Technical Document

Sape - User Guide

Document Number	**89_03_17_01968**
Version	**0.9**
Status	**Draft**
Creation Date	**2003-11**
Last Changed	**2005-05-10 by Tobias Vogler**
File Name	**sape_userguide.pdf**

Important Notice

Change History

Date	Changed by	Version	Status
2003-11-12	TVO	0.1	Draft
Initial.			
2004-04-15	TVO	0.2	Draft
Added concept of external linked documentation.			
2004-10-21	TVO	0.4	Draft
Updated according to Sape v0.7.0.			
2004-10-21	TVO	0.5	Draft
Updated according to Sape v1.0.1.			
2004-12-31	TVO	0.6	Draft
Updated according to Sape v1.1.0.			
2005-03-03	TVO	0.7	Draft
Updated according to Sape v1.1.2.			
2005-03-22	TVO	0.8	Draft
Updated according to Sape v1.2.0.			
2005-05-10	TVO	0.9	Draft
Updated according to Sape v1.2.1.			

Table of Contents

List of Figures

Chapter 1. Installation of Sape

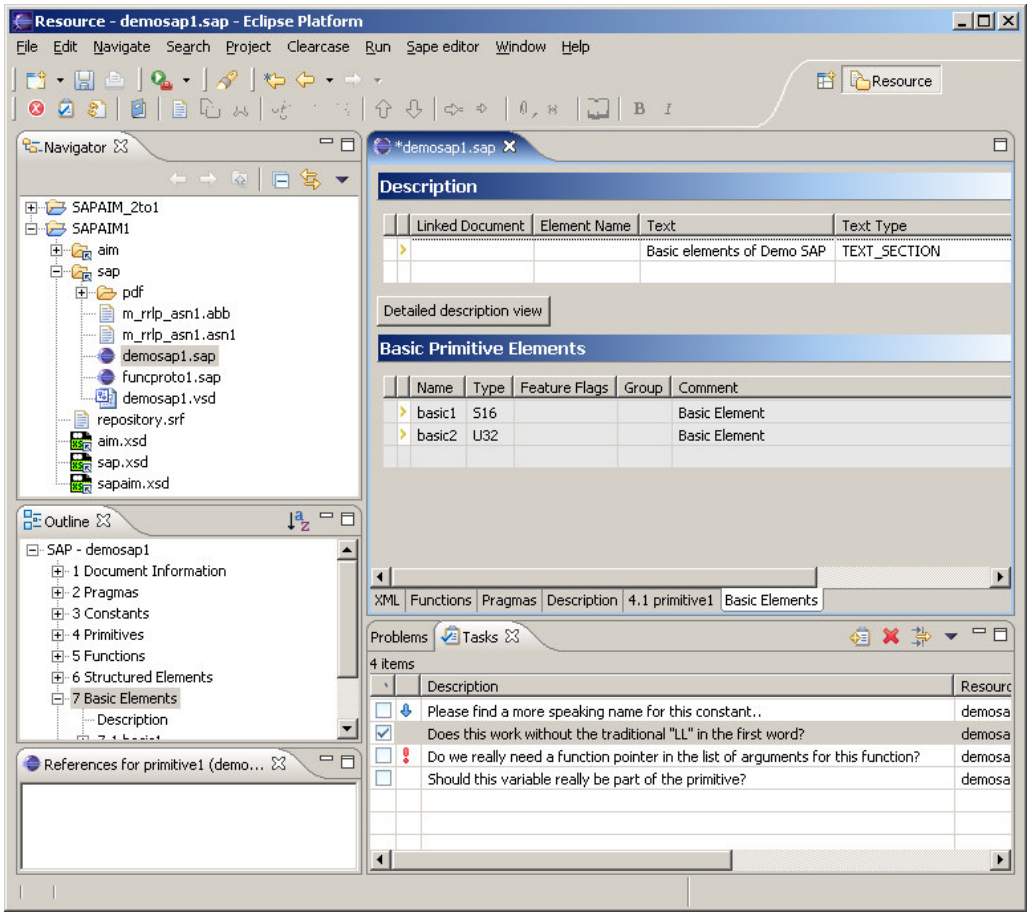
1. Install Sun's java runtime environment, version 1.4.1 or higher if not already installed on your local machine,

2. Start Sape (Figure 1.1) from the gpf/BIN directory by calling "sape.bat" from the command line or by double clicking the icon (the latter may not work in all environments). Please be aware that the full functionality is only available when you start Sape from a 4nt-shell with initvars environment variables set. If the start-up procedure does not work at all, telling you that no java runtime environment has been found, or if it takes the wrong version on your local machine, you can correct this by explicitly giving the path to the java virtual machine via the command line argument **-vm**, as in e.g.

   sape -vm "c:\Program Files\Java\j2re1.4.2\bin\javaw.exe"

   In order to reset the Sape editor, call the batch file with **--reseteclipse** as the first parameter.

Figure 1.1. Sape Editor



3. Read through at least the next two chapters of this manual. Particularly, the second one of them contains essential information for the use of Sape.

Chapter 2. Getting Started

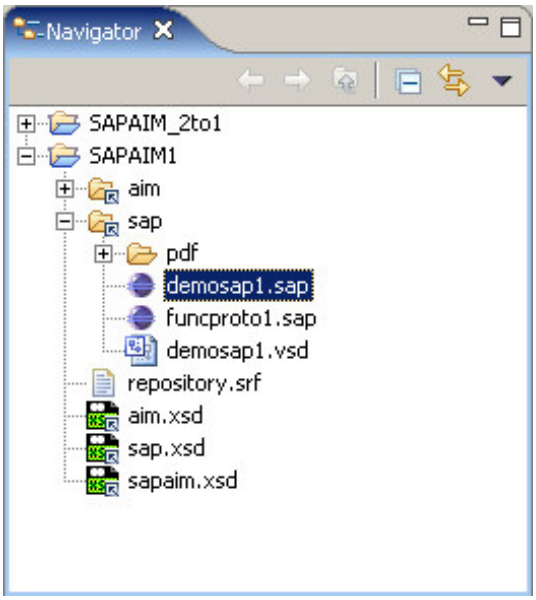
2.1. About

After you successfully got the Eclipse platform to start, you will likely want to do something concerning SAP or AIM documents. To get a feeling for this, you may either create new documents or edit an already existing one.

2.2. Projects

SAP/AIM documents within the Eclipse platform are managed in projects. The open projects can be browsed and modified in the tree-viewer named "Navigator" (Figure 2.1)

Figure 2.1. Navigator Window



which is by default located in the upper left corner of the window. Just after installation, there currently are two standard projects available. These projects are set up so that they fit to the standard paths in which SAP and AIM documents are located in TI ClearCase views. The one, SAPAIM1, is mainly interesting for Berlin based projects like EDGE and GPRS while the other, SAPAIM_2to1 includes both the Berlin paths and the UMTS paths.

The project always works on the view from which Sape has been started. The actual working directories are deduced from the variables GPF_ROOT, TEST_ROOT and UMTS_ROOT (add you own ones via Window->Preferences->Workbench->Linked Resources), so that you can define your own paths within the project based on this information. By default, all projects are automatically synchronized with the file system, with updates occurring in background every few seconds. In order to manually refresh a project's contents, select File->Refresh.

2.3. Directories

If there are exclamation marks on some entries inside the Navigator view from the beginning, this is because the belonging path does not contain the standard SAP/AIM directory paths. For detailed information about the problem, select "Properties" from the affected entry's pop-up menu. If such a path problem cannot be resolved otherwise, you best create your own new folders with paths that are valid for your system. To do this, right click on a project (e.g., SAPAIM1) and select New->Folder. Give it a name, and in the "Advanced" part, check the "Link to folder in the file system" button and

tell the wizard what path you want the new directory to point to. The newly created linked directory and all of its contents are now part of the project.

Now you may open the aim or sap folder (or your newly created folder containing your own SAP/AIM documents - see above) in the project tree. If the file ordering scheme looks chaotic to you, try another one by clicking on the little downwards arrow in the title bar of the Navigator window and selecting, e.g., Sort->by Type.

If now everything is sorted nicely but you cannot see any files with the ending .sap or .aim at all (which are the new XML SAP/AIM files) there probably *are* no such files there in the directory tree you are working with. You could try to generate them from existing word documents as described in the readme_tools help file. Since this is most likely to be a major effort, it is probably better to select some other version of the directory tree with the XML SAP/AIM files already existent. Or, for now, start by creating a new empty SAP or AIM document.

2.4. Files

At last, double click onto the SAP/AIM file you want to open (the ones with the ending *.sap/*.aim) and see the real Sape editor open. Alternatively, if you want to create a new SAP/AIM document, you may go to File->New->Other->SAP Editor->New SAP/AIM document, select next, provide document name and type in the displayed wizard window and finally create the new document by pressing the "Finish" button.

Now, you have to simply use the editor in whatever way you like.

Chapter 3. FAQ

This is a list of Frequently Asked Questions that serves as a short summary of the most essential facts that have to be kept in mind when working with Sape.

- When a SAP or AIM document is being modified, the information in the references window is not updated automatically. So, when it seems like what you see there is no longer valid and you would like it to be refreshed, click the **Rebuild repository database** entry in the **Sape editor** pull-down menu. Do the same whenever something seems to be inconsistent in the internal repository database.
- When you want to reset the standard project ("SAPAIM1") and Eclipse settings to the default settings, delete the view local directory `gpf/tools/bin/eclipse/workspace` and its contents by hand or call **sape.bat** with **--reseteclipse** as the first parameter. This may be needed when e.g. the start-up script did not work correctly and so messed up said directory.
- If you do not know what kind of content you are supposed to put into some field in a table inside the editor, please have a look at the *gtc_userguide*. In that document, all the tables of Sape are described in detail. It can be found at `gpf/DOC/gtc_userguide.doc` .
- If you want to find out more about Eclipse basic functionality, have a look at the Eclipse online documentation in the editor at Help->Help Contents.

Chapter 4. Manual for the SAP/AIM Editor

This manual actually is something of a "walk-through", with the most important highlights of the SAP Editor pointed out and some practical hints for everyday work in between. If you are interested in a specific topic, please refer to the table of contents. If you need to know the details about the meaning of a specific field in the Sape editor, please have a look into the gtc_userguide. If you need information about the command line tools that accompany the Sape editor, please consult the document "readme_tools.txt". And, finally, for questions about the functionality of Eclipse (which is the framework around the Sape editor) you may use the online help functionality (Help->Help Contents->Workbench User Guide).

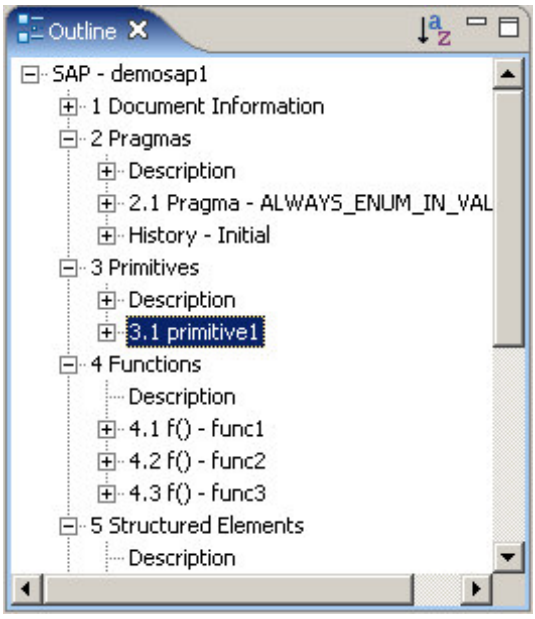
4.1. A new Document

Lets assume that we want to create a new SAP document. The way to a completely new and empty document leads - as already described - via the File->New dialog. If you get some message about invalid XML code here, just tell the program to try and fix it automatically. This most likely means that the internally created XML data is not in the same order as is described in the actually used grammar documents.

4.2. Sape Windows

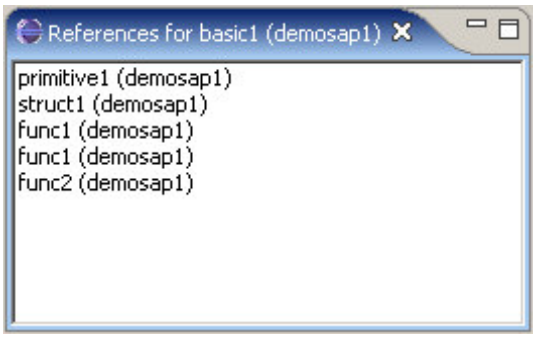
Now that the new document is created, let' s have a look at the other parts of the Eclipse window. On the left, there are two sub windows, "Outline" (Figure 4.1)

Figure 4.1. Outline Window



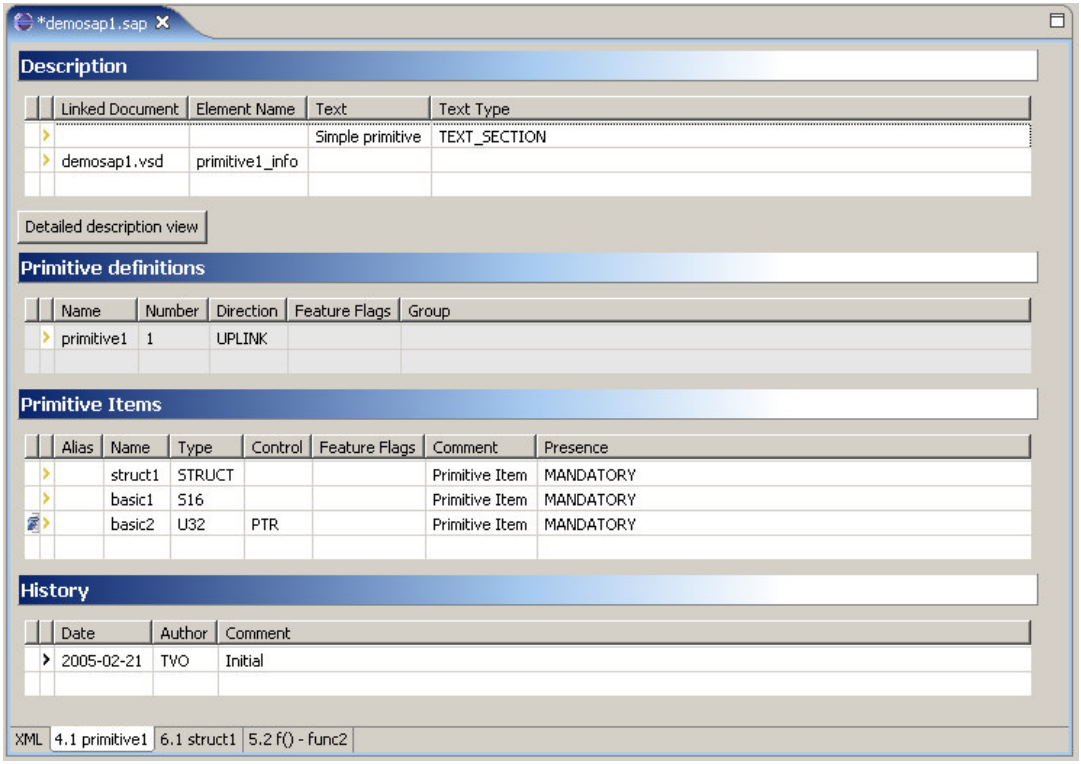
and "References" (Figure 4.2).

Figure 4.2. References Window



The former displays an Outline of the XML document which is actually selected in the Navigator, with all the elements contained in the document (e.g. primitives, messages, values, constants). You may toggle between the real order of the displayed elements and an alphabetically sorted version by pressing the "az"-button in the top right of the window. In the "References" window, the places in which the actually selected element (in the "Outline" window) are used are being displayed. By selecting an element in one of the windows, the big window in the upper right will display additional information about the element.

Figure 4.3. Editor Window

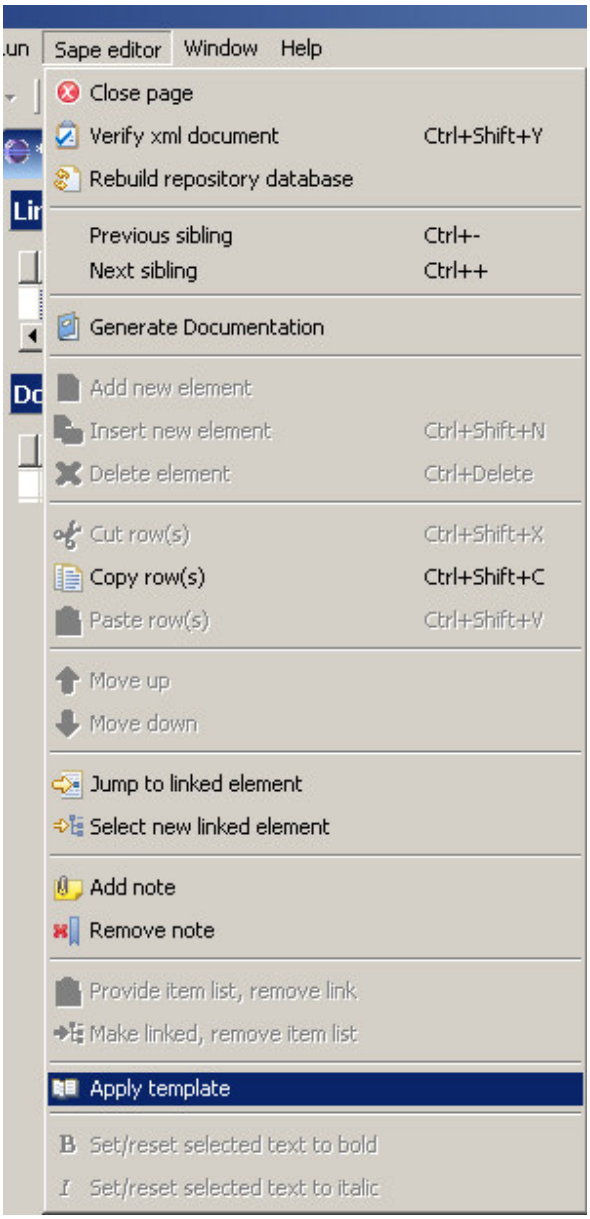


This window (Figure 4.3), the "Editor" window is the place where the actual editing of the document, namely creation and modification of the different types of elements, takes place.

4.3. Editor Commands

Generally, most options concerning the functionality of the Sape editor can be accessed in three standard ways. For one, there is a pull-down menu entry for Sape with the name "Sape editor" (Figure 4.4).

Figure 4.4. Pull-Down Menu

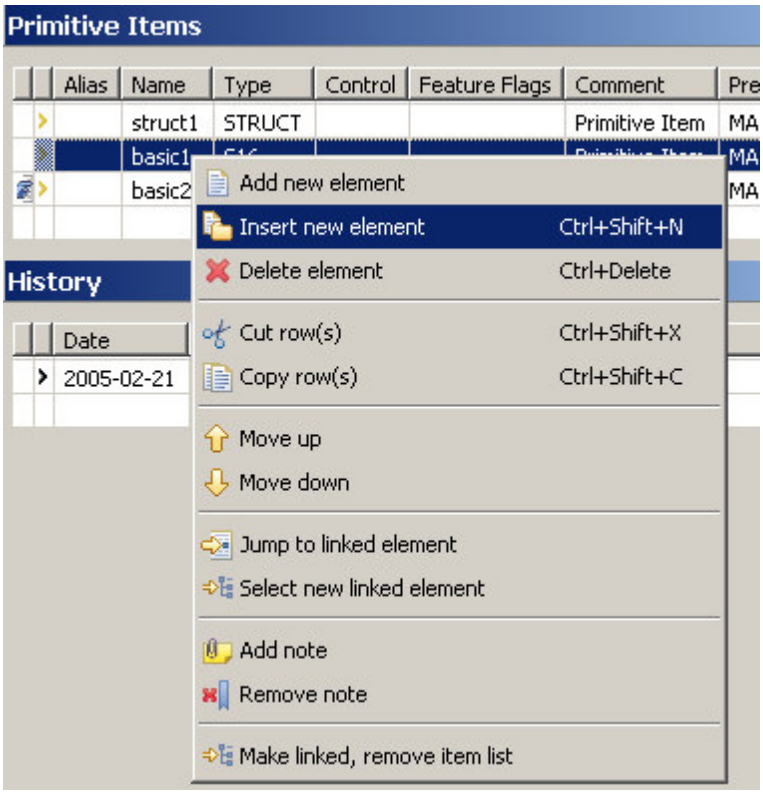


Then, there is the toolbar (Figure 4.5). When you stay the mouse cursor on an item of the toolbar, a short hint will be displayed explaining its specific meaning.

Figure 4.5. Toolbar

Third, most actions can be executed via pop-up menues (Figure 4.6). In order to access these, right click on the item you want to work with (e.g. a row in a table).

Figure 4.6. Pop-Up Menu



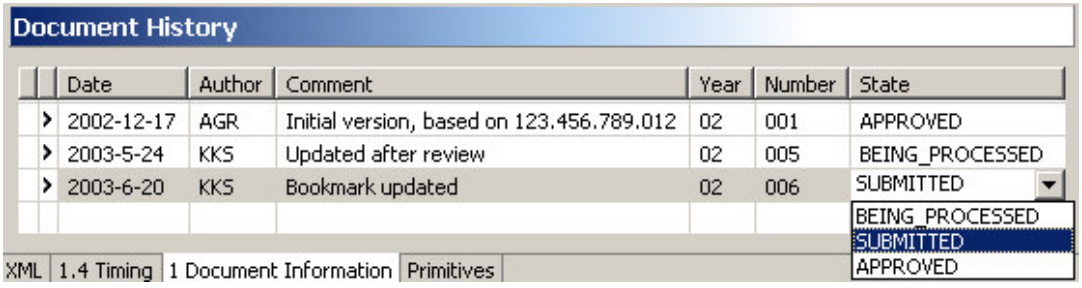
4.4. Editing Data

Initially, there are only the entries "Document information" and "SAP -..." available in the outline. When you select the former entry, information about the overall document will be displayed in the editor. Now would be the right time to fill in some meaningful contents into the "Description" (easiest for now is to just fill/add the "Text" columns, one section per column), "References" and "History" fields respective tables. If you need sub chapters for the description of the document contents, add as many of them to the "Sub Chapters" table as you like, recursively. You may also change the name of the document if it was not supposed to be named as the actual file is.

4.4.1. Modifying Table Fields

To write into some field, just select it by clicking into it once with the left mouse button and begin writing after the cursor appears in it. There are, however, some special fields.

Figure 4.7. Combo Boxes

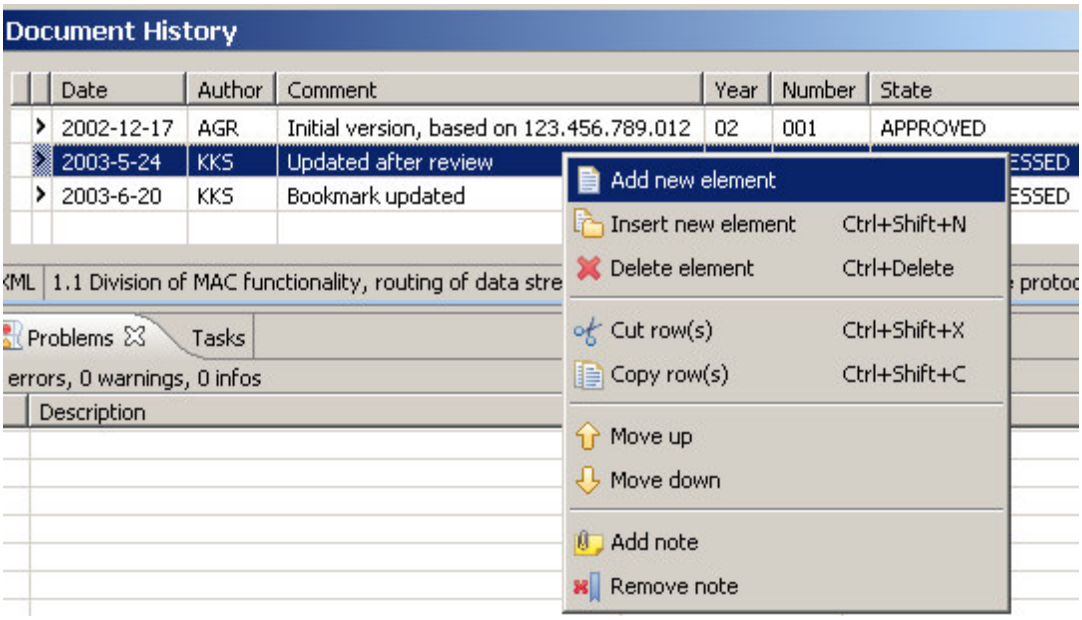


Those which present you with a little arrow in the right corner after you selected them contain a list from which you can select entries after having clicked on the arrow-field. An example is the "State" field in the lower right (Document history, Figure 4.7. See also Figure 4.13). Those which are read only simply ignore all your efforts at clicking on them and remain as they are. You can distinguish

tables which are read only from those whose contents is editable by their light grey background color. Most of the read only tables hide some more detailed editable tables which are reachable by double clicking on the table row you want to edit.

4.4.2. Adding and Removing Elements

Figure 4.8. Adding An Element

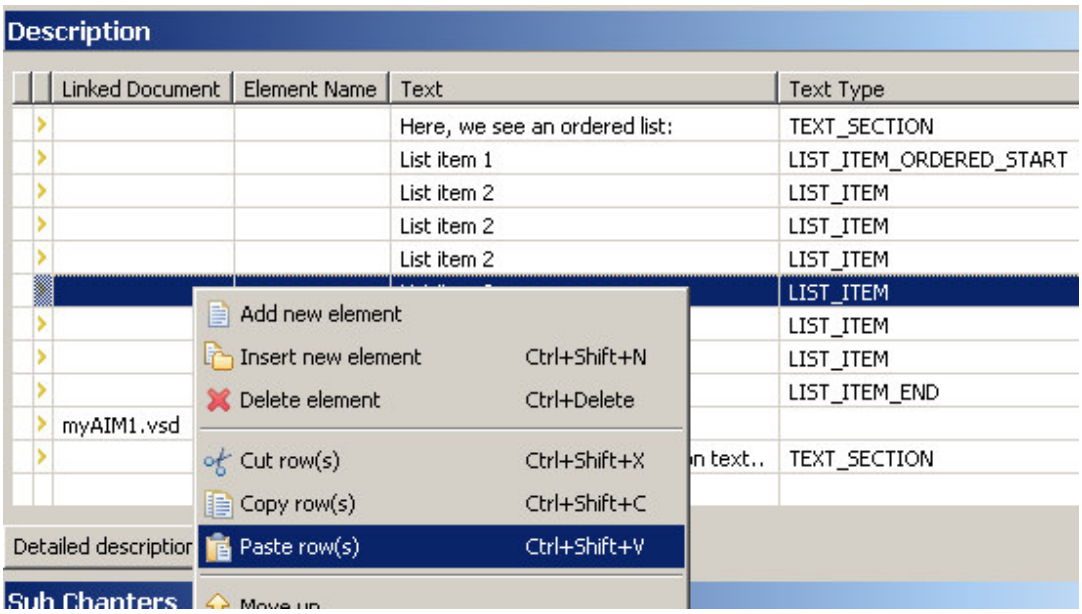


If you want to, for example, add an entry to the document history, click on the table with the right mouse button and select "Insert New Element" or "Add New Element" (Figure 4.8), depending on if you want to add an entry at the end of the list or just before the actually marked table element. You can find out about whether you may add/remove entries in a table by looking on the little black symbols in the second column of each table. A little arrow there means that you may add as many entries to the table as you wish, a filled rectangle means that there must be exactly one entry contained and a non filled one stands for zero or one entries. Addition and removal of whole lines is also possible in most read only tables. To edit such new entries, you must then jump to the created elements themselves. In the pop-up menu of a table, only the options applicable in the actual situation are presented.

4.4.3. Copy/Paste

Regular Cut-Copy-Paste operations on text fields work as is common for most editors. First, the portion of text in question must be marked as selected. It then can be cut or copied by making use of the pop-up menu or the usual keyboard shortcuts <CTRL>+X and <CTRL>+C. Finally, by pressing <CTRL>+V, it can be pasted somewhere else as often as is desired.

Figure 4.9. Copy/Paste of Structures



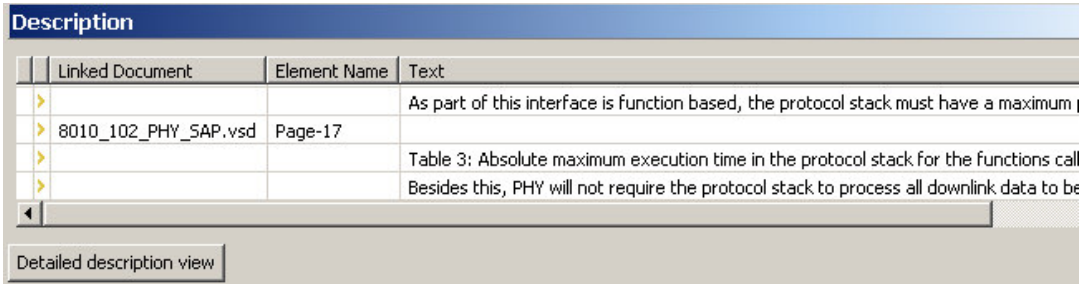
It is also possible to copy entire data structures. The implementation of this functionality is based upon table rows. It works similar to the already explained text copying. Here, one or more rows in a table have to be selected, then cut (they will not directly disappear by marking them thus) or copied by a keyboard or pop-up menu command and finally pasted somewhere else (or into the same table). Again, the paste operation may be executed several times on the same copied elements (Figure 4.9). It is, however, only possible to paste elements into tables which hold the same type of data as the element to be inserted. The default setting for keyboard commands here are <CTRL>+<SHIFT>+X for "cut row(s)", <CTRL>+<SHIFT>+C for "copy row(s)" and <CTRL>+<SHIFT>+V for "paste row(s)". The latter names can be found for the respective operations in the pop-up menu. Here, only the operations which are allowed on an actual table are presented.

4.5. Documentation

All documentation happens within the "Description" sections in the editor. There exists one such section for each part of the document. In addition to this, every part of the introduction and appendices contains one of these sections. In principle, documentation consists of text sections which are held inside the XML document and of external files, e.g. pictures or Visio drawings, that are being linked to the description. In the following, we will first introduce the "Description" user interface to then discuss the basic text features and finally approach the more complex feature of adding external documentation.

4.5.1. Basic Interface

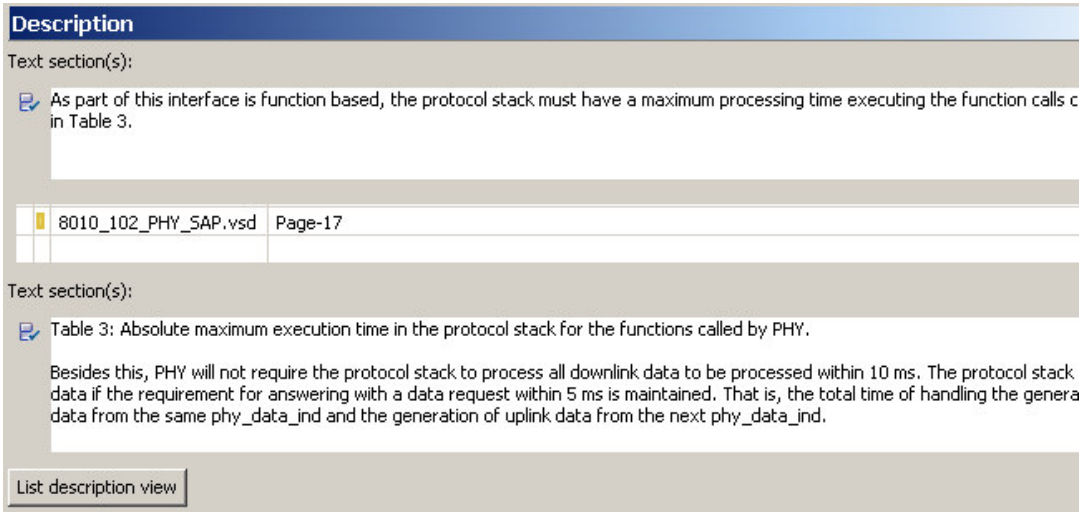
Figure 4.10. Basic Description Field - Tabular



The basic description field (Figure 4.10) is a table in which each row stands for either a section of text or a link to an external document. You can add and edit entries here like in any other table.

There are several possibilities to fill such a table row with contents. The easiest one is to just write down the desired text in the "Text" column *or* the name of a document you want to provide a link to into the "Linked document" column. By double clicking on a row, you can jump to a more detailed page for editing its contents. If you like to have a display that looks more like the resulting documentation, press the "Detailed description view" button at the bottom of the table.

Figure 4.11. Basic Description Field - Detailed

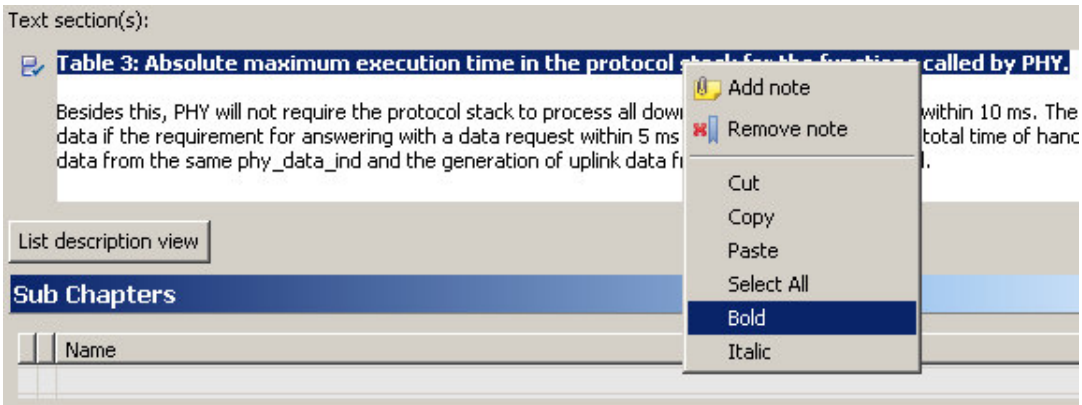


Then you will be provided with a sequence of interspersed text fields and tables for single links which you can edit freely (Figure 4.11). In order to go back to the tabular version, just press the "List Description View" button. Adding, moving and removing of the different description elements is only possible in this view so the general layout should be done from here.

4.5.2. Text

Editing of existing chunks of text is best done from within the detailed view. There you can freely type in text into the displayed text boxes. Sections of text or list items are being separated by line breaks.

Figure 4.12. Bold Text



In order to apply bold or italics style just mark the text in question and press the "Bold" or "Italic" button in the toolbar or pop-up menu (Figure 4.12). In order to reset the style, press the button again. When copying styled text inside the view or between different views of this kind, its style is being preserved.

It is alternatively possible to edit text from within the list description view. There, text has to be typed into the "Text" column. Each section or list item is being represented by one table row. In the

"Text Type" column, it can be determined in which way (regular text section or list item) the affected text should appear.

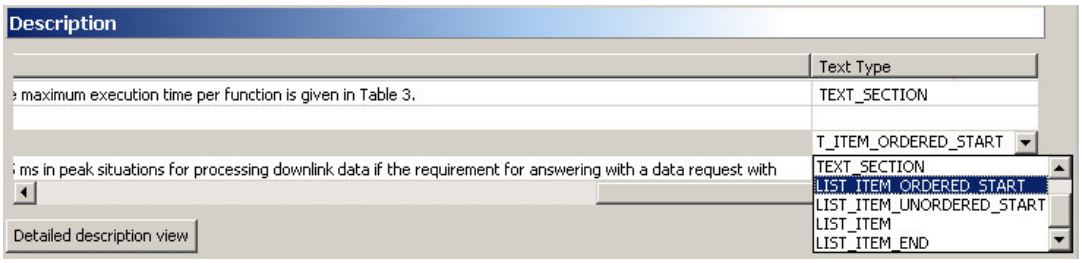
In the list description view, bold and italics styles are represented in regular XHTML syntax. If you more like to generate bold or italic styled text directly from here, add XHTML and tags in the text. Application of this feature is straight forward. Just enter and tags around sections that should be printed bold and and tags around sections that should be in italic format. It is possible to combine both options as long as correct nesting according to XHTML standards is applied. For further information, please have a look at the following two examples. The first of them is perfectly valid, the second one is not valid.

This is some text, with bold formatting, italic formatting and formatting, that is both bold and italic. And now, here comes nesting at its best!

Here, the nestingis not ok!

4.5.3. Lists

Figure 4.13. Type of Text Section



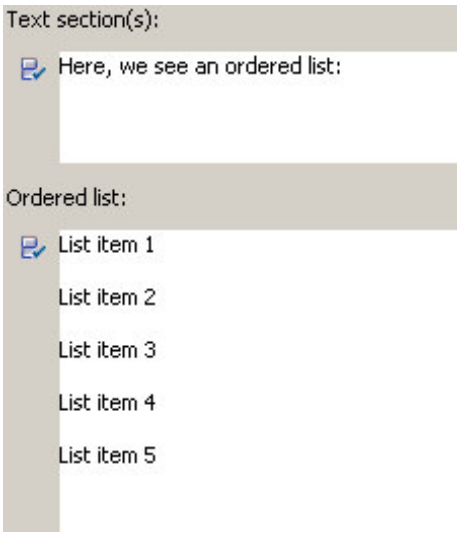
Lists are defined by explicitely dedicating text sections as list items. This is being done in the "Text Type" column in the list description view (Figure 4.13). There you can specify the desired kind of representation for the related text section. The type "TEXT_SECTION" specifies a regular paragraph, while the "LIST_ITEM*" types cause the documentation generation tool to display the section in question as a list item. It is so possible to specify single buttons by just selecting the "LIST_ITEM" type.

Figure 4.14. List - Tabular

Linked Document	Element Name	Text	Text Type
		Here, we see an ordered list:	TEXT_SECTION
		List item 1	LIST_ITEM_ORDERED_START
		List item 2	LIST_ITEM
		List item 3	LIST_ITEM
		List item 4	LIST_ITEM
		List item 5	LIST_ITEM_END

For lists consisting of several items, the first list item must be of type "LIST_ITEM_ORDERED_START" or "LIST_ITEM_UNORDERED_START", followed by as many regular "LIST_ITEM" entries as are needed. The last element of any such list should be of type "LIST_ITEM_END" (Figure 4.14). In this way it is possible to specify ordered and unordered lists for the generated documentation.

Figure 4.15. List - Detailed

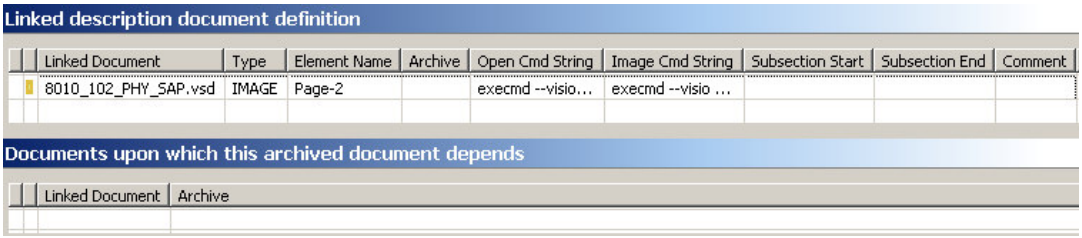


In the detailed description view it is not possible to manipulate the type of text sections. You can, however, see which type a group of sections is by the plain text headline directly above the text field (Figure 4.15). Adding or removing paragraphs in the respective text field will directly affect the underlying list and add respective remove items there as needed.

4.5.4. Linked Documents

Content that is more complex than simple text or lists can be added through links to external documentation files. Such links can be added and modified in the list description view. In order to create a link here, just write down the name of the document to which the link should point in the "Linked Document" column. If such an external document is not located in the same directory as the SAP/AIM document, please provide the relative path with the name of the document using "/" as separators.

Figure 4.16. External Description Link



Upon double clicking on the respective row in the table, a detailed page with information about the underlying link is being opened (Figure 4.15). All further explanations here refer to this detailed page. If you double click on the entry in the "Linked Description Document Definition" table here, Sape tries to automatically open the linked document in an appropriate editor. If this document does not yet exist, a new empty document will be created.

Without further specification, this works for simple images, HTML and ASCII text documents. Depending on the "Type" column, the contents is being interpreted as an image (type "IMAGE"), HTML document (type "HTML"), plain text document (type "TXT") or as a generic link (type "OTHER") in the generated documentation and will be displayed respectively. Best supported image formats are EPS (which is best for printing) and BMP. Some other standard formats are understood, too. A special case here are EMF/WMF-format images which can be used as long as the Visio application is installed on the computer used for documentation generation.

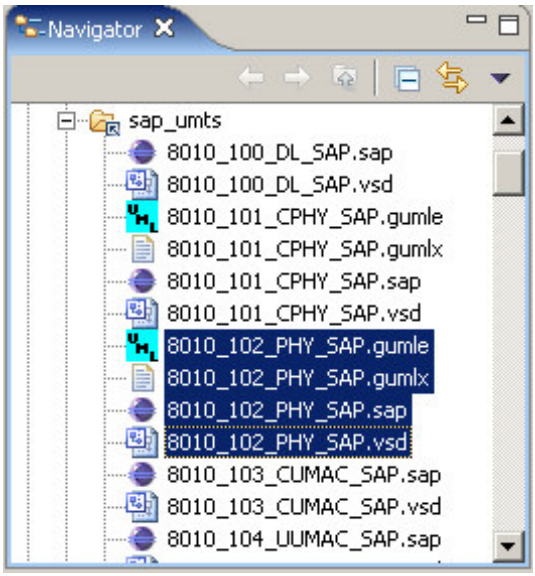
However, using only such basic external documentation comes with a serious drawback. All linked images and other external elements have to be put somewhere in the directory tree. This could easily lead to a huge amount of files that have to be handled (under version control) when working with one single SAP or AIM document. To avoid this, a number of metafile formats have been selected

for preferred use. These are namely the Visio *.vsd format and the Gumle format. In the following section, we will look into how to efficiently make use of such metafiles in some more detail.

4.5.5. Linked Metafiles

There are some different extended mechanisms implemented in the editor for combining several pieces of documentation within one file. One of them is to make use of metafiles in Visio and Gumle format. This is the favored variant within TI. Some additional variants will be described in a later section.

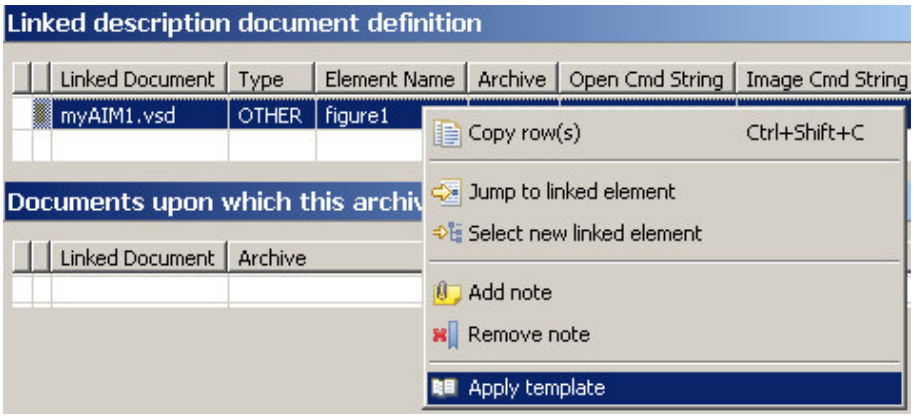
Figure 4.17. External Description Files



Usually, there should be at most one metafile for Visio format images (*.vsd) and two for Gumle format images (*.gumle and *.gumlx) for each single SAP/AIM document. Pragmatically, these should be given the same name as the XML document and should be located in the same directory. So, if we create an AIM document called myAIM which should contain links to Gumle and Visio drawings in its documentation part, there will be the files myAIM.aim, myAIM.vsd, myAIM.gumle and myAIM.gumlx in the end (Figure 4.17).

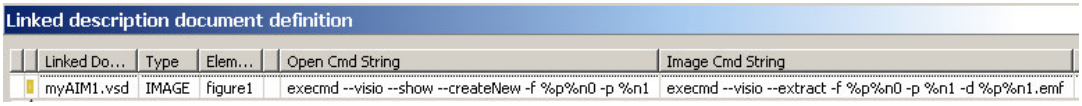
In order to make Sape understand which image from a metafile to place at a given link, some extended information is necessary. In the "Linked Document" column, the name of the metafile has to be put down. The name of the drawing in the file that should appear at the location of the link in the generated documentation has to be placed in the "Element Name" column. If, e.g. for Gumle, a drawing is located somewhere further down in the hierarchy of the metafile, the path has to be given together with the name of the drawing in the standard way, using "/". Additionally, the columns "Open Cmd String", "Image Cmd String" and "Type" must be filled with reasonable values. How this generally works is being explained in a later section. For standard uses, there exist predefined templates which fill in those columns automatically.

Figure 4.18. Select Template for Linked Elements



You may apply these templates via the "Apply template" entry in the pop-up menu of the "Linked Description Document Definition" table (Figure 4.18).

Figure 4.19. Template Applied



Depending on which template you select (e.g. "Visio", or "Gumle"), it will set the type and command string fields of the selected table entry (Figure 4.19) so that the selected tool can be used for handling external documentation without further manual effort. By double clicking on the "Linked Description Document Definition" table entry, the stated file is being opened in the appropriate tool and the stated drawing is selected. For Visio, if the metafile or sub element does not yet exist, it will be created upon first access. For Gumle, the metafiles must be already existing, but not yet existent drawings will be created automatically when opened.

4.5.6. Other External Link Concepts

In this section, alternate concepts for linking of external documentation are being introduced. These are not needed for everyday work and are not officially supported at this time. This means that use of the described features is not recommended except for highly specialized cases. Thus, most readers may as well skip the remainder of this section when reading through the manual.

Arbitrary Managed Files

There are two entries in the "Linked Description Document Definition" table for handling of archive files and the like. In the "Open Cmd String" field, you may put a string which is called upon double clicking on the respective table entry and which should call some tool with which it is possible to edit the described piece of documentation. In the "Image Cmd String" field, a string may be put which will be called by the automated documentation generation tool in order to extract the desired piece of documentation from the archive file.

In order to be able to specify reasonable commands to be executed with these two strings, additional information about names and paths may be needed. This is provided by wildcards which can be freely placed within the command strings. There are two types of wildcards, one, **%p**, gets replaced by the path to the main SAP/AIM document. The other one, **%n0..9**, contains the names of the linked document (%n0), the element name (%n1), and the names of the depending documents (%n2-%n9).

In order to create new templates for your own tools (latex, e.g.) or modify existing ones, go to "Sape->External Document Entries" in the preferences dialog.

Sections of HTML Documents

If you have a HTML document and want to include just part of it into the documentation, please

specify the values for the "Subsection start" and "Subsection end" columns so that they correspond to equally named anchor tags in your HTML document. The included part of the HTML will then start directly after the first occurrence of the tag from "Subsection start" or at the beginning of the document (when no fitting tag has been found) and it will end directly before the first occurrence of the tag from "Subsection end" thereafter or at the end of the document. The mentioned anchor tags are also named "bookmark" or "target" sometimes and should look like this in order to be detected by the parser (this is *not* case sensitive):

```
<a name="THIS_IS_THE_ANCHOR_NAME"></a>
```

Sape Archives

By giving a name for an archive file in the "Archive" field, it can be defined that the respective document is put in the named archive. If this archive is not yet existent, it will be created upon first use, containing an empty document. Otherwise, the document is added to the archive. The handling of such an archive is transparent for the user, editing of documents still works by double clicking on them. In one archive file there can be stored as many documents as seems reasonable. To keep merging simple, it is, however, preferably to use separate archive files for binary and text documents. Since the structure of the archive file itself is XML, when only storing XML files in an archive, the entire archive is an XML file itself. This way, merging is not complicated by this concept of document archives.

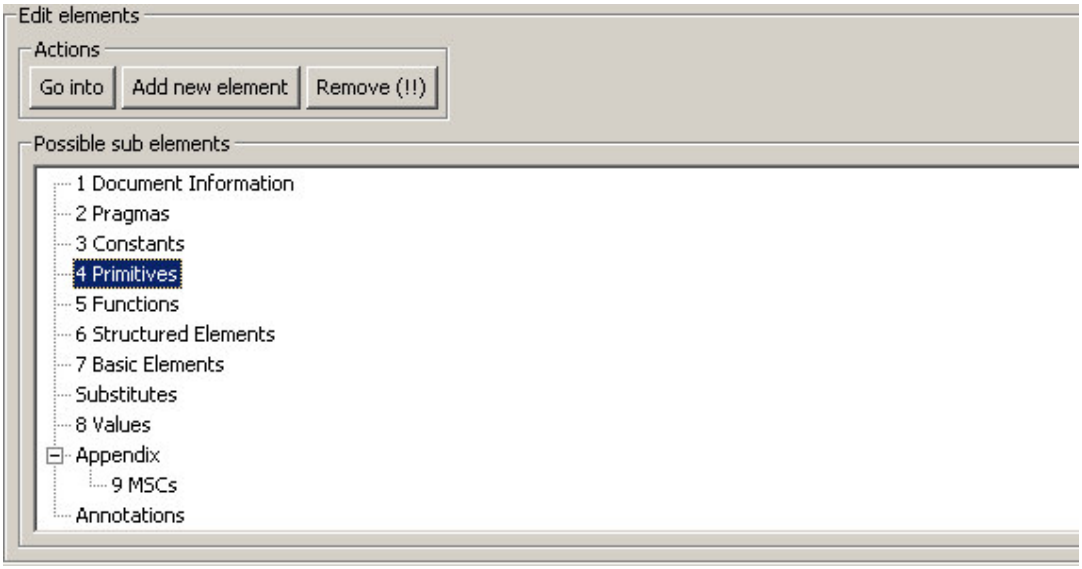
The "Documents upon which this archived document depends" table indicates which additional documents are needed for the main document to function properly. This is important mainly for some of the advanced functionality described below.

There are, however, some things to be kept in mind when working with archives. For one, it is not yet possible to mix archived files with not archived dependent files and vice versa. And, more importantly, automated integration of archived documents into the generated documentation is not yet supported which means that they will not appear there. This issue will be fixed in future versions of the converter tool.

4.6. Adding Contents

Now, after you have completed the document information page with some reasonable values, go to the SAP (respective AIM) page by clicking on the respective entry in the Outline.

Figure 4.20. Root Menu



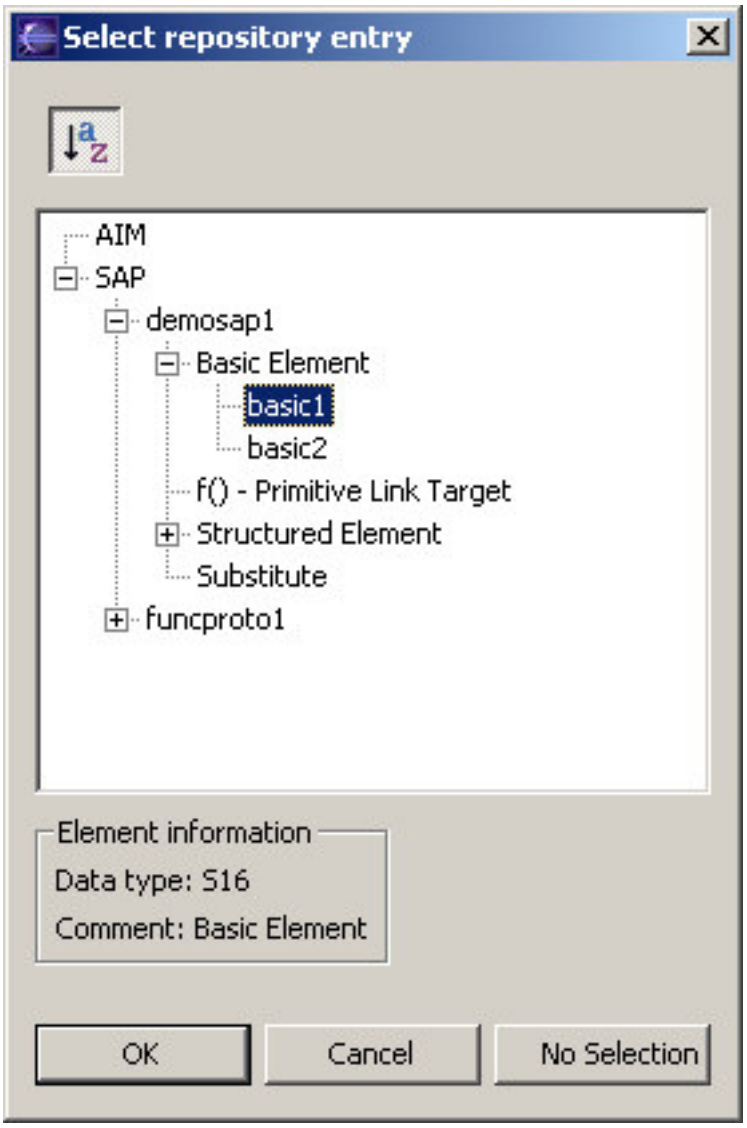
On the according editor page, you can see every kind of element which is allowed for this kind of document (Figure 4.20). When you click on one of the entries in the "Possible sub elements" window, some of the buttons in the "Actions" group will be activated. The "Go into" button leads to the content page of the selected section and is activated only if such a section really exists in the document. The remove button leads to complete removal of a whole section with all its sub elements, except for the document info section, which is mandatory for the document. The third button, "Add new elements", creates a new element of the selected kind in the respective group. If the group does not yet exist, it is created now. Otherwise, this is the same as adding a new element in a section overview page.

Since we want to create a new SAP, select the "Primitives" entry and press the "Add new element" button. A primitive section with a single new primitive is automatically created, as can be seen in the outline. Give the primitive some name and number in the "Primitive Definition" part, and provide a description and history entry for it. If you need different identifications for the same primitive contents, just create as many definition entries as you like. Finally, the newly created "Primitives" section needs some description and information about the stored primitives, too.

4.7. Links to SAP/AIM Elements

Now that we have created an empty primitive, we will add some information about its contents. Go to the newly created primitive and add a new element to the "Primitive Items". This element is a pointer to some other element, namely a basic, structured, substitute or function pointer element, out of some SAP or AIM document within the project. Since there are no such elements in our SAP yet, go back to the root page first ("SAP") and create a new Basic element in the same way you created the primitive. After you have filled in the relevant fields in the description, definition and history parts, go back to the newly created primitive and right click on the still empty new primitive item. Select the entry "Select new linked element" from the bottom of the appearing menu. In the appearing tree view

Figure 4.21. Repository Tree



of all documents in the project, go to the SAP document you are presently editing and select there from "Basic Element" the newly created basic element. If you want to toggle between alphabetical sort order and the real sequence of elements, please press the button in the top left of the dialog box. After pressing the "Ok" button, name and type of the selected element are displayed in the primitive item. When you want to have no linked element at all for this item, press the "No element" button instead. Information about the presently linked document is displayed as tool tips if you stay the mouse cursor in the related line of the table for a short time.

There is no difference between the handling of links to elements within the same document and those leading to other documents. If you want to see more information about the linked element, double click the line in the items table (or select "Jump to linked element" from the menu) and the page for the linked element will be opened. If this is located in another file, that file will be opened first and brought to top. The color of the icons in the second column of each table gives a hint about if double clicking on a line in that table will have any effect. If they are yellow, likely some linked detail information will open then, if they are black, you are already at the deepest possible level of detail.

4.8. Extended Functionality

It is possible to execute all navigation and functionality of the Sape editor via keyboard commands. Special functionality is also available through the toolbar and menues.

4.8.1. Toolbar and Menues

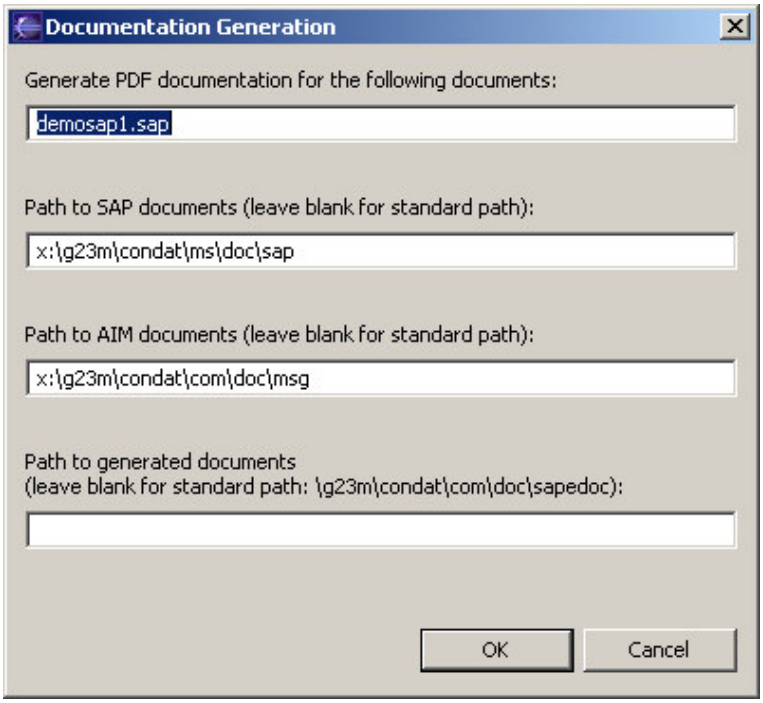
There are some more useful buttons and menu entries to be presented. For example, if you happen to have a lot of open sub windows in your editor window and cannot at all read the tabs at the bottom any more, try the "Close Page" action from the menu or the tool bar. This closes the actual tab and displays the one left of it. This button is also particularly useful to go back to the previous page after jumping to some linked element.

Then, there is the "Verify xml document" action which checks if the active document is valid in respect to the grammar for SAP/AIM documents. For each found error, a line number will be provided by which the problem can be located. Therefore, go to the first tab of the open editor window, named "XML", which displays the underlying code with line numbers. This way you can locate the problem and - hopefully - solve it later on in the editor.

A third action "Rebuild repository database" leads to a complete rebuild of the database which contains information about all linkable elements in all documents contained in the project. Normally, this database is updated automatically. But since this is a very complex task and, thus, does not always work as could be expected, pressing the rebuild button may solve quite some problems with links and such. It can also be very helpful when it comes to references because the "hot" update of those does not yet work properly which may lead to faulty updates of element names and to incorrect information in the "References" window.

Pressing the fourth of the addressed buttons leads to a dialog (Figure 4.22) which allows generation of documentation in PDF format directly from the Sape editor. This option only works when you have started the editor from a 4nt shell with initvars.bat correctly executed. Otherwise, please use the command line version of the documentation generation tool (see "readme_tools.txt" for further details).

Figure 4.22. Documentation Generation Dialog



Last, there are two more entries in the Sape editor menu. "Move up" moves selected table entry(s) up by one position and "Move down" moves them down accordingly.

There are some more functions you should know about. These are the "Find", "Undo", and of course the "Save" function. The latter, to be found in the "File" menu, does probably not need any explanation, as is the case for the belonging "Save as" function. With undo/redo, located in the "Edit" menu,

you can go back/forward through recent changes. By default, the editor remembers about the last hundred or so changes. You may try this out by undoing/redoing your latest changes. The easiest way to do this is to use the provided keyboard shortcuts, "CTRL+Z" and "CTRL+Y". Finally, find, located in the same menu, allows to search the active document for some string, sub string or regular expression. The name of the element from which the search starts is to be found in the title bar of the find dialog box. Depending on if you select "all" or "selected" for the search scope, the search will contain only the sub tree starting at the displayed element or all of the XML tree. Otherwise, the provided buttons and fields work as you will expect them to work. To try and test the find functionality, you now best open some existing document and search for some text contained in that document. For the XML display, a slightly different version of the find dialog is provided, which works basically in the same way.

4.8.2. Keyboard Navigation

All before mentioned custom commands can be executed through hot keys. Which key sequence is connected to which command is noted behind the name of the respective command in the menu.

Furthermore, it is possible to navigate the user interface using the keyboard only. For this, please use the following keyboard shortcuts:

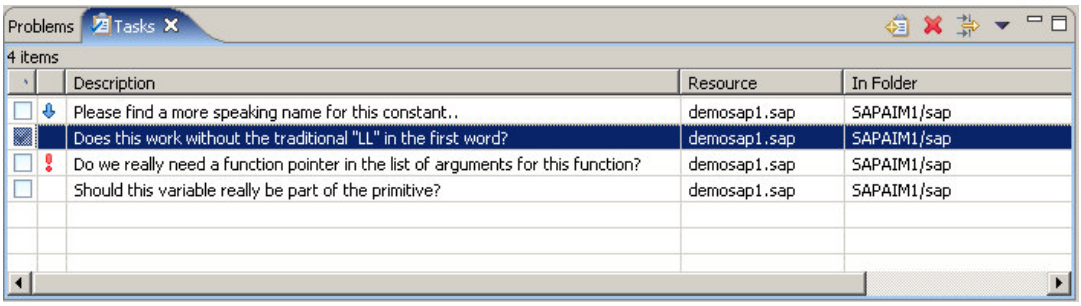
- To switch between open editors, press Ctrl+F6.
- To switch between open views (e.g., the Navigator and Outline windows), press Ctrl+F7.
- To switch between the open tabs of an editor, press Ctrl+PageUp and Ctrl+PageDown.
- To navigate between tables and other controls, use the Tab key. In order to jump out of text windows, use Ctrl+Tab instead.
- To select a specific row in a given table, use the Up/Down keys.
- To start editing in a table, select the row in question and press Ctrl+Enter. Press Esc to cancel editing mode.
- To edit the next/previous cell in a table row, press Ctrl+Shift+. and Ctrl+Shift+, .

For a complete list of Sape specific commands and hot keys, go to Window->Preferences->Workbench->Keys->SAP editor commands. You can also modify the keyboard shortcut settings there according to your own needs.

4.9. Notes

When you want to add notes, i.e., markers that stick to some table entry or text field (Figure 4.3, "Primitive Items"->"basic2"), to the document, just choose "Add note" from the respective pop-up menu and write down your note in the belonging dialog. This note will then be shown as a little icon in the very left column of the affected table row or, in the case of a text field, left of the actual text field as a little blue icon. For text fields, the pop-up menu is always accessible by right clicking on this very icon whether there are notes stored there or not (then it is just deactivated). The contents of the notes belonging to a document is being displayed in an extra window, the "Tasks" window (Figure 4.23).

Figure 4.23. Tasks Window



There, you can manipulate and delete your created notes as you see fit. By clicking on an entry in the list of the tasks-view, the editor will jump to the tree element the note belongs to.

4.10. Real Life

With this, the walk through tutorial is at its end. There is, of course, a lot more standard functionality to be found in the user interface of Eclipse and of the editor than is described here. Its use should be quite obvious and the handling intuitive in most cases. There are - as is the usual case with graphical interfaces these days - always several ways to obtain ones goals. Only one of them is being described here in most cases. Since the describing names are the same for every way of initiating an action, you may just choose the version you find most convenient. To learn more about the editor and to gain some hands on experience, it may be best to open some existing SAP or AIM document from one of the projects and try around a little.

Chapter 5. Additional Modules

With the initial package, two noteworthy additional modules, so called plugins, are provided. One of them is a standard text editor with which you can edit the XML code of a file directly if the need arises. To do so, close the file in question if already open and right click on it in the Navigator view. From the displayed pop-up menu, select "Open With->Text Editor". The functionality of the so opened editor is that known from other standard text editors. The other additional module allows for direct access to ClearCase functionality from within the Eclipse Navigator. To activate this, right click on the project you want to work with and select "Team->Associate with clearcase". Then, select "Team->Refresh State" from the pop-up menu for every folder you want to work with under ClearCase. From now on, the usual possibilities for handling of ClearCase objects are available from the "Team" entry in the pop-up menu.