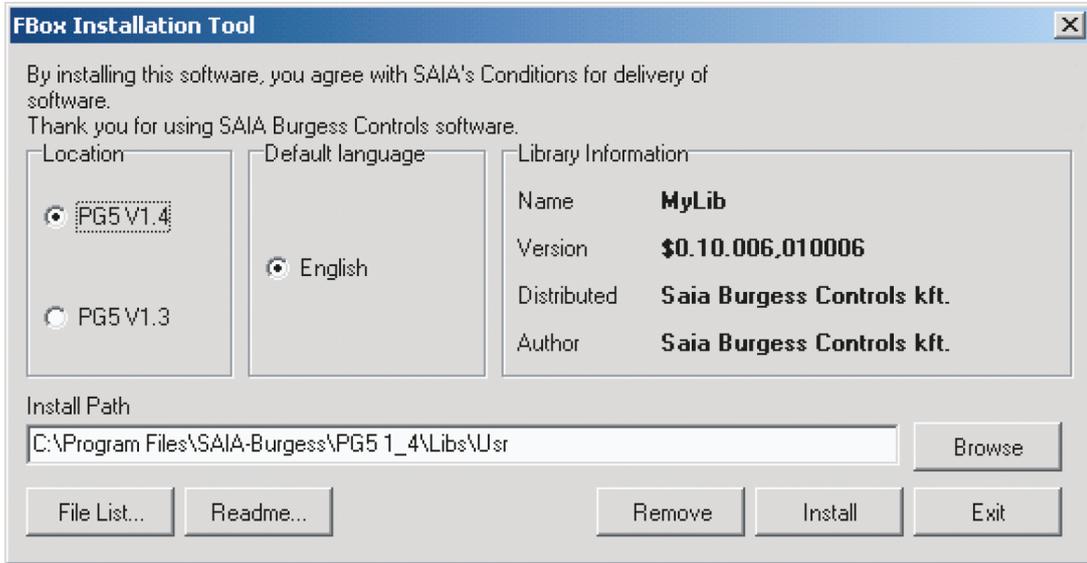
- **Unique id:** Enter a unique identification string, this string is checked in all library files when installing a new library to be sure that you are not going to overwrite an existing library
- **Protection:** Select the kind of protection you want to distribute your library. You can distribute “Encrypted” source file, Licensed library (working with the PG5 license mechanism) or you can distribute your library free
- **Distributed:** Enter the company name that distributes this software, this text will appear during the installation process
- **Author:** Enter the name of the Library developer, this text will appear during the installation process
- **Logo:** Click on the “Logo” button to browse a picture that you want to display during the installation process
- **Licence notification:** Enter a text that should be read during the installation process
- **Supported:** Select the PGX versions that supports your FBox Library. If the Installer does not find a PGX of the selected version, an error message dialog will be popped up; then the user has the choice to cancel the installation or to browse for a directory to install the library anyway
- **Install Lib:** Select in which PG5 directory your library should be installed. (STD, APP or USR; USR is the default)
- **Delivery:** Only for SBC internal use.

Now when you have entered all the needed data, it is the time to create this Install package. Basically you just have to choose the “Create Install Package” option as follows:



This way when you build your FBox Library the Install Package will be automatically created. The package is an executable file, its name is based on the Library Name, the Language and the version. The Following message will be written in the “Message Window”:

If you double click on it, the executable file will be started and you can see the link with the “Library Properties Install” and the Install package:



4

The file list shows simply the files which will be installed and the readme is based on the Library description you wrote in the Library Properties Dialog.

4.1.1 Licensed Saia PG5® FBox Library Install Package

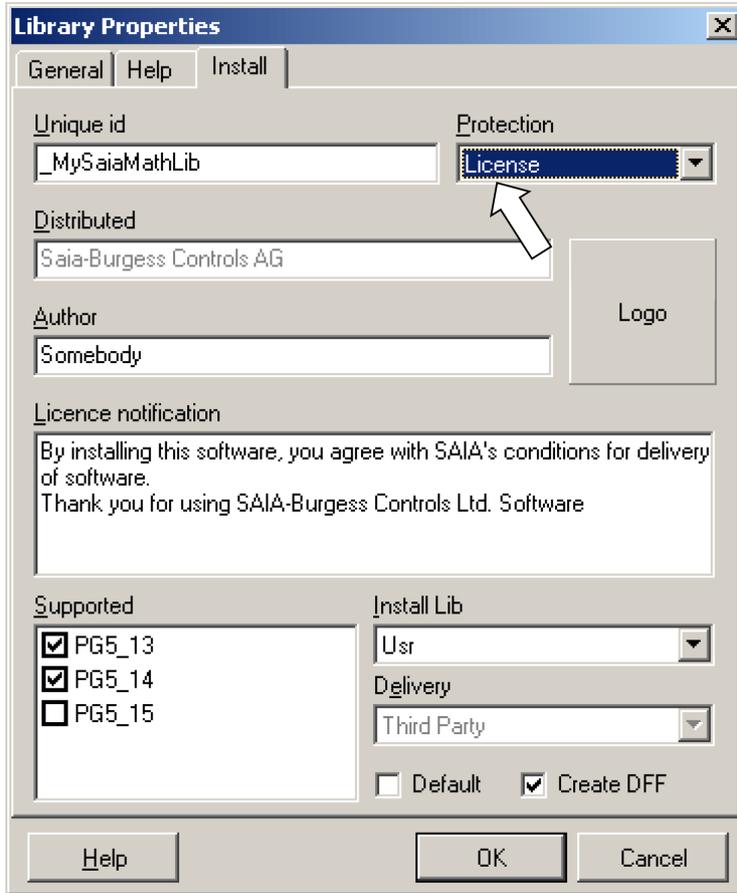
This feature allows you to generate a licensed install package for your own libraries. The FBox Builder generates a license key based out of your customer name and the Unique ID of your library. The install package will then ask this key before installing the library.

If a licensed FBox is used without a license, PG5 won't compile them. This mechanism is implemented since the version PG5 1.3.

1. Set the Protection to License in the Library Properties dialog:



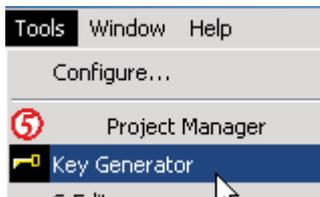
The distributed field is read-only, because it is taken from the PG5 user key and this info is part of the license code generation



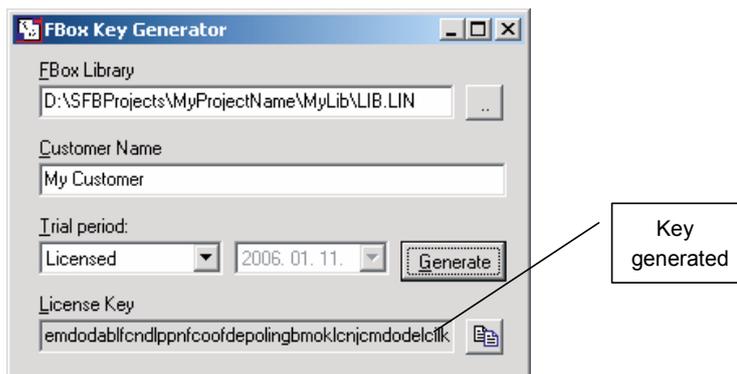
4

Builds the library and generates an Install package.

2. Now you have to generate the key for your customer, to do so click on the “Key Generator” menu of the main tool menu.



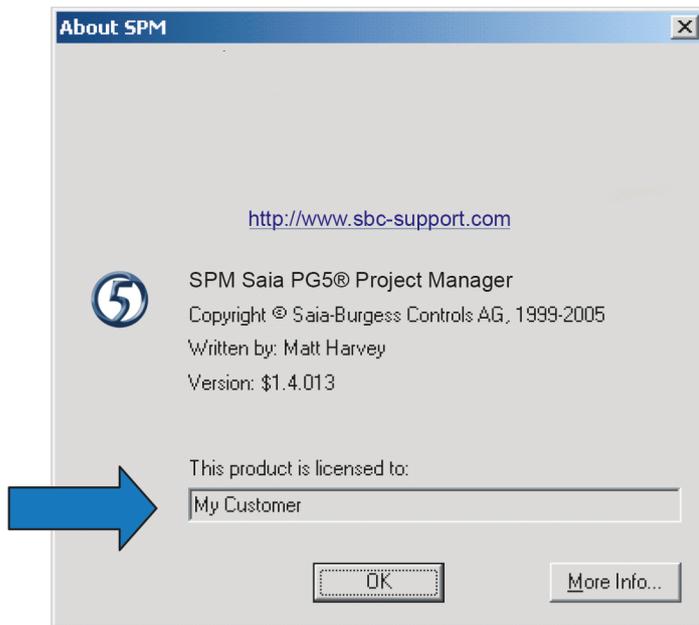
The following dialog comes up:



In the FBox Library field you must enter the path to the LIN file associated to your library. The location of this LIN file is the following:

\$FBOX Builder Project Dir\$\Your Project Dir\$\Your Library Dir_libfiles

The customer name must be the same as the customer’s PG5 key. You can find this name in the PG5 about box:

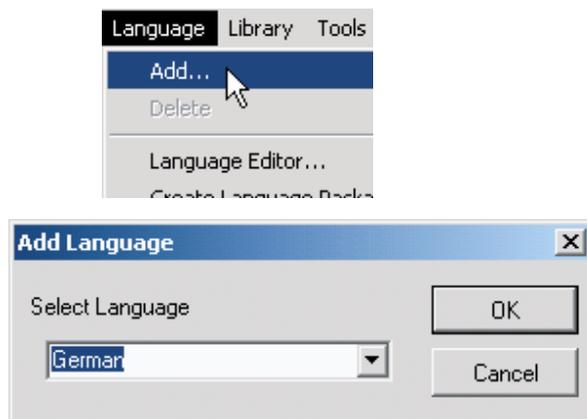


To install your Library, the customer will have to enter the generated key during the installation process. The installer will also enter a key in the computer registry of the customer, which will be used by PG5 to check the right to use of this library.

4.2 Multi-Languages Handling

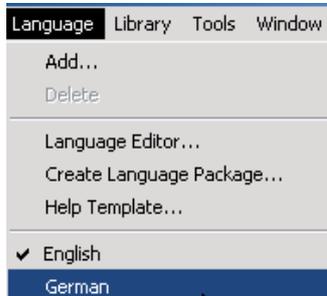
A FBox Library must in some cases support more than one language. The FBox Builder has a “Language Editor” feature that helps you in the handling of the multi-languages library development.

The help file and the FBox strings themselves are language dependent. You can develop you FBox Library in one language and at the end when everything is finished you can add one or more new languages and start the translation. The default language is always English but you can easily add a new one as follows:



Choose whatever language you want.

To change from one language to another, you have to use the “Language” menu as follows:



4

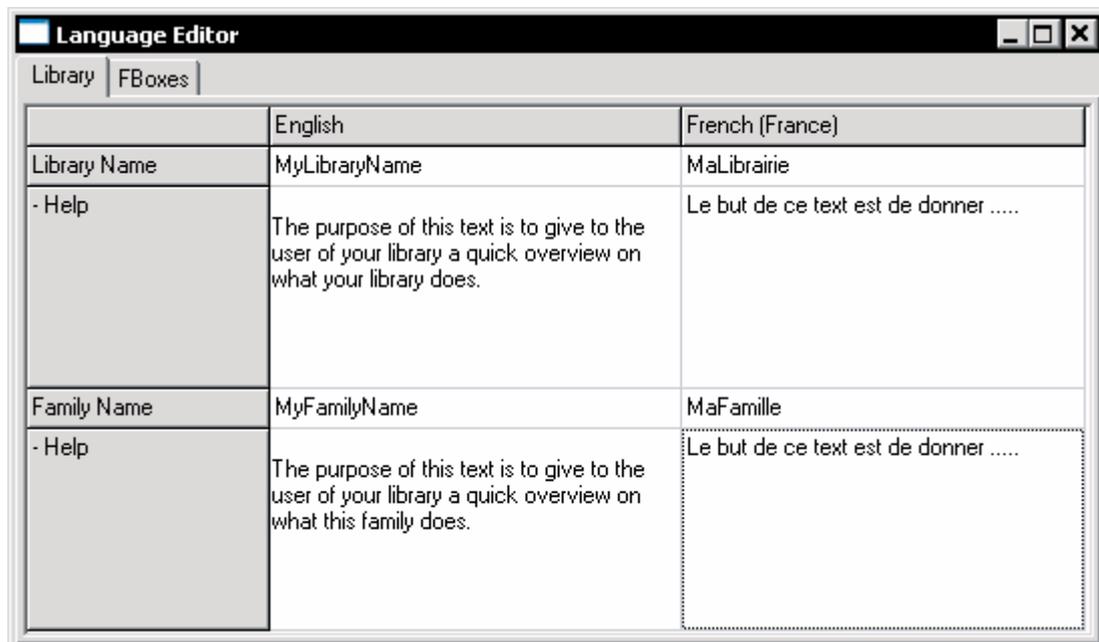
If you want to translate all the strings from one language to the other, there are two possibilities, either to select the language in which should be translated and then to go through each string of each workspace needed to be modified, or the quickest way is to use the “Language Editor”.



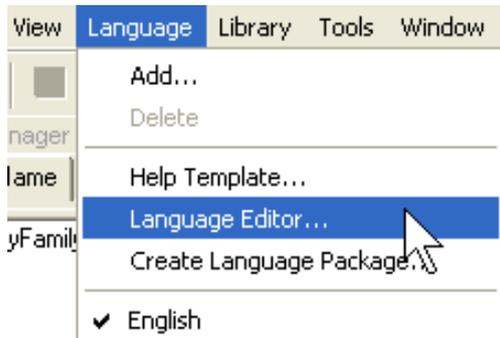
All the languages must be added when creating a project and should never be deleted. The Restore mechanism is not able to restore more languages than the “Actual” project languages.

4.2.1 Language Editor

The “Language Editor” gives you a good overview of all the translation work you have to do. The languages are presented in a table, one beside the other as follows:

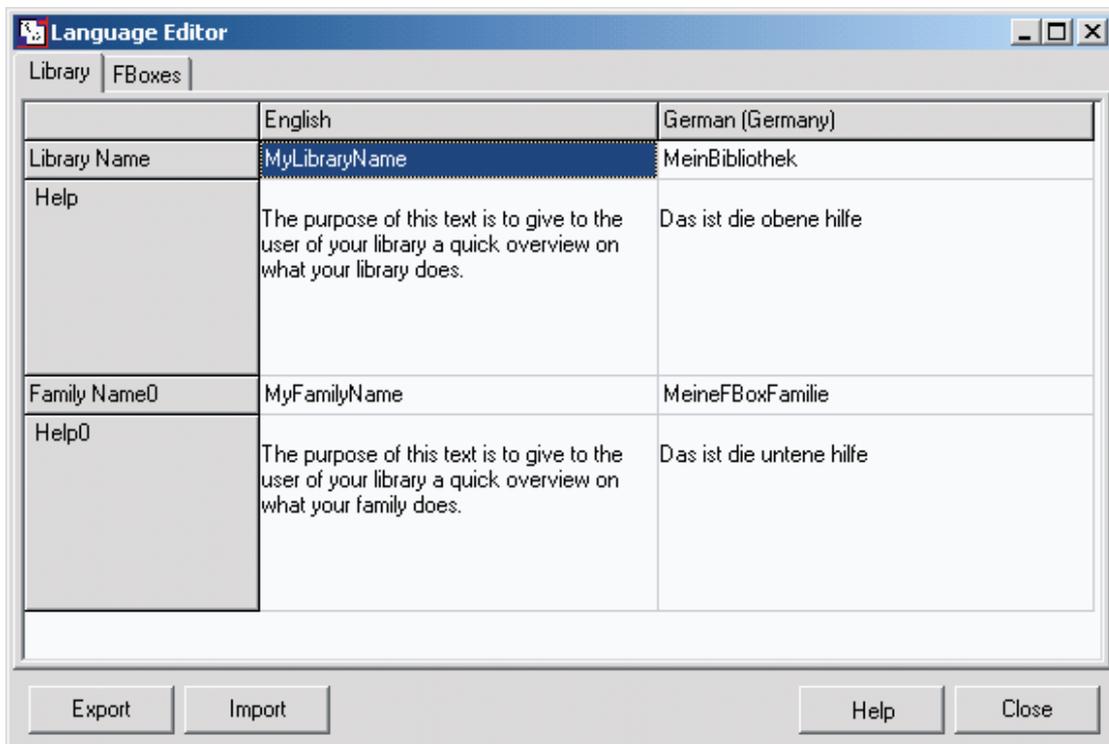


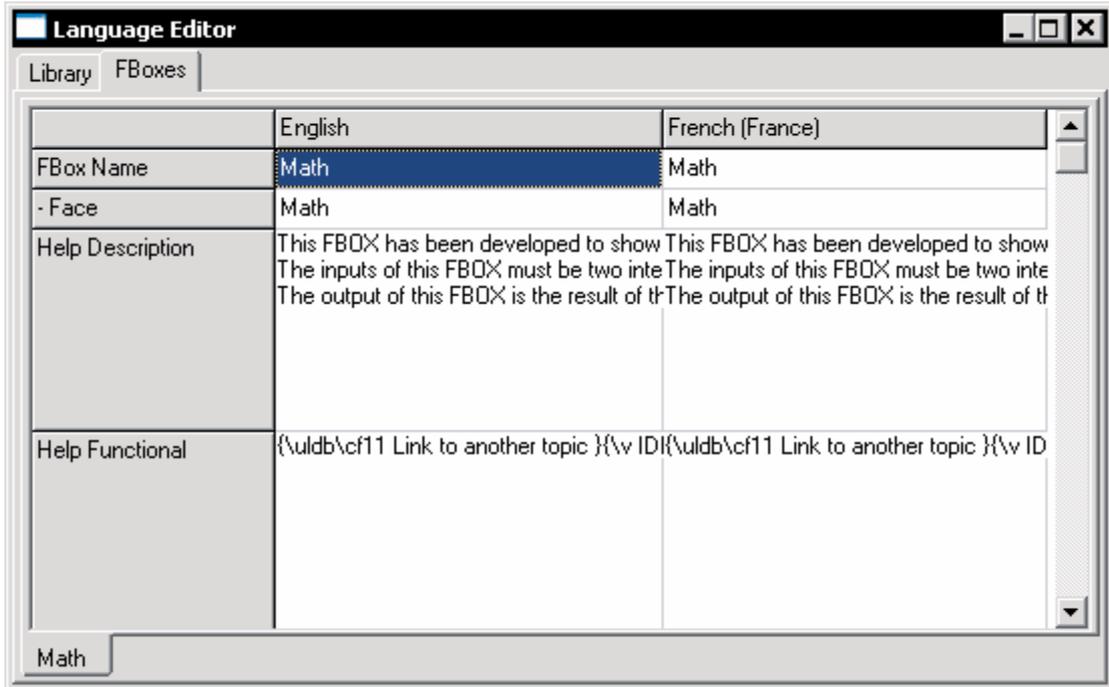
To start this tool, you have to click on the “Language” Menu and select the “Language Editor” as follows:



4

The “Language Editor” covers the three levels, Library, Family and FBox as you can see on the following pictures:





4



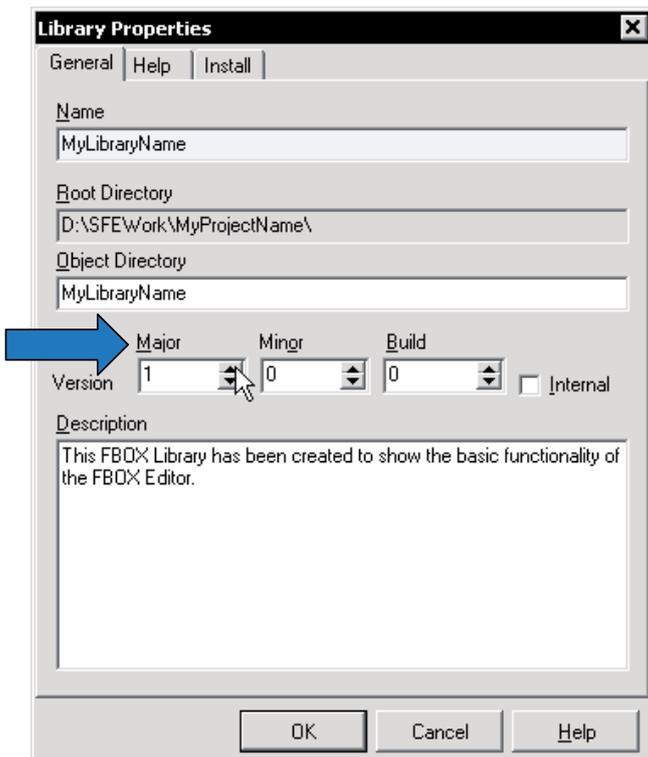
When the translation is finished, you can build your library. You have to do a library build for each language.

4.3 Backup/Restore Mechanism

When you distribute your FBox library, you have to manage different versions. The FBox Builder has a Backup/Restore feature that helps you to handle this versions, not only on the Library level, but also on Family level and single FBox. This mechanism works at every level and it is very simple to use.

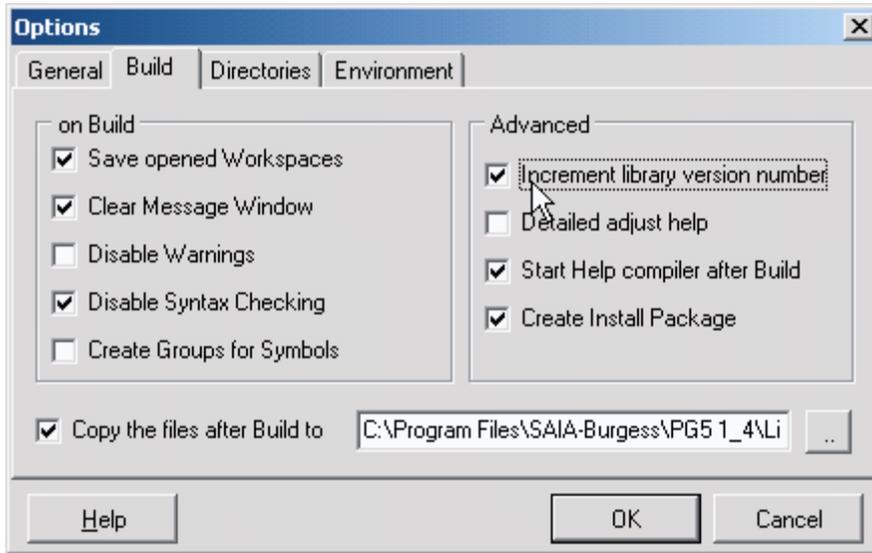


It is important when you do a backup, you must increment the version number. You find this version number for the Library in the “Library Property” window:



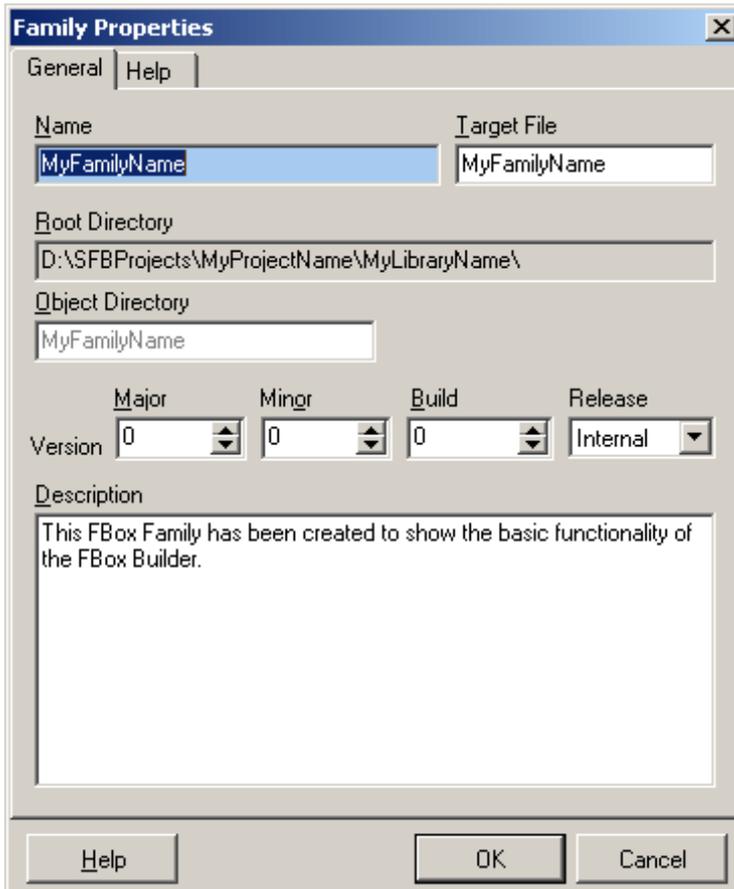


For the library level, you can make the version increment automatically if you choose the “Increment Library version after Build” option:

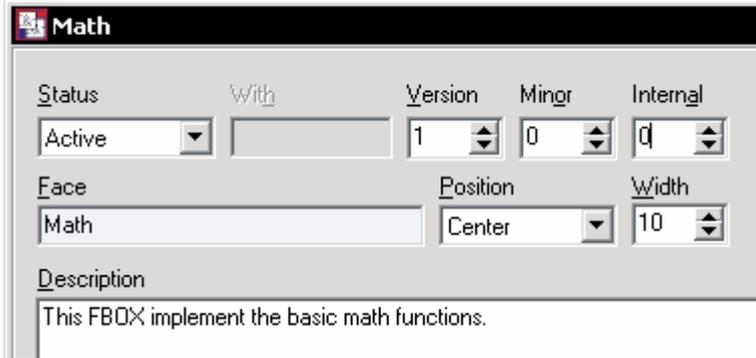


4

For the Family level, you will find the version in the “Family Properties” Dialog:

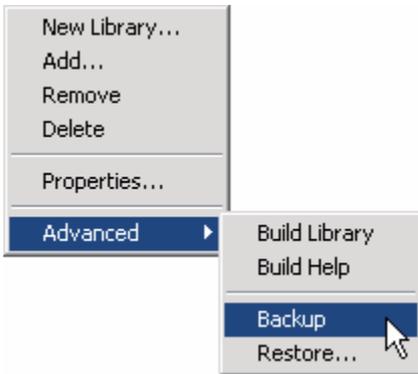


And finally for the FBox Level, you will find it on its workspace:



4

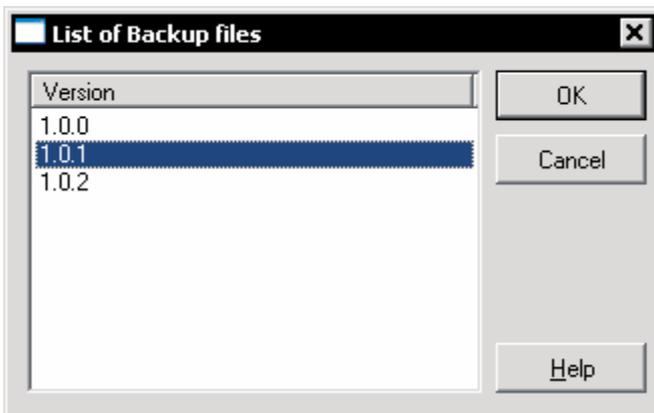
So, to make a backup at any level, you have to click right on the object you want to backup in the “Library Manager” and select the “Backup” menu:



You will get the following message in the “Message Window”:

|| Backup of Library MyLibraryName version 1.0.0 created.

To restore an object that was backuped, the process is exactly the same except that you have the choice which version you want to restore, the FBox Builder asks you with the following dialog which version you want to restore:



Make a backup of the current version before restoring an old one because it will be overwritten!

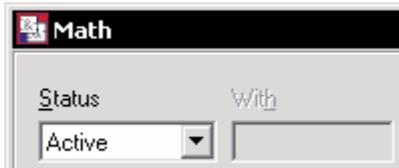


All the languages must be added when creating a project and should never be deleted. The restore mechanism is not able to restore more languages than the “Actual” project languages.

4.4 Library Maintenance Handling

An FBox Library that you distribute is always living, you might have to re-organise your library. The maintenance handling only works at the FBox level. The FBox Builder helps you to handle this kind of problems. If you have a look at the workspace of an FBox, in the “General” tab you will see the following:

4



For each FBox you can choose the following status:

4.4.1 Active

This is the default status of an FBox.

4.4.2 Renamed

Renamed FBox can be used if only the name of the FBox is changed; the interface is identical. The new macro associated to the old FBox can be located in a new family. FUPLA assumes that the new macro is identical to the old one except for its name.



4.4.3 Removed

The FBox developer can decide to remove a FBox from the library. This must be handled with care. FUPLA will cross out all FBox, which were used and are not in the library. This way the user knows immediately that one FBox is not part of the library anymore. Normally when the user will compile files with FBox, which are not part of the library, an error will be shown “FBox not in library”. The user can manually remove this compilation option. FUPLA remember from which family the missing FBox come from, and if some info help is asked for these FBox FUPLA will look in the appropriate family for this info.

4.4.4 Replaced

Replaced FBox are used when an old implementation was changed (the interface of the new macro became incompatible with the old one). The user does not have to modify its code but FUPLA does not allow the user to add the old FBox type. The old FBox still work but the FBox developer would like to force on a long term to use the new one.



4.4.5 Must be replaced

Must be replaced FBox are used when the FBox developer decides that no body should use the old implementation of an FBox. The user must modify its code; replace the old FBox with the new one. When displayed by FUPLA these FBox are crossed-out. When compiled an error warns the user “FBox not in library.” unless the option “error when FBox not in library” is disabled.



4.4.6 Obsolete

This tells FUPLA that this FBox was removed from the library and that it was not replaced by any FBox. The compiler gives no error, but the user cannot any more select this FBox from the FBox library.

4.5 Debugging an Saia PG5® FBox

The functionality of FBoxes created with the “ZIP function” or “Import FBox from FUPLA page” is to be tested and debugged as FUPLA code before importing the FUPLA page(s). The reason therefore is that an FBox created with the function “Import FBox from FUPLA page” does call other FBoxes internally. The source code of this called FBox macros is hidden and cannot be modified in the Basic Version of the FBox Builder. Principally the “Debug FBox (F9)” function in the menu “Library” (or the according button) focuses on the IL code of the FBox, therefore it is not useful for testing your new FBoxes.

The following procedure can be used as guideline for testing the pages that are to be converted to an FBox.

- Create the FUPLA code. It is best if the inputs of this code are not modified by the rest of the application
- Build and then download the test program
- Go online with FUPLA and navigate to the pages to be tested
- Right click on the first FBox of the program to be tested. From the context menu, select “Run to” and then “FBox” The Saia PCD® will be stopped right before executing this FBox
- Use the Watch Window for modifying the inputs of the active FBox
- Set probes to the outputs of the FBox for checking the value
- Execute one FBox with “Step by step” from the menu “Online” or the F11 key
- Step through all pages to be exported and verify the correct functionality of the page(s).



Don't forget testing special cases such as the behaviour of dynamized inputs and at start-up. The start-up of the Saia PCD® can be tested by executing a restart. Therefore use the “Online Debugger” (from the Project Manager menu “Tools” or by pressing the F11 key) and type “RCA” (for Restart Cold All CPUs) before the procedure above.

A Appendix

A.1 Icons

	<p>In manuals, this symbol refers the reader to further information in this manual or other manuals or technical information documents. As a rule there is no direct link to such documents.</p>
	<p>This symbol warns the reader of the risk to components from electrostatic discharges caused by touch. Recommendation: Before coming into contact with electrical components, you should at least touch the Minus of the system (cabinet of PGU connector). It is better to use a grounding wrist strap with its cable permanently attached to the Minus of the system.</p>
	<p>This sign accompanies instructions that must always be followed.</p>
	<p>Explanations beside this sign are valid only for the Saia PCD® Classic series.</p>
	<p>Explanations beside this sign are valid only for the Saia PCD® xx7 series.</p>



A.2 Contact

Saia-Burgess Controls AG

Bahnhofstrasse 18
3280 Murten
Switzerland

Phone +41 26 672 72 72

Fax..... +41 26 672 74 99

Email support: support@saia-pcd.com

Supportsite: www.sbc-support.com

SBC site: www.saia-pcd.com

International Representatives &

SBC Sales Companies: www.saia-pcd.com/contact

Postal address for returns from customers of the Swiss Sales office

Saia-Burgess Controls AG

Service Après-Vente
Bahnhofstrasse 18
3280 Murten
Switzerland

