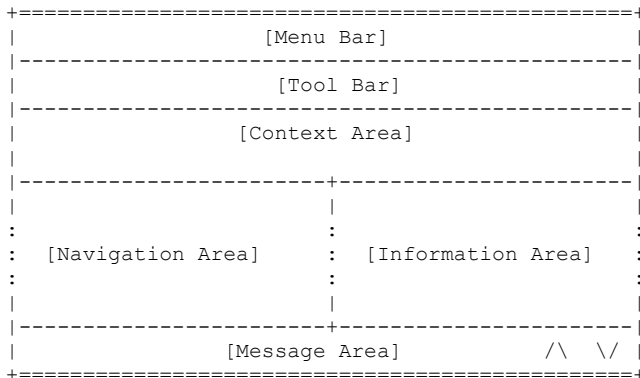


Browser - general

The browser is the starting point of your working session with ObjectTeam. You can navigate the repository from here, open objects and execute commands on these objects. The browser is always on a specific browser level.

Window sections

The default browser window contains the following areas. The Context Area, Tool Bar and Message Area can be toggled off and on by selecting the appropriate options in the View menu.



- **Menu Bar** - Allows you to select menu entries. The available menu entries per browser level can differ.
- **Tool Bar** - Allows you to more easily select frequently used menu entries.
- **Context Area** - Displays information about the current context (your current browser level).
- **Navigation Area** - Allows you to navigate the repository and load objects into the information area.
- **Information Area** - Displays the child objects of the current object.
- **Message Area** - Displays system messages. In Unix, you can browse through the message history using the up and down arrow symbols at the right of the Message Area.

Browser levels

The browser is always on a certain *level*. By default, it is on *Corporate* level after you start ObjectTeam. Which level the browser is on, depends on the browser object that is currently loaded:

CURRENT OBJECT	BROWSER LEVEL
<u>Corporate</u>	<u>Corporate level</u>
<u>Project</u>	<u>Project level</u>
<u>ConfigVersion</u>	<u>Configuration level</u>
<u>PhaseVersion</u>	<u>Phase level</u>
<u>SystemVersion</u>	<u>System level</u>
<u>SystemVersion</u>	<u>System level (implementation)</u>
<u>DocumentVersion</u>	<u>Document level</u>

You open an object in the Browser by clicking on it in the Navigation area or double-clicking on it in the Information area. You can also open an object by selecting it in the Information area and then selecting File > Open.

The available menus and the Browser objects that appear in the Information area are different for each Browser level.

However, although the sequence *Corporate - Project - ConfigVersion - PhaseVersion - SystemVersion* represents a logical hierarchy of objects, you don't have to open every object subsequently. You can unfold the appropriate objects in the Navigation area until you reach the level of the object you want to open. Finally, you open the intended object from the Information area or the Navigation area.

Browser - configuration level

In: Browser

With the browser on configuration level, you can do the following:

- Navigate the repository using the navigation area.
- Execute actions on objects available on this level using menus. Which operations you are allowed to execute on which objects is defined in your access rights.

Window sections

The default browser window contains the following areas:

- Menu Bar
- Tool Bar
- Context Area
- Navigation Area
- Information Area
- **Message Area** - Displays system messages. (Unix-only: You can browse through the message history with the arrow-up and arrow-down symbols at the right.)

Browser - Corporate level

In: Browser

The browser on corporate level is the starting point of your working session with ObjectTeam. From here, you can do the following:

- Navigate the repository using the navigation area.
- Execute actions on objects available on this level using menus. Which operations you are allowed to execute on which objects is defined in your access rights.

Window sections

The default browser window contains the following areas:

- Menu Bar
- Tool Bar
- Context Area
- Navigation Area
- Information Area
- **Message Area** - Displays system messages. (Unix-only: You can browse through the message history with the arrow-up and arrow-down symbols at the right.)

Browser - Phase level

In: Browser

With the browser on phase level, you can do the following:

- Navigate the repository using the navigation area.
- Execute actions on objects available on this level using menus. Which operations you are allowed to execute on which objects is defined in your access rights.

Window sections

The default browser window contains the following areas:

- Menu Bar
- Tool Bar
- Context Area
- Navigation Area
- Information Area
- **Message Area** - Displays system messages. (Unix-only: You can browse through the message history with the arrow-up and arrow-down symbols at the right.)

Browser - Project level

In: Browser

With the browser on project level, you can do the following:

- Navigate the repository using the navigation area.
- Execute actions on objects available on this level using menus. Which operations you are allowed to execute on which objects is defined in your access rights.

Window sections

The default browser window contains the following areas:

- Menu Bar
- Tool Bar
- Context Area
- Navigation Area
- Information Area
- **Message Area** - Displays system messages. (Unix-only: You can browse through the message history with the arrow-up and arrow-down symbols at the right.)

Browser - System level

In: Browser

The browser is on system level if a browser object of type SystemVersion is the current object. On this browser level, you can do the following:

- Navigate the repository using the navigation area.
- Execute actions on objects available on this level using menus. Which operations you are allowed to execute on which objects is defined in your access rights.

Systems with a PhaseVersion of type **Implementation** as parent object are reserved for Implementation purposes and have different characteristics than systems that have PhaseVersions of type **Analysis**, **SystemDesign** and **ObjectDesign** as parent object.

Window sections

The default browser window contains the following areas:

- Menu Bar
- Tool Bar
- Context Area
- Navigation Area
- Information Area
- **Message Area** - Displays system messages. (Unix-only: You can browse through the message history with the arrow-up and arrow-down symbols at the right.)

Browser - System level (implementation)

In: Browser

The browser is on system level (implementation) if both the following are true:

- A browser object of type SystemVersion is the current object.
- The parent object of this object is a PhaseVersion of type **Implementation**.

On this browser level, you can do the following:

- Navigate the repository using the navigation area.
- Execute actions on objects available on this level using menus. Which operations you are allowed to execute on which objects is defined in your access rights.

Systems with a PhaseVersion other than **Implementation** are reserved for design purposes and have different characteristics than systems with a PhaseVersion of type **Implementation**.

Window sections

The default browser window contains the following areas:

- **Menu Bar**
- Tool Bar
- Context Area
- Navigation Area
- Information Area
- **Message Area** - Displays system messages. (Unix-only: You can browse through the message history with the arrow-up and arrow-down symbols at the right.)

Move Definition, dialog box

In: Browser, Utilities | Move Definition...

SystemVersion All the systems of the current phase are listed here. Select the system you want to move the selected item definition(s) to.

Keep in mind that if you move an item of type **cl**, the corresponding cdm is moved to the new system too.

Overwrite Properties dialog box

In: Browser

In this dialog box you specify whether or not you want item property values to be overwritten during Merge From Previous Phase on system level.

The items in a system have property values assigned to them. These values can be different from the assigned values in the previous phase.

OK If you press this button, the property values involved are overwritten by the property values from the previous phase.

NO If you press this button, the current property values are not affected during Import From Previous Phase. However, properties that do not exist yet in the current system, are copied.

Press the *OK* button to confirm the current selection or the *Cancel* button to discard the operation.

Edit/Show Properties dialog box (browser objects)

The Edit/Show Properties dialog box has two sections:

- A *list box* at the left
Here you have to specify the object you want to edit or show the properties of
- A *book* at the right
Here you can edit or view the properties for the selected object. The book can contain more than one page:
 - **Text** : page of the property **Free Text** which you can set for every object.
Use this property to annotate a browser object. Free text is not bound to any syntax or constraints.
 - **Misc** : most of the properties can be found on this page.
Depending on the selected object, you can set or view the values of the following default properties:
 - File System Path Part
 - Document properties (DocumentVersions only)
 - Structure Generator (Local sections only)
 - Contents Generator (Local sections only)
 - Smalltalk System

You can flip pages by selecting a tab or by browsing through the page numbers at the bottom of the book.

Changing the location of an external link

(In: Browser)

An External Link represents a link to a file in the file system.

If you have created an External Link, and the location or the name of the file in the file system changes, you can use File | Change | Location... to make the link consistent again.

To change the location of an external link:

1. Select an file to which an external link refers.
2. Select **File | Change | Location...**
A **File selection dialog box** appears in which you can select the new location.

Changing the link status of a browser object or a UserRoleLink

(In: [Browser](#))

Link Status of a browser object

Use **File | Change Link Status...** to change the link status of the following browser objects:

- PhaseVersion
- SystemVersion
- < file version >

The link status of these objects determine the behavior of the objects when new versions of child objects are created. You can change their status in the [Change Link Status dialog box](#).

Change Link Status UserRoleLink

Use **File | Change Link Status...** to change the link status of a:

- UserRoleLink

The link status of a UserRoleLink determines whether or not the linked [role](#) is part of the user's initial [effective context](#). You can change this status in the [Change Link Status dialog box](#).

Changing the name of a Browser object

(In: Browser)

Use **File>Change>Name...** to change the name of the following:

- configuration
- project
- system
- document
- file
- external file

To change the name of one of these objects:

1. In the Information area of the Browser, select the object.
2. Select **File | Change | Name...**
The Change Name dialog box appears.
3. Enter the new name and click on OK.
If the specified name already exists, a Name Conflict dialog box appears.

Checking Global Model

In: Browser on System Level

Use **Check | Global Model** to check all working Class Diagrams in the current system. It checks all classes and events defined in these diagrams. Events can be specified in:

- Sequence Diagram: SD message flow
- Collaboration Diagram: COD message flow
- State Transition Diagram: Event Message

Reports on the checking process are displayed in a Monitor window.

The check rules depend on the current phase. For example, in the Object Design phase, one rule checks that every defined event has a corresponding operation. For details on the checking rules, refer to the *ObjectTeam Customization Guide*.

Checking Local Model

In: Browser on System Level

Use **Check | Local Model** to check selected classes, or all classes in a selected diagram. What is checked depends on the browser objects selected in the information area:

- **CD selected:** Only the events received by classes from the current Class Diagram are checked. The result is the same as when Check Local Model is invoked from the Class Diagram Editor.
- **COD selected:** Only instances and links of the selected Collaboration Diagram(s) corresponding to the classes and associations in the Class Diagram are checked.
- **SD selected:** Only the classes corresponding to the messages of the selected Sequence Diagram(s) are checked.
- **STD selected:** Only the classes corresponding to the Event Messages of the selected State Transition Diagram(s) are checked.

Reports on the checking process are displayed in a Monitor window.

The check rules depend on the current phase. For a comprehensive list of checking rules, refer to the *ObjectTeam Customization Guide*.

Closing a browser level

In: Browser

Use **File | Close** to close the currently open object, changing the browser level to the next level higher in the hierarchy.

Exporting M4 variables

(In: Browser)

Use **Options | Copy User Environment** to export one or more M4 variables from the current browser level to a browser level higher in hierarchy. The Copy User Env to ... Env dialog box appears.

Editing browser objects

In: Browser

To edit a Browser Object:

1. Select an object in the Information Area.
The type of object you can select to edit depends on the selected Browser level:
 - [Project Level]
 - [Configuration Level]
 - [Phase Level]
 - [System Level]
 - [Implementation System Level]
 - [Document Level]
2. Select **File | Edit**.

Filtering the Information Area

In: Browser

Use **View | Filter | Edit...** to filter out certain objects from the information area.

The **Edit<object>VersionDefault Filter dialog box appears.**

You can define one filter per browser level using the Filter dialog box.

The number of options in the dialog box varies with the level you are on.

The **Filter** field in the context area indicates whether a filter is *on* or *off*.

To turn a filter off, use View | Filter | Delete.

Deleting the filter for the information area

In: Browser

Use **View | Filter | Delete** to turn off the filter for the current browser level.

The **Filter** field in the context area indicates whether a filter is *on*.

Once deleted, a filter cannot be turned on again. You must recreate it using **View | Filter | Edit**.

Setting the Browser Font

The menu entry **Options | Font** allows you to change the font used in the navigation area and the information area of the browser.

The fonts and the font properties that you can select depend on your windowing system (X-Windows or MS-Windows).

Displaying information about a browser object

In: Browser

Use **File | Info** to view the characteristics of the selected object(s). These characteristics are displayed in the Info box.

Moving Item Definitions

In: Browser

Use the menu entry **Utilities | Move Definition** to move item definitions from one system to another. You can only move the definitions of items that have scope *phaseDef*.

To select the items whose definitions you want to move:

1. Open the pseudo object <defined items>, which appears under the browser object system. All items in that system appear in the Information Area.
2. Select the items that you are interested in.
3. Select **Utilities | Move Definition...**
The Move Definition dialog box appears.
4. Select the system that you want to move the definitions to
5. Press OK.

Opening browser objects

In: Browser

In the navigation area, click on the object. In the information area, double-click on the object, or select the object and then select **File | Open**. Depending on the status of the object and the type of object, the following happens:

- Project , Configuration, Phase, System (Status is irrelevant): the object is opened.
- FileVersion , Customization FileVersion and Status is Working: the appropriate editor is started (see File | Edit).
- FileVersion , Customization fileVersion and Status is Frozen: the object is opened as readonly (see File | Show).

If your access rights allow it, the object is opened and its child objects are displayed in the information area. The opened object become the **active** object. By opening certain objects, you change the current browser level.

Specifying a previewer for file sections

In: Browser

Use **Options | Previewer...** to specify the previewer(s) used when File | Preview is selected. The Previewer dialog box appears.

A previewer is used to display certain elements of documents, such as diagrams, PostScript files or plain text files.

Managing Browser Object Properties

(In: Browser)

Use the **Properties** menu to *edit*, *show* and *delete* property values of browser objects.

File | Properties | Edit...

Use this option to change the property values of the browser object using the Properties dialog box.

File | Properties | Show...

Use this option to view the property values of the browser object using the Properties dialog box.

File | Properties | Delete

Use this option to delete all the property values of the browser object.

Printing Browser Objects

In: Browser

You can use **File | Print** on the following browser levels:

System level Use this menu entry to print diagrams.

Implementation System level

Use this menu entry to print (generated) source files.

Documentation level

Use this menu entry to print file sections, local sections or property sections.

Diagrams and local sections are printed with the specified *Graphical printer*. Text files are printed with the specified *Text printer*. You can specify another printer and change some printer options using Options > Printer Setup.

Printing the Browser View

In: [Browser](#)

Use **File | Print View** to make a print of the information currently available in the [context area](#) and the [information area](#) (in *Details* view). To can change the print command that **Print View** uses, select [Options | Printer Setup | Text](#).

The output looks likes this:

```
+-----+
| Project:                docu_proj
| Configuration Version:  docu_conf.1
| Phase Version:         ObjectDesign.1
| System Version:        docu_sys.1
| View:                  Default
| Filter:                off
|
|
| Name      Type   Status  Link
|=====|
| Supplier.1 cad   working fixed
| Product.1  cad   working fixed
| ...      ...   ...    ...
```


Delete Unreferenced Items

In : Browser

Use **Utilities | Delete Unreferenced Items** to clean up the repository. It deletes items that are no longer used.

An item is made as soon as a diagram component is named. However, if you decide to delete the component from the diagram, the item is not deleted. Therefore, it is best to regularly use this menu entry to delete these unreferenced items from the repository. It may improve the performance and avoid problems within the same scope if you define another component with the same name and item type.

To delete all unreferenced items from one or more systems in your project: move to Phase level, select the systems, and then select this menu option.

To delete all unreferenced items from one or more files in a particular system: move to System level, select the files, and then select this menu option.

Show - menu entry

In: Browser

Use **File | Show** to open objects in read-only mode. The following objects can be opened in this way:

- FileVersion
The appropriate diagram editor is started in read-only mode.
- ExternalLink
The viewer is started in an Execution window.
- Customization file
Depending on the type of customization file currently selected, the customization editor or the viewer is started in read-only mode.
- Local section (Document level)
The word processor or DTP package is started, if possible in read-only mode.
- Property section (Document level)
The viewer is started in an Execution window.

Changing the way browser objects are displayed

(In: [Browser](#))

View | Refresh rereads the repository and refreshes the information area and the navigation area. The following **View** menu entries determine the format of the information displayed in the [information area](#).

View | Large Icons

Displays the objects in the information area as icons with the *Name* underneath every icon.

View | Small Icons

Displays the objects in the information area as small icons with the *Name* next to every icon.

View | Details

Displays the objects in the information area as rows, showing the (small) icon, the *Name*, the *Type* and the *Status* of every object.

What information you see in the information area is, to a large extent, determined by the current [view](#). The view called **Default** is the default view on most browser levels, but you can create your own views using the [View Customization Editor](#).

Selecting a View

(In: [Browser](#))

What you see in the [information area](#) is determined by a *view*. In a view, the following is defined:

- The objects that appear in the information area
- How these objects are sorted
- Which columns must be displayed in which order *Details view*

You can create your own views with the [customization editor](#). However, the following views are available by default:

[[Default](#)] [[Pseudo](#)] [[Data](#)] [[Process](#)] [[SQL](#)]

Default	This is the default view on all browser levels .
Pseudo	This is an alternative view to Default, available on all levels, showing the contents of the various pseudo-objects.
Data	This view is only available on system level . In this view, only the objects types that are data-oriented are displayed. These are: <ul style="list-style-type: none">• __CD• __COD
Process	This view is only available on system level . In this view, only the objects types that are process-oriented are displayed. These are: <ul style="list-style-type: none">• __COD• __SD• __STD
SQL	This view is only available on implementation system level . In this view, only (generated) SQL scripts are displayed.

Setting the Viewer...

In: Browser

Use **Options | Viewer...** to define the tools used when File | Show is selected.

The Viewer dialog box appears.

You can specify different viewers for different types of files.

Menu Bar - Browser (configuration level)

Below , the default menus of the browser on configuration level are listed. You can add new menus and redefine existing ones by using the Customization Editor.

[File] [Edit] [View] [Options] [Version] [Security] [Utilities] [Help]

- **File** menu
 - New
 - Delete ...
 - Close
 - Edit
 - Open
 - Show
 - Change
 - Link Status...
 - Properties
 - Edit ...
 - Delete ...
 - Show ...
 - Interface
 - Interface ...
 - Print
 - Print View
 - Exit
- **Edit** menu
 - Undo
 - Cut
 - Copy
 - Paste
 - Select All
 - Deselect All
- **View** menu
 - Refresh
 - ToolBar
 - ContextArea
 - MessageArea
 - Large Icons
 - Small Icons
 - Details
 - Filter
 - Edit ...
 - Delete
 - Default
 - Pseudo
- **Options** menu
 - Compare Command...

- Editor ...
- Font ...
- Previewer ...
- Printer Setup
 - Text ...
 - Graphical ...
- Viewer ...
- Copy User Environment
 - To Corporate Environment...
 - To Project Environment...
 - To ConfigVersion Environment...
- **Version** menu
 - Freeze ...
 - Unfreeze
 - New
 - Copy ...
 - Delete ...
 - Select ...
 - New ...
 - Selected ...
 - Deselect
 - Compare ...
- **Security** menu
 - Show Access Rights...
 - Effective Roles...
 - Activate Role...
 - Deactivate Role...
 - Role Rights...
 - Edit ...
 - Edit Controlled Lists...
 - Show
 - Show Controlled Lists
- **Utilities** menu
 - Clone
 - Monitoring Window...
 - Execution Window...
 - Reports
- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

Menu Bar - Browser (corporate level)

Below, the default menus of the browser on corporate level are listed. You can add new menus and redefine existing ones by using the Customization Editor.

[File] [Edit] [View] [Options] [Version] [Security] [Utilities] [Help]

- **File** menu
 - New
 - Delete
 - Close
 - Edit
 - Open
 - Show
 - Change
 - Link Status...
 - Properties
 - Edit ...
 - Delete ...
 - Show ...
 - Interface
 - Print
 - Print View
 - Exit
- **Edit** menu
 - Undo
 - Cut
 - Copy
 - Paste
 - Select All
 - Deselect All
- **View** menu
 - Refresh
 - ToolBar
 - ContextArea
 - MessageArea
 - Large Icons
 - Small Icons
 - Details
 - Filter
 - Edit ...
 - Delete
 - Default
 - Pseudo
- **Options** menu
 - Compare Command...

- Editor ...
- Font ...
- Previewer ...
- Printer Setup
 - Text ...
 - Graphical ...
- Viewer ...
- Copy User Environment
 - To Corporate Environment...
- **Version** menu
 - Freeze ...
 - Unfreeze
 - New
 - Copy ...
 - Delete ...
 - Select ...
 - New ...
 - Selected ...
 - Deselect
 - Compare ...
- **Security** menu
 - Show Access Rights...
 - Effective Roles...
 - Activate Role...
 - Deactivate Role...
 - Role Rights
 - Edit ...
 - Edit Controlled Classes...
 - Edit Controlled Lists...
 - Show
 - Show Controlled Classes
 - Show Controlled Lists
 - Show
- **Utilities** menu
 - Clone
 - Monitoring Window...
 - Execution Window...
 - Reports
- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

Menu Bar - Browser (phase level)

Below , the default menus of the browser on phase level are listed. You can add new menus and redefine existing ones by using the Customization Editor.

[File] [Edit] [View] [Options] [Version] [Security] [Utilities] [Help]

- **File** menu

- New
- Delete ...
- Close
- Edit
- Open
- Show
- Generate ...
- Uppdate Document Directory
- Change
 - Link Status...
- Properties
 - Edit ...
 - Delete ...
 - Show ...
- Info ...
- Print
- Print View
- Exit

- **Edit** menu

- Undo
- Cut
- Copy
- Paste
- Select All
- Deselect All

- **View** menu

- Refresh
- ToolBar
- ContextArea
- MessageArea
- Large Icons
- Small Icons
- Details
- Filter
 - Edit ...
 - Delete
- Default
- Pseudo

- **Options** menu
 - Compare Command...
 - Editor ...
 - Font ...
 - Previewer ...
 - Printer Setup
 - Viewer ...
 - Copy User Environment
 - To Corporate Environment...
 - To Project Environment...
 - To ConfigVersion Environment...
 - To PhaseVersion Environment...

- **Version** menu
 - Freeze ...
 - Unfreeze ...
 - New
 - Copy ...
 - Delete ...
 - Select
 - New ...
 - Selected ...
 - Deselect
 - Compare ...

- **Security** menu
 - Show Access Rights...
 - Effective Roles...
 - Activate Role...
 - Deactivate Role...
 - Role Rights...
 - Edit ...
 - Edit Controlled Lists...
 - Show
 - Show Controlled Lists

- **Utilities** menu
 - Clone
 - Class Browser
 - Monitoring Window...
 - Execution Window...
 - Move Definition...
 - Delete Unreferenced Items
 - Merge From Other Phase
 - Merge From Previous Phase
 - New Systems
 - Specific Systems...
 - Compare With Previous Phase...
 - Reports

- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

Menu Bar - Browser (project level)

Below, the default menus of the browser on project level are listed. You can add new menus and redefine existing ones by using the Customization Editor.

[File] [Edit] [View] [Options] [Version] [Security] [Utilities] [Help]

- **File** menu
 - New
 - Delete ...
 - Close
 - Edit
 - Open
 - Show
 - Change
 - Link Status...
 - Properties
 - Edit ...
 - Delete ...
 - Show ...
 - Ino ...
 - Print
 - Print View
 - Exit
- **Edit** menu
 - Undo
 - Cut
 - Copy
 - Paste
 - Select All
 - Deselect All
- **View** menu
 - Refresh
 - ToolBar
 - ContextArea
 - MessageArea
 - Large Icons
 - Small Icons
 - Details
 - Filter
 - Edit ...
 - Delete
 - Default
 - Pseudo
- **Options** menu
 - Compare Command...

- Editor ...
- Font ...
- Previewer ...
- Printer Setup
 - Text ...
 - Graphical ...
- Viewer ...
- Copy User Environment
 - To Corporate Environment...
 - To Project Environment...
- **Version** menu
 - Freeze ...
 - Unfreeze
 - New
 - Copy ...
 - Delete ...
 - Select
 - New ...
 - Selected ...
 - Deselect
 - Compare ...
 - Activate
 - Deactivate
- **Security** menu
 - Show Access Rights...
 - Effective Roles...
 - Activate Role...
 - Deactivate Role...
 - Role Rights...
 - Edit ...
 - Edit Controlled Classes...
 - Edit Controlled Lists...
 - Show
 - Show Controlled Classes
 - Show Controlled Lists
- **Utilities** menu
 - Clone
 - Monitoring Window...
 - Execution Window...
 - Reports
- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

Menu Bar - Browser (system level)

Below , the default menus of the browser on system level are listed. You can add new menus and redefine existing ones by using the Customization Editor.

[File] [Edit] [View] [Options] [Version] [Security] [Utilities] [Check] [Help]

- **File** menu
 - New
 - Delete ...
 - Close
 - Edit
 - Open
 - Show
 - Change
 - Link Status...
 - Location ...
 - Properties
 - Edit ...
 - Delete ...
 - Show ...
 - Interface
 - Interface ...
 - Print
 - Print View
 - Exit
- **Edit** menu
 - Undo
 - Cut
 - Copy
 - Paste
 - Select All
 - Deselect All
- **View** menu
 - Refresh
 - ToolBar
 - ContextArea
 - MessageArea
 - Large Icons
 - Small Icons
 - Details
 - Filter
 - Edit ...
 - Delete
 - Default
 - Data
 - Process
 - Pseudo

- **Options** menu
 - Compare Command...
 - Editor ...
 - Font ...
 - Previewer ...
 - Printer Setup
 - Text ...
 - Graphical ...
 - Viewer ...
 - Copy User Environment
 - To Corporate Environment...
 - To Project Environment...
 - To ConfigVersion Environment...
 - To PhaseVersion Environment...
 - To SystemVersion Environment...
- **Version** menu
 - Freeze ...
 - Unfreeze
 - New
 - Copy ...
 - Delete ...
 - Select ...
 - New ...
 - Selected ...
 - Deselect ...
 - Compare ...
 - Activate ...
 - Deactivate ...
 - Make Corporate...
 - Restore
 - Snapshot ...
- **Security** menu
 - Show Access Rights...
 - Effective Roles...
 - Activate Role...
 - Deactivate Role...
 - Role Rights...
 - Edit ...
 - Edit Controlled Lists...
 - Show
 - Show Controlled Lists
- **Utilities** menu
 - Clone
 - Class Browser
 - Monitoring Window...

- Execution Window...
- Move Definition...
- Delete Unreferenced Items
- Merge From Other Phase
- Merge From Previous Phase
- Compare With Previous Phase...
- Reverse Engineer
- Reports

- **Check** menu
 - Check Contents
 - Global Model
 - Use Case Model
 - Local Model

- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

ConfigVersion - browser object

In: Browser

The browser object ConfigVersion appears in the information area on:

- Project level

A configuration offers a working environment which is preferably used by only one user. By using different configurations within the same project, various project members or groups of project members can work at a project simultaneously and independently.

Entering a project, you choose a version of the configuration you are going to work in. Inside this ConfigVersion you can change the composition of the ConfigVersion by selecting other versions of objects.

You can also create new objects (or new versions of objects) in a ConfigVersion. The new versions of objects receive an identification of the configuration they are created in.

Versions of objects originating from other configurations can be selected in a configuration version. In this way, parallel developments can be joined.

The browser object ConfigVersion is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
|
+- ConfigVersion
|
+- PhaseVersion
|
+- SystemVersion
|
+- FileVersion
```


Corporate - browser object

In: Browser

The browser object Corporate is the entrance to the repository. All projects and their contents can be accessed through this object. The corporate object is a kind of rack every thing else is attached to at the highest browser level.

The browser object Corporate is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- SystemVersion
        |
        +- FileVersion
```

Since Corporate is the top object in a hierarchy of browser objects, it can only appear in the navigation area of the browser.

childRight - browser object

A browser object *childRight* specifies the *initial* role rights assigned to objects added to a **ControlledList**.

You can change the setting of the *childRight* role rights of a controlled list (provided you are authorized to do so) by using Security | Role Rights > Edit. Select an Object and a Role in the dialog box. By default, the ownRight role rights are shown. To set the *childRight* role rights, select *childRight* in the *Role Right* field.

You can *view* the *childRight* role rights of a controlled list by selecting Security | Role Rights > Show. You have to select the appropriate Controlled Object and the appropriate Role in the dialog box. By default, the ownRight role rights are shown. To view the *childRight* role rights, select *childRight* in the *Role Right* field.

Controlled Class

The role rights that you specify for a *controlled class* apply to all objects of that type. This provides an easy way to specify role rights for a large number of objects at one time.

ObjectTeam defines controlled classes on two levels:

- Corporate Level
- Project Level

To view the controlled classes defined by ObjectTeam, select Security > Role Rights > Show Controlled Classes on Corporate or Project level. To edit them, select Security > Role Rights > Edit Controlled Classes.

Controlled List - browser object

In ObjectTeam, a parent object has a controlled list for each of its child object types. When you create a child object, ObjectTeam adds that object to the appropriate controlled list on the parent object. For example, each System object has a controlled list, SystemVersionList. When you create a new version of the system, ObjectTeam adds the System version to SystemVersionList.

For each controlled list, you can specify two types of role rights:

- ownRight determines who can add objects to or remove objects from the controlled list.
- childRight specifies the initial role rights assigned to the objects added to the controlled list.

List of controlledLists

The following table lists all parent objects and their controlled lists. These naming conventions identify the contents of each list:

- xxxList lists child objects other than versions.
- xxxVersionList lists versions.
- xxxLinkList lists links between versions of the parent object and versions of the child object.

Parent Object	ControlledList
Corporate	ProjectList
Project	ConfigList CustomFileList LevelCustomFileLinkList PhaseList
Config	ConfigVersionList CustomFileList LevelCustomFileLinkList ConfigPhaseLinkList
Phase	CustomFileList PhaseSystemLinkList PhaseVersionList SystemList LevelCustomFileLinkList
System	CustomFileList ExternalLinkList FileList GroupList LevelCustomFileLinkList SystemCorporateLinkList SystemFileLinkList SystemGroupLinkList SystemVersionList
Group	GroupVersionList
File	FileVersionList
CustomFile	CustomFileVersionList

FileVersion - browser object

In: Browser

File Versions are stored on the following two browser levels:

- System level
- Implementation System level

On system level, they refer to diagrams, on implementation system level they refer to (generated) source files.

File Version on System Level

A *File Version* on system level can have one of the following diagram types:

- cd
 - cdm
 - cod
 - sd
 - std
 - ucd
-

File Version on Implementation System Level

The supported file types for a *File Version* on implementation system level depend on your target language. The available set of supported file types is usually a subset of the file types supported by the compiler of your target language.

The type **tcl** is supported for every target language. You can extend the code generation through files of this type.

The browser object *file version* is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- SystemVersion
        |
        +- FileVersion
```


ownRight

An *ownRight* specifies the role rights for the following objects:

- ControlledClass
- ControlledList

You can change the setting of the ownRight role rights of an object (provided you are authorized to do so) by using Security > Role Rights > Edit.

If you only want to *view* the setting of the ownRight role rights of an object, select Security > Role Rights > Show. You have to select the appropriate Controlled Object and the appropriate Role in the dialog box. By default, the ownRight role rights are shown. (You can select childRight in the *Role Right* field to see the childRight role rights.)

PhaseVersion - browser object

In: Browser

The browser object PhaseVersion appears in the information area on:

- Configuration level

A phase is a stage in the development in which the information system to be built is viewed from a specific angle. By default, the following phases exist in ObjectTeam:

- Analysis
- System Design
- Object Design
- Implementation

To create the separation of data between phases, the versions of objects in a phase receive the identification of that particular phase. In a PhaseVersion, no versions of objects can be selected that originate from another phase.

A phase is uniquely identified in a project by its *Name*. Of course, the order of the phases in a project is essential to allow special tools to copy data from one phase to the next or to compare this data with data from a previous phase. This order between phases is defined explicitly.

The browser object PhaseVersion is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- SystemVersion
        |
        +- FileVersion
```


Project - browser object

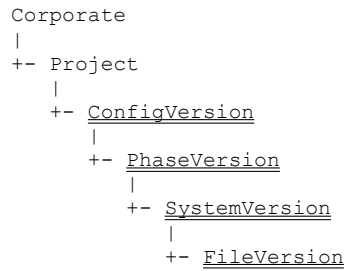
In: Browser

The browser object Project appears in the information area on:

- Corporate level

A project is a clear-cut piece of work that is executed in a company and that is quite independent from other projects. A project must result in an information system of some kind. To achieve that aim, a project can borrow pieces of other projects through the corporate model.

The browser object Project is part of a tree of (versions of) other objects:



Role - browser object

A Role can be specified on the following browser levels:

- Corporate level

A role is identified by a name. For each role, the users who can work in that role are specified at Corporate and/or Project level.

ObjectTeam defines several special roles:

Default	The default role has the same name as the user. This role is created when a new user is added to the repository using <u>File New User</u> . The user and role are linked at the Corporate level.
Guest	This role is assigned to users who are not registered in the repository, but can access it. It can be thought of as a default role for unknown users.
Superuser	This role is available on <u>corporate level</u> and on <u>project level</u> . A user creating a repository automatically becomes SuperUser for the entire repository. A user creating a project automatically becomes SuperUser for that project. Any user who can adopt the Superuser role has <u>access rights</u> to all objects at that browser level and below.

The link between a role and the user who is allowed to fulfill that role is represented by the browser object UserRoleLink. UserRoleLinks can be created at the Corporate or Project level. A user who is allowed to fulfill a role, can activate and deactivate their roles using Security > Activate Role and Security > Deactivate Role.

The browser object Role is part of a tree of other objects:

```
{Corporate | Project}
|
+- ...
|
+- ...
|
+- Roles
   |
   +- Role
      |
      +- UserRoleLink
```


SystemVersion - browser object

In: Browser

The browser object SystemVersion appears in the information area on:

- Phase level

A system is a logical and independent part of the information system to be developed as defined by the method. A system is a unit representing a division of the information system according to contents, as opposed to the configuration and the phase which are purely units of organization. A SystemVersion is a version of a system.

SystemVersions with a PhaseVersion **Implementation** as parent object are reserved for implementation purposes. SystemVersions with a PhaseVersion **Analysis**, **SystemDesign** or **ObjectDesign** as parent object are reserved for design purposes. Therefore, the characteristics of these two types of SystemVersions are different.

The browser object SystemVersion is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
|
+- ConfigVersion
|
+- PhaseVersion
|
+- SystemVersion
|
+- FileVersion
```


User - browser object

There are two types of users using ObjectTeam:

- Users who are registered in the repository
- Users who are not registered in the repository

A user can be registered in the repository using File | New | User on corporate level. The name of the browser object **User** must be the same as the user name in the operating system.

Users who are registered in the repository can be linked to certain roles with UserRoleLinks. For every role a user can fulfill theoretically such a UserRoleLink must exist. The available roles a user can fulfill can then be activated and deactivated. All the activated roles together make up the effective context of the user.

Moreover, access rules can be defined for users related to particular objects on particular browser levels. In this way, security can be customized in great detail.

Users who are not registered can only access ObjectTeam as guest; their effective context only consists of the role *Guest*.

The browser object User is part of a tree of other objects:

```
Corporate
|
+- ...
|
+- Users
   |
   +- User
```


UserRoleLink - browser object

A UserRoleLink can be specified on the following browser levels:

- Corporate level
- Project level

A UserRoleLink represents a link between a User and a Role, indicating the user can fulfill this role. A user cannot activate a role unless there is a UserRoleLink between the user and the role.

With a UserRoleLink, a status is saved indicating the way the role behaves in the effective context of the user. The status can have the following values:

- Default off* By default, the role is *off*; the user can activate the role if desired.
Default on By default, the role is *on*; the user can deactivate the role if desired.
Always on The role is always *on*; the user cannot deactivate it.

To change the status of a UserRoleLink, select File | Change | Link Status.

The browser object UserRoleLink is part of a tree of other objects:

```
{Corporate| Project}
|
+- ...
|
+- Roles
  |
  +- Role
    |
    +- UserRoleLink
```


Assigning a Role to a User

1. Make sure the browser is on corporate level or project level.
2. Open the role you want to give the user access to.
3. Select **File > New > User Role Link(s)**.
4. Select the appropriate User Name(s) from the dialog box and confirm your selection by pressing the OK button.
If your access rights allow you to, the new UserRoleLink(s) are created. The users can now add the role to their effective contexts.

You can change the status of the User Role Link by selecting File > Change > Link Status.

Changing options in the browser

In: [Browser](#)

In the browser on every level, you can use the [Options menu](#) to customize the browser. You can specify items such as the default printer, text editor, and previewer that you want to use.

Creating a New Role

1. Make sure the browser is on corporate level.
2. Open the browser object <roles>.
3. Select **File | New | Role**.
4. Enter the new role name in the dialog box and confirm it by pressing the OK button.
If your access rights allow you to, the new object Role is created.

After you have created a new role, you have to assign this role to one or more users so the users can activate it.

Creating a New User

1. Make sure the browser is on corporate level.
2. Open the browser object <users>.
3. Select **File > New > User**.
4. Enter the new user name in the dialog box and confirm it by pressing the OK button.
If your access rights allow you to, the new object User is created.

Moving an Item

Moving items is especially useful when splitting systems. Since items must always be defined somewhere, you can move them by hand to a new system.

1. Unfold the system in the navigation area.
The pseudo object <defined items> appears as child of the system.
2. Open the pseudo object <defined items>.
The items defined in the system appear in the information area.
3. Select the item(s) you want to move.
Note that you can only move items with scope *phaseDef*.
4. Select Utilities > Move Definition...
The Move Definition dialog box appears.
You can move item definitions to any system in the current phase.
5. Select the system you want to move the selected items to and click on OK.

When you move an item of type **cl**, the CDM, if there is one, is moved too.

Navigating the Repository

By folding and unfolding browser objects in the navigation area you can navigate the repository. Only objects preceded by an arrow pointing right (Unix) or a plus sign (Windows) can be unfolded.

1. To unfold an object, click on its preceding arrow or plus sign.
The child objects of the selected object appear and the navigation symbol changes into an arrow pointing down (UNIX-based systems) or a minus sign (Windows-based systems). These child objects can also be unfolded, if they have arrows pointing right or plus signs in front of them.
2. To fold an object, click on its arrow pointing down (Unix) or its minus sign (Windows).
The child objects of the selected object disappear and the navigation symbol now changes into an arrow pointing right (Unix) or a plus sign (Windows).

Besides folding and unfolding objects, you can also open browser objects in the navigation area or information area. If you open browser objects this way, the browser level changes and so do the contents of the information area.

Opening an Object

From the information area :

1. Move to the appropriate browser level.
2. In the information area, double-click on the object that you want to open, or select the object and then select File > Open.
If your access rights allow it, the selected object is opened and the browser level is changed.

Note : To find out details of a selected browser object, select File > Info.

From the navigation area:

1. Unfold the appropriate objects in the navigation area until the desired object is visible.
2. Click on the desired object.
If your access rights allow it, the selected object is opened and the browser level is changed.

Opening a Project

From the information area :

1. Make sure the browser is on corporate level.
2. Select the project you want to open in the information area.
3. Select File > Open.
If your access rights allow it, the selected project is opened and you move to project level.

From the navigation area:

1. Unfold the appropriate objects in the navigation area until the desired project is visible.
2. Click on the object.
If your access rights allow it, the selected project is opened and you move to project level.

Deleting an Browser Object

1. Make sure the browser is on the appropriate browser level.
2. Select the object you want to delete in the information area.
3. Select **File > Delete**.
If your access rights allow it, the selected object by the specified name is deleted from the corporate database.

Bear in mind:

- If you delete a parent object, all its children will also be deleted.
- If you want to delete a browser object that is versionable, and more than one version of the object does exist, only the selected version of the object will be deleted.
- If you delete the selected version of an object, all the other versions of the object do still exist. However, they are not visible in the browser: they exist in the background.
- If you want to delete a version other than the one currently selected, you have to use Version | Delete.
- A non-versionable object can be a child object of a versionable object. If you want to delete such an object, the status of the (versionable) parent object must be *working*.

Deleting a Project

1. Make sure the browser is on corporate level.
2. Select the project you want to delete in the information area.
3. Select **File > Delete**.

If your access rights allow it, the selected object project by the specified name is deleted from the corporate database.

Warning : All the configurations, phases, system and files residing under the selected project are also deleted.

Context Area (browser)

In: Browser

The *Context Area* in the browser is located below the tool bar and above both the navigation area and the information area. It shows the following information about the current object (if applicable):

<u>Project</u>	<u>Phase Version</u>
<u>Configuration</u>	<u>System Version</u>
<u>View</u>	<u>Filter</u>

Information Area

In: [Browser](#)

[[corporate level](#)] [[project level](#)] [[configuration level](#)] [[phase level](#)] [[system level](#)] [[system level \(implementation\)](#)] [[document level](#)]

The information area shows the child objects of the current browser object. In the information area, you can select objects and then select a menu item. This allows you to perform actions such as [editing](#), [version management](#), [deleting](#), and so on.

Which browser objects are displayed in the information area depends on the following factors:

- the [access rights](#) of the browser object
Here we assume all the access rights are enabled.
- the [current view](#)
Here we assume the default view.
- the current [filter settings](#)
Here we assume the filter is switched *off*.

From the information area, you can also change the current browser level by:

- [Opening objects](#)
- [Closing objects](#)

Information Area (corporate level)

The Information area discussed here is part of the [browser on corporate level](#).

Below at the right is a list of objects that the Information area can contain on Corporate level. At the left, the corresponding current objects are listed.

Current Object	Information Area
Corporate	Project
Customization files	Customization file
Corporate Groups	CorporateGroupVersion
CorporateGroupVersion	SavedGroupVersion
SavedGroupVersion	FileVersion
Roles	Role
Role	UserRoleLink
Users	User

In the [navigation area](#), you can browse through the hierarchy of objects.

Information Area (project level)

The Information area discussed here is part of the [browser on project level](#).

Below at the right is a list of objects that the Information area can contain on Project level. At the left, the corresponding current objects are listed.

Current Object	Information Area
Project	ConfigVersion
Customization files	Customization file
Roles	Role
Role	UserRoleLink

In the [navigation area](#), you can browse through the hierarchy of objects.

Information Area (configuration level)

The Information area discussed here is part of the [browser on configuration level](#).

Below at the right is a list of objects that the Information area can contain on Configuration level. At the left, the corresponding current objects are listed.

Current Object	Information Area
ConfigVersion	<u>PhaseVersion</u>
Customization files	<u>Customization file</u>

In the [navigation area](#), you can browse through the hierarchy of objects.

Information Area (phase level)

The Information area discussed here is part of the [browser on phase level](#).

Below at the right is a list of objects that the Information area can contain on Phase level. At the left, the corresponding current objects are listed.

Current Object	Information Area
PhaseVersion	<u>SystemVersion</u> <u>DocumentVersion</u>
Customization files	<u>Customization file</u>

In the [navigation area](#), you can browse through the hierarchy of objects.

Information Area (system level)

The Information area discussed here is part of the [browser on system level](#) and the [browser on Implementation system level](#).

Below at the right is a list of objects that the Information area can contain on System level. At the left, the corresponding current objects are listed.

Current object	Information Area
SystemVersion	<u>FileVersion</u> <u>GroupVersion</u>
Customization files	<u>Customization file</u>
Saved Groups	<u>SavedGroupVersion</u>
Defined Items	<u>Defined Item</u>
SavedGroupVersion	<u>FileVersion</u>

In the [navigation area](#), you can browse through the hierarchy of objects.

Information Area (document level)

The Information area discussed here is part of the [browser on document level](#).

Below at the right is a list of objects that the Information area can contain on Document level. At the left, the corresponding current objects are listed.

Current Object	Information Area
DocumentVersion	<u>dsm</u> <u>local section</u> <u>file section</u> <u>property section</u>
Customization files	<u>Customization file</u>

In the navigation area, you can browse through the hierarchy of objects.

Navigation Area

In: Browser

You can use the navigation area to view and browse through the structure of the repository. You can use it to see which child objects a particular object has.

In the Navigation area, you can navigate the repository by:

- unfolding objects
- folding objects
- opening objects by:
 - using File > Open
 - clicking on the object

An object has no child objects if it cannot be unfolded in the navigation area. If an object can be unfolded, but no child objects appear after doing that, then the child objects are invisible in the current view.

Checking Use Case Model

In: Browser on System Level

Use **Check | Use Case Model** to check the contents of all working Use Case Diagrams in the current system.

The following constraints must be met:

- For each Use Case there must be a UCD with the same name or at least one SD qualified by the name of the Use Case.
- Each Use Case must have exactly one initiator actor.

For details on the exact checking rules involved, refer to the *ObjectTeam Customization Guide*.

Changing Link Status dialog box

In: Browser

(PhaseVersion, SystemVersion, <file version>):

The link status of these objects is one of the following:

dynamicFrozen A *dynamicFrozen* link is updated if a new frozen version of the child object is created in another configuration of the project. With a *dynamicFrozen* link, you keep up with the latest frozen version of an object.

fixed The link is fixed on a specific version of the child object. The link is never automatically reset.

(User Role Link):

The link status of a UserRoleLink is one of the following:

alwaysOn The role cannot be deactivated; it is always part of the user's effective context.

defaultOff By default, the role is switched off. The role isn't part of the user's initial context. The user can activate the role if desired.

defaultOn By default, the role is switched on. The role is part of the user's initial effective context. The user can deactivate the role if desired.

Depending on your roles, you can change the status of a UserRoleLink. You can change the status of your own UserRoleLink, but only between Default Off and Default On. To change a status of Always On, or to change the status of another user's UserRoleLink, you need to have the SuperUser role active.

Edit Group Structure, dialog box

In: [Browser](#)

Subsections : [[Mode](#)] [[New](#)] [[Show](#)] [[Delete](#)] [[Contents](#)]

Use this dialog box to specify the [files](#) and the [items](#) you want to add to the current [group](#). The Edit Group Structure dialog box can be divided into the following sections:

- [Mode](#) , a list button to choose a selection mode
- Buttons : [New](#), [Show](#), and [Delete](#) to add, view and remove filters and items from the group
- [Contents](#) : the area in which you can view the results of your (filter)selections.

Mode

<i>overview</i>	In this mode, all files and items of this group version are listed. What you see in the <i>Contents</i> list box is the aggregation of the results of all the following filter modes.
<i>explicit file</i>	In this mode, all files added using explicit file mode are listed in the <i>Contents</i> list box.
<i>file filter</i>	In this mode, initially, all files added using file filters are listed in the <i>Contents</i> list box. You can use Show to list only those files added using a particular filter.
<i>file selector</i>	In this mode, all items and files in the group are listed in the <i>Contents</i> list box. It is not available in the Implementation phase.
<i>item filter</i>	In this mode, initially, all items added using item filters are listed in the <i>Contents</i> list box. You can use Show to list only those items added using a particular filter.
<i>item selector</i>	In this mode, all items and files in the group are listed in the <i>Contents</i> list box. It is not available in the Implementation phase.

New...

<i>explicit file mode</i>	In this mode, New allows you to select one or more specific file(s) to add to the group version.
<i>file filter mode</i>	In this mode, New allows you to specify a file filter, which filters on name, file type, or property values. The files matching the filter are added to the current group version. File filters are created with the New File Filter dialog box .
<i>file selector mode</i>	In this mode, New allows you to add files to the group based on what items are already in the group. You specify file selectors in the New File Selector dialog box .
<i>item filter mode</i>	In this mode, New allows you to specify an item filter, which filters on name, type, or property values. The items matching the filter are added to the current group version. Item filters are created with the New Item Filter dialog box .
<i>item selector mode</i>	In this mode, New allows you to add items to the group based on what files are already in the group. You specify item selectors in the New Item Selector dialog box .

Show...

Available only for filter modes. Select this button to display a dialog box that lists the filters defined for the currently selected *mode*. If you only want to see the files and items that are the result of a specific filter, select this filter rule in the dialog box (without pressing *OK*). You can select more than one filter rule.

Delete...

Use this button to remove items, files, or filters selected or defined using the currently selected *mode*. When you select this button, a dialog box appears listing the items, files, or filters, select one or more and then press the *OK* button.

Contents

This is the list box listing the files and items that are in the group or in the group as a result of the selections in the current mode. Overview mode displays all files and items in the current group.

Explicit File <GroupVersion>

In: Browser

In the Edit Group Structure dialog box, use **Explicit File** mode to add selected files to the group.

Adding a file adds an item When you add a file to the group, ObjectTeam also adds the item associated with that file. The Contents list box displays the files added to the group using Explicit File mode.

Notes :

- To see both the files and items that are in the group, select Overview mode.
- The Show button is unavailable in Explicit File Mode.
- In Explicit File mode, you can only remove files that were added using Explicit File mode.

To add one or more selected files to the group:

1. Select Explicit File mode.
The Contents list box displays the files added to the group using Explicit File mode.
2. Click on the New... button.
*The **New Explicit File dialog box** appears, displaying a list of all files in the current system that are not in the group.*
3. Select one or more files, then select OK.
ObjectTeam adds the files and their items to the group and updates the Contents list box, displaying the files added to the group.

To remove one or more files from the group:

1. In Explicit File mode, click on the Delete... button.
A dialog box appears, displaying the files added using Explicit File mode.
2. Select one or more files, then select OK.
ObjectTeam removes the files and their items from the group and updates the Contents list box.

New File Selector dialog box

In: Browser

In the File Selector dialog box you specify selection criteria that adds files to the current group based on what items are already in the group. The selection criteria work as follows.

Note : When you add a file to the group, ObjectTeam also adds the item associated with that file. For example, if you specify selection criteria that adds the CoreClasses CD to the group, ObjectTeam adds to the group both the file CoreClasses of type cad and the item CoreClasses of type cl.

item type ObjectTeam collects all items that are already in the group *and* that have one of the item types selected in this field:

- **cl** : class item
- **de** : data element
- **et** : event trace
- **pe** : process element
- **st** : state

decomp flags ObjectTeam then uses the collected items and the decomp flags selected in this field to locate files that are not already in the group. The decomp flags are used as follows:

- **Files** : Locates any file attached to any of the items.
- **Components** : Locates any file that contains a component that is attached to any of the items.
- **Parents** : Locates any file that contains a parent component (for example, a super class) that is attached to any of the items.
- **Leafs** : Locates any file that contains a leaf component (for example, a sub class) that is attached to any of the items.

file types Finally , ObjectTeam adds to the group any of the located files that have a file type selected in this field.

Note : Per item type, you can specify one or more Decomp Files and one or more File Types. The item types that are already part of a selector do not appear in the list, and you cannot change an existing selector. You can use the Delete button in the Edit Group Structure dialog box to delete an existing selector and create a new one.

New Item Selector dialog box

In: [Browser](#)

In the **Item Selector** dialog box you specify selection criteria that adds items to the current group based on what files are already in the group. The selection criteria work as follows.

file type ObjectTeam collects all files that are already in the group *and* that have a [file type](#) selected in this field.

item types ObjectTeam then adds to the group all the items in all the collected files that match the criteria specified in the *type* and *qualified* fields. The *qualified* field for each [item type](#) can be **yes** or **don't care**:

- If you specify **yes**, all qualified items of the specified type are added to the group.
- If you specify **don't care**, all items of the specified type are added to the group.

Note : Per file type, you can specify one or more item types. The file types that are already part of a selector do not appear in the list, and you cannot change an existing selector. You can use the [Delete](#) button in the [Edit Group Structure dialog box](#) to delete an existing selector and create a new one.

Comparing Previous Phase dialog box

(In: Browser)

Systems / Files / Components

Select the range of comparison.

Compare For: If you select *All Object*, all objects within the range are compared. If you select *Selected Objects*, only those objects currently selected in the information area of the browser are compared.

Phase : Select the phase against which you want to compare the objects of the current phase.

Press the *OK* button to confirm the current selection or the *Cancel* button to discard the operation.

Copy User Env To ... Environment dialog box

(In: [Browser](#))

In this dialog box you select the M4 variables you want to export. The variables listed here correspond to the ones in the **Meta4UserEnv** file. However, in this dialog box you cannot see *which* M4 variable is mapped to which option. In the following Tcl file you can:

```
<M4_home>/tcl/m4vardescr.tcl
```

Copying M4 variables results in the creation of a customization file *m4env* on the specified browser level. If the customization file *m4env* already exists on the specified level, it is **replaced** by the new one. So in that case, all the M4 variables that were set in the original *m4env* file are lost.

Edit <object>VersionDefaultFilter dialog box

(In: Browser)

You can filter on one or more of the following characteristics. Only objects that satisfy the specified criteria appear in the Information area. An asterisk (*) indicates that no filter is set.

Name Enter a string here to filter on the *Name* of objects.
You can use wild cards.

Type Filters on the *Type* of objects.
The type of object(s) that can be filtered out is different for every browser level.

Version Enter a string here to filter on the configuration of the objects. The format of the string is:

<ConfigVersion_name>.<ConfigVersion_number>

You can use wild cards.

Status Enter one of the following strings to filter on the Status of objects:

- working
- frozen
- background

You can use wild cards.

Link Enter one of the following strings here to filter on the Link Status of objects:

- fixed
- dynamicFrozen

You can use wild cards.

In Corporate Enter one of the following strings here to filter on objects from the Corporate Model:

- Yes
- No

Info Browser Object box

In: [Browser](#)

Select an object type from the (alphabetically ordered) list below to find more information on the object:

[Access]	[Link]
[Comments]	[Name]
[Controlled Actions]	[Role]
[Created]	[Status]
[Documented System]	[Text]
[Editor]	[Type]
[Frozen]	[Updated]
[Identity]	[Version]
[In Corporate]	

Access	(ownRight only) The setting of the controlled action involved.
Comments	If the status of the selected object is <i>frozen</i> , this entry shows the comment that you may have entered when you froze the object using Version Freeze . If the selected object is a group, it may show the comments you entered when you made a snapshot of the group using Version Snapshot .
Controlled Actions	These are the actions you are allowed to carry out on the selected object in your current effective context .
Created	This entry shows the date and time as to when the the selected object was created using File New .
Frozen	If the status of the selected object is <i>frozen</i> , this entry shows the date and time when the selected object was frozen using Version Freeze .
Documented System	This entry is displayed only for objects of type DocumentVersion . It displays the name of the system the document documents.
Editor	This entry is displayed only for objects of type DocumentVersion . It indicates which word processor or Desk Top Publishing package you selected when you created a new document using File New Document the system.
Identity	This string represents how the selected object is known within the repository.
In Corporate	If the selected file or diagram is part of a corporate group , this object is set to Yes.
Link	This entry displays the link status of the selected object.
Name	The name the selected object is known by in the browser.
Role	(ownRight only) The role for which the controlled action involved is defined.
Status	Indicates the status of the selected object: <ul style="list-style-type: none">• working• frozen
	The status of File Sections can be: <ul style="list-style-type: none">• fixed• selected
Text	Some explanatory text on the selected object.
Type	Within the environment of the browser, objects are recognized by their <i>Type</i> .
Updated	This entry shows the date and time when the the selected object was last edited using File Edit .
Version	This entry shows the current version of the selected object, which includes the name of the Configuration in which the object was created.

New Configuration

In: Browser

File | New | Configuration Version...

Conditions :

- The browser must be on project level.

You specify the new Configuration Version in the New Object dialog box.

New Customization File Version

In: Browser

File | New | Customization File Version...

Conditions :

- The current object must be *<customization files>*.
- The new customization file must not yet exist on the current browser level.

You specify the new customization file in the New Customization File Version dialog box.

New Document

In: Browser

File | New | Document Version...

Conditions :

- The browser must be on phase level.
- The status of the current object must be *working*.

You specify the new document in the New Document Version dialog box.

New External Link

In: Browser

Use **File | New | External Link** to create a new External Link.

Conditions :

- The browser must be on system level or implementation system level.
- The status of the current object must be *working*.

You select the external link in a file selection dialog box.

New File Versions

In: Browser

File | New | <file version>

Conditions :

- The browser must be on system level or Implementation system level.
- The Status of the current object must be *working*.

You specify a new file version in the New Diagram/File dialog box.

New File Section

In: Browser

File | New | File Section(s)...

Conditions :

- The browser must be on document level.
- A system to be documented must be available.

You select a file section in the New File Section dialog box. In this dialog box, the *Type* of an available file section, its *Name*, and the System in which it is defined are displayed.

New Group

In: Browser

File | New | Group Version

Conditions :

- The Browser must be on System level or on Implementation System level.

You specify a new group version in the New Diagram/File dialog box.

New Local Section

In: Browser

File | New | Local Section(s)...

Conditions :

- The browser must be on document level.

You specify the new local section in the New Local Section dialog box.

New Phase

In: Browser

File | New | Phase Version...

Conditions :

- The browser must be on configuration level.
- The status of this object must be of Type *working*.
- There is at least one Phase in the current Project that has no PhaseVersions with the current ConfigVersion as parent.

You select a new Phase in the New Phase Version dialog box. In the default situation, you can select the following Phases:

- Analysis
- SystemDesign
- ObjectDesign
- Implementation

New Project

In: Browser

File | New | Project...

Conditions :

- The browser must be on corporate level.

You specify the new project in the New Project dialog box.

New Property Section

In: Browser

File | New | Property Section(s)...

Conditions :

- The browser must be on document level.
- A system to be documented must be available.

You select a property section in the New Property Section dialog box. With this dialog box, you can create new item property sections and new file property sections.

New Role

In: Browser

File | New | Role...

Conditions :

- The browser must be on corporate level or on project level.
- The current object must be of Type *Roles*.

You specify the new Role in the New Role dialog box.

New System

In: Browser

File | New | System Version...

Conditions :

- The browser must be on phase level.
- The status of the current object must be *working*.

You specify the new System in the New System dialog box.

New User

In: Browser

File | New | User...

Conditions :

- The browser must be on corporate level.
- The current object must be of Type *Users*.
- The role SuperUser must be part of your effective context.

You specify the new User in the New User dialog box. For every new user you create, a role by the same name is created as well.

New User Role Link(s)

In: Browser

File | New | User Role Link(s)...

Conditions :

- The browser must be on corporate level or on project level.
- The current object must be of type *Role* or *User*.
- There must be at least one User who cannot play the current Role yet, or at least one Role not assigned to the current User yet.

By creating a new *User Role Link*, you make a role available to a user. In the New UserRoleLink dialog box, you can select the User(s) you want to assign the current Role to or select the Role(s) you want assigned to the current User.

Previewer dialog box

In: Browser

Previewer

Enter the name of the preferred preview program here. Keep in mind that you define this previewer only for the file type currently selected in the *Context* field.
The default previewer is defined in the M4_variable **M4_previewer**.

Window Previewer

Switch this button *on* if the defined previewer needs an Execution window to start up with. Switch it *off* if it doesn't.

This setting is defined in the M4_variable **M4_win_previewer**.

Context : Select the file type here for which the currently specified *Previewer* must be used. Keep in mind that a separate *Previewer* has to be specified for every *Context*. You can select the following file types:

- none
- **DoctextSection** : Plain ASCII text
- **EpsfSection** : Encapsulated PostScript with TIFF preview
- **EpsSection** : Encapsulated PostScript without preview
- **PropertySection** : Plain ASCII text
- **PsSection** : PostScript
- **EpsiSection** : Encapsulated PostScript with bitmap preview

Version - Make Corporate dialog box

In: Browser

Name (compulsory)

Define a name for your corporate group. Default it is the name of the Saved Group Version.

Comment (optional)

Fill in comments.

Setting the Viewer...

In: Browser

Viewer

Enter the name of the preferred viewer tool here (for example, **more** in Unix). Keep in mind that you define this tool only for the file type currently selected in the *Context* field. The default viewer is defined in the M4_variable **M4_viewer**.

Window Viewer Switch this button *on* if the defined tool needs an Execution window to start up with. Switch it *off* if it doesn't. This setting is defined in the M4_variable **M4_Xviewer**.

Context : Select the file type here for which the currently specified *Viewer* must be used. Keep in mind that a separate viewer has to be specified for every *Context*. You can select the following file types:

- none
- **DocTextSection** : Plain ASCII text
- **PropertySection** : Plain ASCII text

Version - Merge

In: [Browser](#), [Version Browser](#)

You can merge configuration, phase, and system versions that have a status of working or frozen. You cannot merge versions that have a status of background. If the status of the target version is working, the source elements are merged into that target version. If the status is frozen, ObjectTeam creates a new version of the target object and merges the source elements into that new version.

To merge two configurations:

1. In the Browser, open the Project that contains the configurations.
2. Select the target configuration.
3. Select **Version | Merge**.
*The **Select Source dialog box** appears.*
4. Select the source configuration and select OK.
*ObjectTeam examines each object in the source, determines how it should be merged into the target, and displays the results in the **Merge window**.*
5. Examine the results, as described in [Reviewing the Results in the Merge Window](#)
6. Optionally , merge the source elements into the target

To merge two versions of a phase or system:

1. Open the **Version Browser** for the phase or system.
2. Select the target version.
3. Press and hold the Control key while you select the source version.
4. Select **Version | Compare**.
ObjectTeam examines each object in the source, determines how it should be merged into the target, and displays the results in the Merge window.
5. Examine the results, as described in [Reviewing the Results in the Merge Window](#)
6. Optionally , merge the source elements into the target.

Version - Create Merge Link

In: Merge Tool

To create a Merge Link in the Merge Tool:

1. In the Merge Tool, select the source object from which you want to create a Merge Link.
2. Select **Version | Create Merge Link**.
ObjectTeam adds to the target object version a Merge Link that points from the source version to the target version.

Note . There may be Merge conflicts when you are creating Merge Links while merging from another or the previous phase. See the Project Management Guide for more details on these conflicts.

Version - Overwrite

In : [Version Browser](#)

Select **Version | Overwrite** when the source element conflicts with the target and you want to use the source version of the selected element instead of the target version.

The Overwrite command is only available when the source element conflicts with the target.

Creating New Browser Objects

In: Browser

Use **File | New** to create new browser objects in the repository. Which browser object you can create is mainly determined by:

- The current browser level
- The browser object that is currently loaded
- Your access rights

The following table displays which default browser objects you can create on which browser level.

Browser Level	New Browser Objects
<u>Corporate level</u>	[<u>Project</u>] [<u>User</u>] [<u>Role</u>] [<u>UserRoleLink</u>]
<u>Project level</u>	[<u>Configuration</u>] [<u>UserRoleLink</u>]
<u>Configuration level</u>	[<u>Phase</u>]
<u>Phase level</u>	[<u>SystemVersion</u>] [<u>DocumentVersion</u>]
<u>System level</u>	[<u>FileVersion</u>] [<u>GroupVersion</u>] [<u>ExternalLink</u>]
<u>Document level</u>	[<u>DocumentStructureMatrix</u>]
All levels	[<u>CustomizationFile</u>]

Show active modules

In : Browser

To see which modules are currently active, in the Browser:

1. Select **Utilities | Show Active Modules**.
ObjectTeam displays a dialog box listing the currently active modules.

Version - Activate

In: Browser

You can use this menu entry

- on Project level to activate a Configuration version that has been deactivated, or
- on System level to add corporate groups to the current system.

To activate a Configuration Version:

1. Select an FileVersion on System level.
2. Select **Version | Activate**.
*The status of the FileVersion changes from **backGround** to **frozen**, and moves it from the file system into the repository.*

Note : Use Version | Deactivate to deactivate a Configuration version.

To add corporate groups:

1. Go to the system to which you want to add a Corporate group
2. Select **Version | Activate ...**
The Activate Version dialog box appears.
3. Select the Corporate Group that you want to add and press OK.
The added objects have status reused and the corporate flag is set to yes.

Notes :

- Objects from corporate groups cannot be versioned. Also, they cannot be deleted individual, only the corporate group as a whole can be deactivated.
- You can also activate corporate groups by dragging them from the <corporate groups> pseudo object in the navigation area to the desired system in the information area.

Version - Compare

In: Browser,

You can compare an active version of a diagram or text file with another version of the same object, using **Version | Compare** menu entry.

Regarding diagrams, the following elements are compared:

- Nodes
- Connectors
- Node connectors
- Rows

Text files are compared using a compare command of the operating system. You can change the default command using **Options | Compare Command**.

To compare a version:

1. Select a browser object on any System level or Document level.
2. Select **Version | Compare**.
The Compare Version dialog box appears.
3. Fill in the dialog box and press OK.
4. The result of the comparison is displayed in a Monitoring Window.

Version - Copy

In: Browser

Use **Version | Copy** to copy the contents of another version to the version currently selected in the information area.

The copied version can be a frozen version of the same object, or a working or frozen version of any compatible object in the Project.

Versioning - Deactivate

In: Browser

You can use this menu entry

- on Configuration level to deactivate a Configuration version that has been deactivated, or
- on System level to remove corporate groups to the current system.

To deactivate a Configuration version:

1. Select an FileVersion on System level.
2. Select **Version | Deactivate**.
*The status of the FileVersion changes from **frozen** to **backGround**, and moves it from the repository into the file system.*

Note : Use **Version | Activate** to activate a activate a Configuration version.

To remove a corporate group:

1. Go to the system from which you want to remove a Corporate group and select the Corporate Group.
2. Select **Version | Deactivate ...**
The Deactivate Version dialog box appears.
3. Select the Corporate Group that you want to remove and press OK.

Version - Delete

In: Browser

Use this menu entry to delete one or more *leaf* versions of the selected object from the repository. (*Leaf* versions are those versions that have not been used to create subsequent versions.)

To delete an object:

1. Select a browser object on any level except Corporate with a working status.
2. Select **Version | Delete**.
The Delete Warning dialog box appears.
3. Press OK if you are sure that this is the object you want to delete.

Note :

If you only want to delete the *selected version* of an object, you can also use File | Delete. Deleting all the versions of a selected object results in the removal of the entire object.

Version - Deselect

In: Browser

Use this menu entry to move an entire object to the background. This can be useful if you don't want your information area being cluttered with objects you are not working on at the moment anyway.

To deselect a version:

1. Select a version on one of the following levels
 - Configuration level
 - Phase level
 - System level
 - Implementation System level
 - Document Level
2. Select **Version | Deselect**.
The object is removed from the information area and becomes invisible in the browser.

Note

Although a deselected object is invisible in the browser, it is not deleted, since you can put it back in the foreground again by using **Version | Select**.

Version - Freeze

In: Browser

Use this menu entry to freeze an object with status working, including all its child objects. If you freeze a system, for instance, all the diagrams and other objects residing in that system are frozen as well.

You cannot change an object with status frozen. However, you can unfreeze it, and perform other versioning operations like making, deleting, selecting and comparing version.

To freeze an object:

1. Select a browser object on any level except Corporate with a working status.
2. Select **Version | Freeze**.
The Freeze Version dialog box appears.
3. Fill in the dialog box and press OK.

Notes :

CD : If you freeze a Class Diagram, the belonging Class Definition Matrices are not automatically frozen. You must freeze these separately.

GroupVersion : If you freeze a group, you only freeze the information on how the group is structured; not the files that are part of the group.

dynamic frozen: If you freeze an object that has a link status *dynamic frozen*, all links to this object are rerouted to the frozen version. If user A for instance creates system X and user B has this system in his configuration as well, and user A freezes system X, the link user B has to system X will automatically be rerouted to the frozen version.

Version - Make Corporate...

In: Browser

You use **Version | Make Corporate** to promote a saved group version to a corporate group, provided your access rights allow you to do so.

To make a saved group version corporate:

1. Select a SavedGroupVersion in a <saved groups> object (available on any system level).
2. Select **Version | Make Corporate ...**.
The Make Corporate dialog box appears.
3. Fill in the Make Corporate dialog box and press OK.
The SavedGroupVersion is copied and stored as CorporateGroupVersion under the browser object <corporate groups>. *(This browser object is a child object of the object Corporate and can be unfolded or opened in the navigation area on corporate level.)*

Version - Make Current

Browser

Available on:

- Document Level

You use this menu entry to change the status of an Item Property Section from *snapshot* into *current*.

An Item Property Section with a status of *current* always refers to the current state of the item. If the item changes, this reference will change to.

With Version | Make Snapshot you can change the status *current* into *snapshot*.

Version - Make Fixed

In: Browser

To change the status of a File Section or File Property Section from *selected* into *fixed*:

1. Select a File Section or File Property Section with status *selected* at Document Level
A File Section or File Property Section with status fixed always refers to the same version of a file: to the version that was the selected version at the moment your selected the status fixed.
2. Select Version | Make Fixed.
The status has now changed from selected to fixed.

With Version | Make Selected you can change the status *fixed* of an object into status *selected*.

Version - Make Selected

In: Browser

To change the status of a File Section or File Property Section from *fixed* into *selected*:

1. Select a File Section or File Property Section with status *fixed* at Document Level
A File Section or File Property Section with status fixed refers to the file version that is currently selected in the system of origin. If another file version is selected in the system of origin, the Section is automatically rerouted to the newly selected file version.
2. Select **Version | Make Selected**.
The status has now changed from fixed to selected.

With **Version | Make Fixed** or **Version | Select Fixed...** you can change the status *selected* of an object into status *fixed*.

Version - Snapshot

In: [Browser](#)
[Versioning](#)

Use **Version | Make Snapshot** to change the status of an [Item Property Section](#) from *current* into *snapshot*.

Available on:

- [Document Level](#)

You use this menu entry

An Item Property Section with a status of *snapshot* always refers to the state the item was in at the moment you activated this menu entry. Item changes in the repository do not affect this reference; it is a snapshot.

With [Version | Make Current](#) you can change the status *snapshot* into *current*.

Version - New

Browser

You can make a new version of any object that does not have the status *background*. After a new version is created, this new version becomes the selectedobject.

Creating a new version of an object affects other objects that are dynamically linked to this object. They are automatically rerouted to this new version.

To make a new version of an object:

1. Select a browser object on any level except Corporate with a working or frozen status.
2. Select **Version | New**.
The new object is created.

Version - Restore

In: Browser

Available on:

- System level
- Implementation System level

When you do a restore on a saved group version all operations that were executed during the snapshot are reversed. This means that all objects in the group are unfrozen and the saved group version in the <saved groups> object is deleted.

Version | Restore... is only possible with saved group versions that are *not* promoted to corporate level.

Version - Select

Browser

Use this menu entry to select versions of objects. You don't have to select an object in the information area to use Version | Select..., as you can select every version of every object that is currently in the background.

To select a version:

1. Select **Version | Select** on any level below Project level.
The Select Version dialog box will appear.
2. Select the version you want to put back into the foreground and press OK.

Note :

You can also use Version | Select to put objects back in the foreground that have been removed from the browser with Version | Deselect.

Version - Select Fixed...

In: Browser

As opposed to Version | Make Fixed, you have a choice with **Version | Select Fixed...** which version you want to fix the reference to.

To change the status of a File Section or File Property Section from *selected* into *fixed*:

1. Select a File Section or File Property Section with status *selected* at Document Level
2. Select **Version | SelectFixed**.
The Select Fixed dialog box appears.
3. Select the Version for which you want to change the status, and press OK.
A File Section or File Property Section that is made fixed this way always refers to the same version of a file: to the version that was selected in the Select Version dialog box.

With Version | Make Selected you can change the status *fixed* of an object into status *selected*.

Version - Snapshot

Use Version | Snapshot... to make a snapshot of a group. The Snapshot dialog box appears. This option is available on:

- System level
- Implementation System level

You can only use this menu entry for GroupVersions.

When you make a snapshot of a group, you create a list containing all objects of the group with their versions. All objects of the group are then *frozen*.

When you use **Version | Snapshot...** for a group, a Saved Group is created.

Version - Unfreeze

Browser

Use this menu entry to unfreeze an object with status *frozen*. By unfreezing an object, you make it a *working* version again. If you unfreeze an object, its child objects are **not** automatically unfrozen.

To unfreeze an object:

1. Select a browser object on any level (except Corporate) with a *frozen* status.
2. Select **Version | Unfreeze**.
*The object status changes from **frozen** to **working**.*

Version - Version Browser, Starting

In: Browser

To start the Version Browser:

1. Select an object version below project level.
2. Select **Version | Version Browser**.
The Version Browser is started.

Version Browser - Menu Bar

Below, the default menus of the Version Browser are listed.

[File] [Edit] [View] [Options] [Version]

- **File** menu
 - Edit
 - Show
 - Info
 - Print View
 - Exit
- **Edit** menu
 - Paste
 - Select All
 - Invert Selection
- **View** menu
 - Refresh
 - Context Area
 - Toolbar
 - Message Area
- **Options** menu
 - Redraw
 - Zoom In
 - Zoom Out
 - Primary Font
 - Secondary Font
 - Color
 - Foreground
 - Background
 - Node
 - Merge Arrow
 - Primary Selection
 - Secondary Selection
 - Print Options
 - Printer Setup
- Version
 - Compare
 - Merge
 - Create Merge Link
 - Freeze
 - Unfreeze
 - New
 - Delete
 - Select
- **Help** menu
 - About ...

External File, External Link - browser objects

In: Browser

External Files are files that exist outside the Repository but are linked to it. Mainly they are found in the Implementation Phase, the Document Generator, and the Customization Editors.

Whether an External File is versionable depends on the way it is stored. An External file can be stored in the following ways:

Select from Document level: File | New | DocumentVersion, ...sections

Object : External FileVersion (Document and -sections)
Repository : Document , Document properties, Document Structure Matrix, Links to filenames in file system
File system: Data files for each section.

Select from System (in Implementation Phase): File | New | External File Version

Object : External File Version (Code Generation File)
Repository : Generated file, File properties, File System Path Part (link to file in file system)
File System: Data file

Select from System (in Implementation Phase): File | New | External Link

Object : External Links (used in Implementation Phase to link a file generated outside ObjectTeam to the OT Repository)
Repository : link to file in file system-file
File System: Data file

Select from <customization files> at any level: File | New | Customization File Version

Object : Customization File Version (from Corporate level and lower)
Repository : Customization file

Select from <user customization file> (at user level): File | New | External File

Object : External File
File System: Customization

Group - browser object

In: Browser

Subsections : [GroupVersion] [Saved Group] [Corporate Group]

A *Group* is a collection of diagram or text files and items:

- A *GroupVersion* defines a group
- A *Saved Group (Version)* is a snapshot of the contents of a group
- A *Corporate Group (Version)* is a *SavedGroup* that has been copied to the Corporate level.

GroupVersion

Groups can be created on System level and on Implementation System level. If your access rights allow you to, you can create a Group. You then specify the files and items you want to include in the Group by editing the group definition in the Edit Group Structure dialog box.

Group versions contain the definition of the group that you specify in the Edit Group Structure dialog box; they do not identify the specific file versions and item generations in the group.

SavedGroup

A Saved Group (Version) is a snapshot of a group.

Saved Groups Versions are listed in the pseudo object <saved groups>:

SavedGroup versions identify the specific file versions and item generations that were in the group at the moment the snapshot was taken. (When a snapshot is taken, all objects in the group are frozen.) This guarantees that the group, as it appeared at the moment of the snapshot, can be restored even if the group objects have evolved since then.

CorporateGroup

A Saved Group can be promoted to a Corporate Group.

Corporate Groups are stored on corporate level. You can include the contents of a Corporate Group in any system by selecting **Version | Activate**.

The Corporate Model consists of all the files and items in all the Corporate Groups. For these files and items, the indication *In Corporate* is set to **Yes**. When you add a Corporate Group to a system, their status is *reused*.

Corporate Groups are child objects of the browser object <corporate groups>:

Link status of Browser objects and UserRoleLinks

In: Browser

ObjectTeam has two types of link status:

Browser Object Link Status

The link status of the browser objects PhaseVersion, SystemVersion and <file>Version determine the behavior of their child objects when created.

UserRoleLink Status

The link status of a UserRoleLink determines whether or not a linked role is part of the user's initial effective context.

You can change both the object link status and the userrolelink status with **File | Change | Link Status**.

Merge Rules

In: [Browser](#), [Merge Window](#)

When you merge a source configuration, phase, or system version into a target source configuration, phase, or system version, ObjectTeam merges each source element into the target.

See the documentation for the rules that apply to each object merge.

Versionable Objects

In: Browser

The Version menu contains a number of menu entries that you can use for version management on certain browser objects.

You can use **File | Info** to find out if a browser object is versionable. The object is only versionable if the Info dialog box contains an entry **Version**.

Note that access to objects can be limited by the access rules on the objects.

Version Numbers

In: Browser

The first version of an object is always 1. The version number for any other object version is derived from the version number of the frozen version used to create it. Following are examples:

- First new version based on a frozen version, increment the last part of the frozen version number.
1 to 2
1.2 to 1.3
- Second new version based on the same frozen version, add .1 to the frozen version number.
1 to 1.1
1.2 to 1.2.1
- Third new version based on that same frozen version, add .0.1 to the frozen version number.
1 to 1.0.1
1.2 to 1.2.0.1

Version Status

All existing versions have a status. Three types exist:

- | | |
|-------------------|---|
| Working | This is the version that is editable. By default this is the latest version and the one that is visible. Of all versions of an object, only one can be working. |
| Frozen | Frozen versions cannot be changed, but are readable. |
| Background | These versions are not accessible from the repository; they have been moved to the file system. |

Version Status

In: [Browser](#)

All existing [versions](#) have a status. Three types exist:

- | | |
|-------------------|--|
| Working | This is the version that is editable. By default this is the latest version and the one that is visible. Of all versions of an object, only one can be working. |
| Frozen | Frozen versions cannot be changed, but are readable. |
| Background | These versions are not accessible from the repository; they have been moved to the file system. There is fourth status type if you have installed the Corporate Modeling Module : |
| Reused | This is the version that is copied from the pseudo object <corporate groups> into your system (see Using Corporate Groups). You cannot edit reused objects. |

Create a Group

In: Browser Working with Groups

Note . You must have installed the Corporate Modeling Module before you create a Group.

To create a group:

1. Move to System level.
2. Select File | New | Group Version.
The New Group Version dialog box appears.
3. Enter the group name.
4. Do you want to specify the contents of the group at this time?
 - If yes, select Edit.
The Edit Group Structure dialog box appears.
 - If no, select OK.
ObjectTeam creates an empty group. At some later time, use the Edit Group Structure dialog box to specify the contents of the group.

Merging Source Elements into Target

In: Merge Window

To merge source elements into the target object version:

1. Select one or more source elements in the Information area of the Merge window.
2. To select multiple elements, press and hold the Control key while selecting the elements, or use the **Edit | Select All** and **Edit | Invert Selection** commands.
3. Select one of the following commands from the Version menu:
 - Merge
 - Overwrite
 - Create Merge Link

ObjectTeam executes the command and removes the selected elements from the Information area of the Merge window.

4. Repeat steps 1 and 2 until all source elements have been merged into the target.

Note : Select Version | Merge when either

- the selected source element is a phase or system and you want to merge the source version into the target version.
If the source version of the phase or system contains an element that conflicts with the target, that element is not merged. It remains in the Merge window.
- the selected source element is in a System version (it is a file, property, or item), it does not conflict with the target, and you want to use the source version of the element.

Version Browser - Examining an Object

In: Version Browser

To examine an object version:

1. Select the object version
2. Select one of the following commands from the Version Browser Menubar:
 - **File | Info** to display the Info dialog box. Use this command to see which object versions have selected the current object version.
 - **File | Edit** (file versions only) to open an editor.

Corporate Modeling

In: Browser

Often a diagram or portion of a diagram can be reused by different projects. If you create a group that contains the relevant diagram components and related items, you can promote that group to the corporate level.

A corporate group can be accessed by other projects, but cannot be changed by any project.

The Corporate Modeling module must be active on the Browser levels where it is going to be used.

With this module you can create Groups, Saved Groups, and Corporate Groups.

The module adds the following objects and menus to your browser:

Objects :

- <saved groups>
- < corporate groups>

Menus :

- At system level: File | New | GroupVersion

When you select this menu-item, the Edit Group Structure dialog box appears in which you can create and edit the Group Structure.

You can check if a module is active on a particular Browser level by selecting Utilities | Show Active Modules on that level.

The module can be activated with the Modules Customization Editor

Merging Versions

In: [Browser, Merge Window](#)

Merging Versions allows for parallel development which occurs when a system must be modified along two or more paths simultaneously. The following gives an overview of the merging process

Start Merge

1. You select a target configuration, phase, or system.
2. You select a source configuration, phase, or system.
3. Start the merge.

Merge Rules

ObjectTeam examines all elements in the source in preparation for merging them into the target:

- If the source version of the element is already in the target, it does not need to be merged.
- If no version of the element is in the target, ObjectTeam can merge the element into the target.
- If the source and target elements are directly derived from one another, ObjectTeam can merge the element into the target.
- If the source and target elements are not directly derived from one another, the source element conflicts with the target element. You must resolve the conflict before ObjectTeam can merge the element into the target.

The Merge Window appears.

The Merge Window

Once the elements have been examined, the Merge Window is opened in which you can review the results. It lists each source element that must be merged into the target and whether or not the source element conflicts with a target element.

Merge source elements into the target

In the Merge Window you can merge the source elements into the target elements:

- If the source element does not conflict with the target element you can either merge the source element into the target or use the target element.
- If the source element conflicts with the target element, you either use the source element or use the target element. Typically you examine the conflicting elements, merge them by hand, and use your merged version of the element

Merge Tool Window

In: [Browser](#)

Title bar

The title bar lists the source and target object versions that you are merging.

Menu Bar

The [menu bar](#) contains the options you need to perform the merge process.

Display area

The Merge window lists all elements in the source object version that are not in the target object version. If the source version of an element is in the target, it is not listed in the Merge window.

If the Merge window is empty, the source object version is a subset of (or identical to) the target object version.

Like the Browser, the Merge window has a Navigation area and an Information area. When you select an object in the Navigation area, the Information area displays the following information for each element in that object:

Column	Description
Name	Object name.
Type	Object type.
Property Name	Property name.
Conflict	A value of No or Yes: <ul style="list-style-type: none">• No . The source and target elements do not conflict; ObjectTeam can merge the source element into the target. This occurs when the source element is not in the target, the source and target elements are directly derived from one another, or the source element is a System or Phase that contains no conflicting elements.• Yes . The source element conflicts with the target; you must resolve the conflict before ObjectTeam can merge the source element into the target. This occurs when the source and target elements are not directly derived from one another, or the source element is a System or Phase that contains conflicting elements.
Base Version	Version of Name from which both the From Version and the To Version were derived. Tip : If there is no Base Version, the From and To Versions are not derived from the same version.
From Version	Version of Name (if any) that is selected in the source version of the object selected in the Navigation area.

Version Browser

In: [Browser](#)

The Browser displays object versions in the context of the project hierarchy. Thus in the Browser you can examine all object versions that are currently selected in your project.

The Version Browser displays all versions of an object, regardless of whether or not they are currently selected in your project.

The results of the [Version Browser menu-items](#) are identical to those of the Browser Version Menu.

To examine an object version:

1. Select the object version
2. Select one of the following commands:
 - **File | Info** to display the Info dialog box. Use this command to see which object versions have selected the current object version.
 - **File | Edit** (file versions only) to open an editor.
- 3.

Version Management

Browser

Versions are the cornerstone of workgroup development. Versions allow simultaneously usage and development of the same objects. Development team **A** can use fixed versions of objects from team **B** while team **B** is refining working versions of the same objects.

The Version menu contains a number of menu entries that you can use for version management on versionable browser objects

More information:
Version Numbering
Version Status
Link status

Show All Merge Objects

To filter the information displayed in the Merge Window:

1. Select **View | Show All Merge Objects** to show all elements in the source object version that must be merged into the target object version.
2. Clear this command (the default) to show the minimum amount of information necessary to merge the source into the target:
3. If a phase or system in the source object version contains conflicting elements, show all elements in the phase or system.
4. If a phase or system in the source object version contains no conflicting elements, show the phase or system but not the elements that it contains.

Version - View Merge Link

In: [Version Browser](#)

To view Merge Links, use the Version Browser:

1. In the Browser, or Merge window, select an object version that has a Merge Link.
2. Select Version | Version Browser.
The Version Browser appears. It displays all versions of the object, including any Merge Links between versions.

Merge Links

In: [Browser](#), [Merge Window](#)

A Merge Link is stored on a target object version. It points from a source object version to the target object version.

When you merge the source version into the target version, ObjectTeam sees the Merge Link and ignores the source version. Merge Links are only used during merge.

Select the [Create Merge Link](#) command when you want to use the target version of the selected element rather than the source version.

The Create Merge Link command is only available when the source element is an object version. It is not available when the source element is a property or item.

Specifying the default text editor, dialog box

Text Editor Enter the name of the preferred Ascii editor here. Keep in mind that you define this editor only for the file type currently selected in the *Context* field. The default Ascii editor is defined in the M4_variable **M4_editor**.

Text Window Editor Switch this option *on* if the selected text editor creates its own window. If it doesn't, switch this button *off*.

Context Select the file type here for which the currently specified *Ascii Editor Command* must be used. Keep in mind that a separate editor has to be specified for every *Context*. You can select the following file types:

- none
- **DocTextSection** : Plain ASCII text

Specifying the default text editor

Use **Options | Editor** to specify the default ASCII editor used for editing text files such as:

- some customization files
- generated source code files

The Text Editor dialog box appears.

Change Name, dialog box

In: Browser, Diagram Editors

New Name Enter the new name of the object here.

Press the OK button to confirm the New Name of the object. Press the Cancel button to discard the operation. Note, in the case of Qualified diagrams, like Collaboration Diagrams, Sequence and State Transition Diagrams, you cannot alter the name of the Classifier.

Edit - Versionbrowser

In Version browser

This option opens an editor for a selected File Version (available for File Versions only).

To open an editor:

- Select a File Version.
- Select File | Edit.
The appropriate editor starts up.

Show - Versionbrowser

In [Version browser](#)

Select File | Show when you want to view a file. The appropriate editor is started and the file is shown in read only mode.

Version - Snapshot dialog box

Browser

With this dialog box, you can enter a comment that is stored with the saved group version (the group you made a snapshot of). You can view the comment entered here when you select File | Info for a saved group version.

Press the *OK* button to confirm the current comment or the *Cancel* button to discard the operation.

Edit <object>VersionDefaultFilter dialog box

(In: Browser) You can filter on one or more of the following characteristics. Only objects that satisfy the specified criteria appear in the Information area. An asterisk (*) indicates that no filter is set.

Name Enter a string here to filter on the *Name* of objects.
You can use wild cards.

Type Filters on the *Type* of objects.
The type of object(s) that can be filtered out is different for every browser level.

Version Enter a string here to filter on the configuration of the objects. The format of the string is:

<ConfigVersion_name>.<ConfigVersion_number>

You can use wild cards.

Status Enter one of the following strings to filter on the Status of objects:

- working
- frozen
- background

You can use wild cards.

Link Enter one of the following strings here to filter on the Link Status of objects:

- fixed
- dynamicFrozen

You can use wild cards.

In Corporate Enter one of the following strings here to filter on objects from the Corporate Model:

- Yes
- No

Copy Conflict - dialog box

In: Browser,

Note : This dialog box appears when you copy files from one configuration to another using Edit | Copy and Edit | Paste, and some of the file, cdm, and/or item names conflict with names in the system you are copying to.

Conflicting files

This box displays the conflicting files:

- **Conflicting diagram names:**
The conflicting files are displayed in a list box from which you can select either one or more files
- **Conflicting cdms and items:**
The conflicting files are displayed. It is not possible to single out cdms or items.

Buttons

Overwrite	The file to be copied overwrites the file in the target system.
Skip	Skips the file to be copied, and leaves the file in target system intact.
Cancel	The entire Copy/Paste procedure is cancelled.

New File/Item Filter dialog box

In: Browser

Use this dialog box to create a filter that selects files or items that you want to add to the group that is edited in the Edit Group Structure dialog box.

You can specify the following filter criteria:

- name of the object
- type of the item or file
- properties of the object:
 - name of the property
 - value of the property

You can Use Tcl glob-style pattern matching in all fields, except the property name field. Valid properties are defined in the property customization files.

Refer to your Tcl documentation for details on glob-style type of pattern

New External File

In: Browser

To create a new (non-versionable) External File, use **File | New | External File**.

Condition :

- The pseudo object <user customization files> must have been selected.

You can select a user customization file from the list box displayed here.

If you want to create a customization file that is not listed in the list box, enter its name in this field.

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the information area.

Edit Create the new object, leave the dialog box and start the **Edit <Object> Properties dialog box**

Cancel Do not create the object and leave the dialog box.

Select Merge Source

In: Browser

Phase

Select a phase.

Only phases that can be selected according to the merge rules are displayed here (see Merge From Other Phase).

Defined Item - browser object

In: [Browser](#)

Defined [items](#) can be found in the pseudo object <defined items> at system level.

Defined Items allow different graphical components to refer to the same semantic element. See [Item](#) for more information on items, components, defined and qualified items.

Security

In: Browser

Install the Security Module if you want to apply Access Control to your objects. This module adds the following objects and menus to your Browser: Objects

- Roles
- Users
- Userrolelinks

Menu_items :

- Security
 - Show Access Rights
 - Show Effective Context
 - Activate Role
 - Deactivate Role
 - Role Rights
 - Edit ...
 - Edit Controlled Classes...
 - Edit Controlled Lists...
 - Show
 - Show Controlled Classes
 - Show Controlled Lists
- The Change Link Status dialog box is adapted for editing userRoleLinks when a user or role is selected.

Activating a Role

(In: Browser)

Use **Security | Activate Role...** to add roles to the current list of effective roles.

There are two sets of effective roles:

- one for the current user on Corporate level
- one for the current user on Project level

If you activate a role on Corporate level, the role is added to your list of effective roles on Corporate level and applies to all levels from Corporate down.

If you activate a role on any browser level below Corporate level, then the role is added to your list of effective roles on Project level and affects all levels from Project down.

To be able to activate a role, it must not yet be included in your current list of effective roles, and must have a UserRoleLink to your user object.

To deactivate a role, removing it from the current list of effective roles, use Security | Deactivate Role.

Deactivating a Role

(In: Browser)

Use **Security | Deactivate Role...** to remove roles from the current list of effective roles. T

There are two lists of effective roles:

- one for the current user on Corporate level
- one for the current user on Project level

You can deactivate roles on any browser level. Roles deactivated on any level except Corporate level affect the effective roles on Project level. You cannot deactivate your default role (usually the role that has your user name).

To activate roles, adding them to the current list of effective roles, use Security | Activate Role.

Specifying Role rights for Controlled Class

(In: [Browser](#), [Security Module](#))

To specify the role rights for a controlled class

1. Move to Corporate or Project level, depending on which controlled class you are interested in.
2. Select **Security | Role Rights | Edit Controlled Classes...** .
The dialog box [Edit Controlled Class Role Rights](#) appears.
3. Select a [Controlled Class](#) from the Objects list.
4. Select a [Role](#) from the Roles list.
5. Specify the [role rights](#) of the controlled class in the Actions list..
6. Note : When you specify the role rights for a controlled class, ObjectTeam does not change the role rights for all objects of that type.

However , when evaluating [access rules](#) for an object, ObjectTeam checks the role rights of its controlled class.

Specifying Role rights for Controlled Class

In : Browser, Security Module

To specify the role rights for a controlled list:

1. Select **Security | Role Rights | Edit Controlled Lists...**
The dialog box Edit Controlled List Role Rights appears.
2. Select a Controlled List from the Objects list.
3. Select one or more roles from the Roles list.
4. Specify the role rights of the controlled list in the Actions list..

Editing Role Rights

(In: Browser)

Use **Security > Role Rights > Edit...** to allow or prohibit a particular role to carry out certain controlled actions on a particular *controlled object*.

Use **Security > Role Rights > Show...** to view the settings for these actions.

With Security > Show Access Rights... you can view the access rights for a particular object, i.e. the collection of Role Rights for your effective context.

A **role right** defines a role's access to all actions on an object.

An access rule is the access definition for a single action of an object for a certain role.

Displaying Your Access Rights

In: Browser

Your Access Rights are determined by a user's effective context for the browser object currently selected in the information area.

A user's **effective context** is determined by the user's effective roles. The user's **effective access rights** are determined by the Role Rights for all roles, both inside and outside the effective context.

As roles can be switched on and off at runtime, the effective context is dynamic. Checks for access are made at the moment of access.

To view your access rights:

1. Select the object,
2. select Security | Show Access Rights.
The Show Access Rights dialog box appears, displaying your access rights to the object based on your effective context.

Note : Your access rights are based on the ownRights of the controlled list. The childRights of the controlled list do not affect your access rights to the controlled list.

Specifying Role rights for Controlled Class

(In: Browser, Security Module)

To view the role rights of controlled classes:

1. Move to Corporate or Project level, depending on which controlled class you are interested in.
2. Select **Security | Role Rights | Show Controlled Classes...**
The dialog box Show Controlled Class Role Rights appears.
3. Select a Controlled Class from the Objects list
4. Select a Role from the Roles list
Under Actions the role rights of the selected controlled class are now displayed.

Specifying Role rights for Controlled Lists

(In: Browser, Security Module)

To view the role rights of controlled lists:

1. Select Security | Role Rights | Show Controlled Lists...
The dialog box Show Controlled List Role Rights appears.
2. Select a Controlled List from the Objects list
3. Select a Role from the Roles list
Under Actions the role rights of the selected controlled list are now displayed.

Showing and Editing Role Rights

(In: Browser)

Use **Security > Role Rights | Edit...** to allow or prohibit a particular role to carry out certain *controlled actions* on a particular *controlled object*. The Edit Role Rights dialog box appears.

Use **Security > Role Rights | Show...** to view the settings for these actions.

With Security | Show Access Rights... you can view the access rights for a particular object, i.e. the collection of Role Rights for your effective context.

A role right defines a role's access to all actions on an object.

An access rule is the access definition for a single action of an object for a certain role.

Showing Effective Roles

(In: Browser)

Use **File > Effective Roles...** to show the roles that are active for the current user at the current browser level.

The Show Effective Roles dialog box appears.

The two browser levels that have an effective context are:

- corporate level
- project level

To activate a role, use Security > Activate Role. To deactivate a role, use Security > Deactivate Role.

Show Access Rights

(In: Browser)

Effective context

Lists the currently activated roles.

Objects

The browser object for which the current access rights hold can be selected here (if applicable).

Actions

The actions that are not dimmed are applicable for the selected object. For each such action, the current setting (Allowed, Prohibited, Undefined) is displayed.

These settings can be changed using Security > Role Rights > Edit...

Activate Role dialog box

In: Browser, Security > Activate Role...

Roles Your inactive roles are shown in this list.
Select the role you want to activate.

Edit Role Rights Controlled Class dialog box

In: Browser

Objects

Select the controlled class for which you want to allow or prohibit an action

Roles

Select the role for which you want to allow or prohibit an action.

Role Right

Select the type of Role Right you want to define. Options are:

OwnRight

This is the Role Right for the object itself.

ChildRight

This is the initial Role Right of the child objects or for versions of the object.
This option is only available for controlled lists.

Actions

Change the setting of a certain action. The actions that are not grayed-out are the available actions.

For each action you can set a permission: Allowed, Prohibited, or Undefined.

Edit Role Rights Controlled Lists dialog box

In: Browser

Objects

Select the Controlled List for which you want to allow or prohibit an action

Roles

Select the role for which you want to allow or prohibit an action.

Role Right

Select the type of Role Right you want to define. Options are:

OwnRight

This is the Role Right for the object itself.

ChildRight

This is the initial Role Right of the child objects or for versions of the object.
This option is only available for controlled lists.

Actions

Change the setting of a certain action. The actions that are not grayed-out are the available actions.

For each action you can set a permission: Allowed, Prohibited, or Undefined.

Edit/Show Role Rights dialog box

In: Browser,

Objects

Select the object for which you want to allow or prohibit an action

Note . Only one object is available in this section: the object that you selected in the Browser before you selected Security | Edit... or Security | Show.

Roles

Select the role for which you want to allow or prohibit an action.

Role Right

Select the type of Role Right you want to define. Options are:

OwnRight

This is the Role Right for the object itself.

ChildRight

This is the initial Role Right of the child objects or for versions of the object.

This option is only available for controlled lists.

Actions

Change the setting of a certain action. The actions that are not grayed-out are the available actions.

For each action you can set a permission: Allowed, Prohibited, or Undefined.

Deactivate Role dialog box

In: Browser, Security | Activate Role...

Roles

This list shows your inactive roles.
Select the role you want to activate.

Showing Effective Roles

In: [Browser](#)

In this box you can see which of your roles are active.

Use Security | [Activate Role](#) or [Deactivate Role](#) to (de)activate roles.

Use Security | Show Access Rights to see how the [access rules](#) are set for the active role.

Actions (Security)

A role sets permissions for a set of actions belonging to a (controlled) object.

When actions are not available to the selected object, the childrights of the parent object do not allow them.

controlAction	Controls access to all other actions. If a role is allowed this action, it can change the permissions of other actions; otherwise, the role cannot change permissions of other actions.
createAction	Applies to controlled classes. By setting rights to this action you can control whether someone can make a particular controlled object. For example, if create for controlled class ConfigurationVersionList is set to denied, the user cannot make any new configuration versions.
destroyAction	Applies to controlled classes.
readAction	Applies to all object and controls whether a user can examine an object's contents.
modifyAction	Applies to all objects and controls a user's ability to change some property of an object, including its status. For example, without modify rights to a system version object, a user cannot make a new version of that system version or edit its properties.
insertAction	This applies to controlled lists and controls a user's rights to insert object in that list, that is, to make new versions of an object. For example without insert rights to a SystemVersionList, a user cannot add any objects to that list and make a new version of a particular system.
removeAction	Applies to controlled lists.
freezeAction	Applies to version of objects. It controls the ability to freeze objects, and thus change their status.
unfreezeAction	Applies to versions of objects.
modifyStatusAction	Applies to versions of objects. It controls the ability to change the status of objects. For example, you could prevent a user from changing the status of an object version from fixed to dynamic frozen.

Access Permissions

In: Browser, For each Action there are three kinds of Permission:

Allowed	Users that have the selected role in their effective context are allowed to perform the action on the selected object
Prohibited	Users that have the selected role in their effective context are not allowed to perform the action on the selected object
Undefined	There is no setting for this action for the selected role on the selected object. If the settings for all actions are set to <i>Undefined</i> , no access rule is entered in the database. As a result, no ownRights are visible for that object in the browser.

Access Rules

An access rule determines a role's access to a particular action for a particular object. The access rule settings are: **Allowed**, **Prohibited**, or **Undefined**. Generally speaking, if no role has an **Allowed** setting for an action, **Undefined** is interpreted as **Allowed**. If, however, a role has an **Allowed** setting for an action, and that role is not in the user's effective context (the list of currently active roles), **Undefined** is interpreted as **Prohibited**.

Below are the formal rules and an equivalent commented table.

Rules

The following rules determine the user's access to operations on objects:

- If the superuser role is in the effective context of the user, access is allowed.
- If the access rule is undefined for all roles, access is allowed.

If the rules above do not apply, class action access rules are checked:

- If the effective context contains a prohibiting access rule for the class of the object, access is denied.
- If the effective context contains an allowing access rule for the class of the object, access is granted.
- If outside the effective context an allowing access rule exists for the class of the object, access is denied.

If none of the above rules apply, object action access rules are checked:

- If the effective context contains a prohibiting access rule for the object, access is denied.
- If the effective context contains an allowing access rule for the object, access is granted.
- If outside the effective context an allowing access rule exists for the object, access is denied.

If none of the above rules apply, access is granted.

Table

This commented table holds the same information as the rules above, only the presentation is different. First the terms of the table are explained and how the access definition is determined. Then the three checking rules are executed.

- access is checked per action on an object for all roles.
- inside effective context is the result of all active roles. More occurrences of the same setting count as one:

```
allowed + undefined = allowed
allowed + prohibited = prohibited
undefined + prohibited = prohibited
allowed + undefined + prohibited = prohibited
```

- outside effective context is the result of all other roles. More occurrences of the same setting count as one:

```
allowed + undefined = allowed
allowed + prohibited = allowed
undefined + prohibited = prohibited
```

allowed + undefined + prohibited = allowed

Checking rules:

1. If the superuser role is active, access is granted.
2. If not, the table is applied to the action of the class of the object (controlled class).
3. If the action is allowed for the class, the table is applied to the action of the object.

access definition inside effective context	access definition outside effective context	resulting access
undefined	undefined, prohibited	allowed
undefined	allowed	prohibited
allowed	don't care	allowed
prohibited	don't care	prohibited

Role Rights and Access Rights

Note: controlled object in the explanation below includes Browser object, Controlled Lists and Controlled Classes.

The Role Rights of a Controlled Object are the Access rights of one or more roles for a selected Controlled Object.

You can distinguish Role Rights from Access Rights as follows:

- Access Rights = The set of Actions + Permissions for a controlled object
- Role Rights = Access Rights of one or more roles

Use the Utilities | Role Rights menu to edit and show role rights.

Use the Utilities | Show Access Rights menu to edit and show role rights.

Controlled Object

A controlled object is an object for which Access Rights have been specified.

To edit or view role rights for a controlled object:

- Select an object.
- Select Security | Role Rights | Edit... or Show.

Role Rights (menu)

The Utilities | Role Rights menu contains all the options to edit and show your Role Rights:

Edit ... / Show A dialog box appears containing the role rights for the selected object.

Edit Controlled Lists ... / Show Controlled Lists

A dialog box appears containing the role rights for all existing Controlled Lists in the repository.

Edit Controlled Classes ... / Show Controlled Classes

A dialog box appears containing the role rights for the Controlled Classes. Note: This option is only available at Corporate and Project level.

Browser Tool Bar

In: Browser

The *tool bar* is a window area that contains icons for quick activation of frequently used menu entries. Instead of executing such a menu entry from the menu bar, you can just click on the corresponding icon in the tool bar.

The tool bar is located between the menu bar and the context area. You can find out which menu entry a toolbar button represents by dragging your mouse pointer over the button in the toolbar and keeping it there for a few seconds. A ToolTip with the name of the menu entry appears.

The tool bar in the browser contains the following buttons by default:

- File | Effective Roles...
- File | Info...
- File | Print
- Security | Access Rights...
- (Version | Snapshot)

New Document Structure Matrix

In: Browser

File | New | Document Structure Matrix

Conditions :

- The browser must be on document level.
- Only one Document Structure Matrix per document level can exist.

Security: Setting up Access Control

In: [Browser](#) The option Security | Setup Access allows you to automatically install the Access Control for new projects and configurations.

To prepare for running Setup Access:

1. Edit the customization file roles (See the Project Management Guide.).
This file defines which roles are have which permissions for which actions.
2. Edit the customization file userroles (see the Project Management Guide).
This file defines which users can play which roles.

To run Setup Access :

(verify that the Security model is installed and you have a license)

1. Verify that SuperUser is one of your effective roles.
Tip : Use Security | Show Effective Roles to see your effective roles. Use Security | Activate Role to make the SuperUser role part of your set of effective roles, you are allowed to do so.
2. Create a customization file roles and edit it.
See the ObjectTeam Customization Guide for details on how to create a customization file. The customization file roles on page 534 has further details on the structure of the file roles.
3. Create a customization file userroles and edit it.
See The customization file userroles on page 536 for details on the structure of this file,
4. Make sure that the PhaseVersions mentioned in the roles file exist in the configuration version for which you are setting up the access control.
5. Make sure that these phase versions are empty.
Automatic access control will not work on objects that have already been created in these phase versions.
6. In the ObjectTeam Browser, go to the configuration level.
7. Select Security | Setup Access.
An Execution Window starts up, showing the execution of all necessary commands.
8. Restart the ObjectTeam Browser.
The access control rules as defined in the customization files roles and userroles are now in effect.

Security: Overview of Access Control

In: [Browser](#) (see also: [Modules](#) on how to install the Security Module.)

Access Control consists of the following processes and subjects:

Access Rights This is the set of Role Rights for one Controlled Object/List/Class within the effective context of the user. You can view the Access Rights with [Utilities | Show Access Rights](#).

Controlled Objects

Controlled means that these objects are subject to Access Control. See the leaf objects of the class Controlled in the Repository structure in the Repository Interface Guide for an overview of the classes that are leaf objects of Controlled Object, Controlled List, and Controlled Class.

Controlled Class

A Controlled class is a class of controlled objects. This is used to define which actions are supported as controlled actions and the related access rights are.

Controlled Object

A Controlled object is an instance of a Controlled Class.

Controlled List

A Controlled list represents the child objects of a Controlled Object, i.e. the Controlled list of a Configuration contains the Access Rules for the new Phases and customization files that are created within that Configuration.

Effective context

The effective context of a user is defined as:

A User (defined in the pseudo object <user>) linked to an active Role (defined in the pseudo object <role>) through a UserRoleLink.

Access Rules Access Rules are the combination of Actions and Permissions set for a **Controlled Object/List/Class**

Role Rights Role rights are the Access Rules that apply to a **Role** (activated role)

ChildRight and OwnRight

OwnRight represents a set of Access Rules that are owned by a Controlled Object/List/Class and that can be inherited as ChildRight by a Controlled Object/List/Class that is derived from it.

Document Properties

In Browser, Edit Properties dialog box

You can set or view the values of the following properties for the browser object DocumentVersion. Except for the property *Document Structure File*, the values of these properties are included in the title page that is part of the default document generation:

- Document Authors
- Document Date
- Document Keywords
- Document Reference
- Document Status
- Document Subject
- Document Structure File
This property can be used to specify another Document Structure file than the default ones on corporate level. Whatever you have entered as Document Structure File property will appear as selection option in the Generate Document dialog box.
- Document Title
- Document Type

Local section Properties

In [Browser, Edit Properties dialog box](#) When you open the Edit Properties dialog box for a Local Section, you can specify two generators on the Misc Page:

Structure Generator

Through this property you can enter a Tcl procedure that calls a structure generator. The structure generator of a local section that must contain all diagrams of a certain type is for instance:

```
createFileSections <fileTypes>
```

This generator refers to the Tcl procedure of the same name, which is defined in the Tcl file `docprocs.tcl`. You can find this file in the directory `<M4_home>/tcl`. In this file you can also find structure generators for [property sections](#).

Contents Generator

Through this property you can enter a Tcl procedure that calls a contents generator. A contents generator is a Tcl procedure that generates the contents of local sections by calling methods from classes defined in Object Tcl. These classes are associated with a particular local section type. The internal contents generator used depends on:

- The specified section type (`sect_type`)
- The current word processor or DTP package (**Fm, II** or **Word**)

If you want to specify a contents generator that is different from the default, you can enter a Tcl procedure here that calls this (alternative) contents generator.

For details on Contents and Structure Generators, refer to the *ObjectTeam Document Generation Guide*.

FileSystemPath Property

In Browser, Edit Properties dialog box

Use this property to customize your **user environment**. The user environment is a directory tree in your file system, into which generated source files and Document files are copied.

The user environment is a concatenation of the property values of the following browser objects:

Browser object	default value
<u>Corporate</u>	<Corporate_name>
<u>Project</u>	<Project_name>
<u>ConfigVersion</u>	<ConfigVersion_name>_<ConfigVersion_number>
<u>PhaseVersion</u>	<PhaseVersion_name>
<u>SystemVersion</u>	<SystemVersion_name>
<u>FileVersion</u>	<FileVersion_name>

So by default, the user environment is setup as follows:

```
<user_home>/<Corporate_name>/<Project_name>/  
<ConfigVersion_name> <ConfigVersion_number>/  
<PhaseVersion_name>/<SystemVersion_name>/
```

If you want your user environment to be created elsewhere in the file system, you have to set the property **File System Path Part** for one of the browser objects mentioned above. For instance, if you set it to `/home/test` for *Configuration*, the resulting user environment is:

```
/home/test/Implementation/systemtest/
```

In this example, `systemtest` is the name of the System.

Editing Customization files

In: Browser, Edit

Edit - menu entry (project, configuration and phase level)

Use **File | Edit** on

- Project level
- Configuration level
- Phase level

to edit customization files. Depending on the type of customization file you select in the information area, the customization editor or the default ASCII text editor is started.

Editing FileVersion, External Link, Group, and Customization file at system level

In: Browser, Edit

You can use **File | Edit** only for objects selected in the information area.

Edit - menu entry (system level - in Analysis, System Design and Object Design)

Use **File | Edit** to open objects on system level with the following type:

FileVersion The appropriate diagram editor is started in a separate window.

External Link The Ascii editor is started in an Execution window in read-only mode.

Group The Edit Group Structure dialog box is started.

Customization file
Depending on the type of customization file currently selected, the customization editor or the default ASCII text editor is started.

Edit - menu entry (system level - implementation)

Use **File | Edit** to open objects on System level (implementation) with the following type:

FileVersion The Ascii editor is started in a Execution window.

External Link The Ascii editor is started in an Execution window, in read-only mode.

Group The Edit Group dialog box is started.

Customization file
Depending on the type of customization file currently selected, the customization editor or the default ASCII text editor is started.

Editing documents and document structure matrix

In: Browser, Edit

Use **File | Edit** to open objects on Document level with the following type:

dsm The Edit Document Structure Matrix dialog box appears, in which you can change the document structure.

Customization file

Depending on the type of customization file currently selected, the customization editor or the default ASCII text editor is started.

Open Source

In: Merge Window

To display information about an element:

1. Select a FileVersion in the information area of the Merge Window.
2. Select **File | Open Source**.

This opens an editor for the source version of the element.

Notes

1. When examining conflicting file versions, you might find it useful to compare them.
2. You can also use the following menu items to display information about objects
 - File | Info
 - File | Open Target

Open Target

In: Merge Window

To display information about an element:

1. Select a FileVersion in the information area of the Merge Window.
2. Select **File | Open Target**.

This opens an editor for the target version of the element.

Notes

1. When examining conflicting file versions, you might find it useful to compare them.
2. You can also use the following menu items to display information about objects
 - File | Open Source
 - File | Info

\

Version - Freeze Version dialog box

In: Browser

Comments

You can enter a one line comment in the **Freeze Version dialog box**. This comment is stored with the frozen version and can be used to identify versions more easily.

\

Version - Select Fixed dialog box

In: Browser, Version | Select Fixed...

File <filename> Select the Version for which you want to change the status, and press OK.
A File Section or File Property Section that is made fixed this way always refers to the same version of a file: to the version that was selected in the Select Version dialog box.

Info (merge window)

In: Merge Window

To display information about an element:

1. Select an object, item or property in the information area of the Merge Window.
2. Select **File | Info**.
This displays a dialog box that shows
 - For an object version, where the source and target versions of the element are selected.
 - For an item, the scope of the source and the target element.
 - For a property, the value of the source and the target element.

Notes

1. When examining conflicting file versions, you might find it useful to compare them.
2. You can also use the following menu items to display information about objects
 - File | Open Source
 - File | Open Target

Select Source (From) Object to Merge, dialog box

In: Browser, Version Browser

List Displays a list of ConfigVersions from which you can select the source object to merge with.

OK . The Merge Window appears.

Cancel

Compare Command

In : Browser

Use **Options | Compare Command...** to change the default command used for comparing ASCII files during Utilities | Compare With Previous Phase and Version | Compare.

The Compare Command dialog box appears.

Compare Command dialog box

In: [Browser](#)

Command Use one of the following commands:

Compare command for text files and local sections: fcomp

By default, ObjectTeam uses the **fcomp** utility to compare text files and local sections.

Compare command for diagrams and matrixes: udmcmp

ObjectTeam uses the **udmcmp** utility to compare diagrams and matrices. You can customize the **udmcmp** utility, as described in the ObjectTeam Customization Guide.

By default, this utility reports diagrams or matrices as identical if the elements in the following table are identical:

Elements	Diagram	Matrix
Nodes	X	
Connectors	X	
Node connectors	X	
Rows		X
Cells		X

By default, the udmcmp utility finds nodes, connectors, connected nodes, and rows to be identical if the elements in the following table are identical:

Elements	Diagram component	Matrix components
Component type	X	X
Labels	X	X
Component attributes	X	X
Coordinates	X	
Size	X	X

Comparing a Phase with the previous one

(In: [Browser](#))

Use **Utilities | Compare With Previous Phase** to compare (elements of) a [system](#) with (elements of) a system in a previous [phase](#). It produces a report of the differences between the current system and the one in the previous phase.

To compare a phase with the previous one:

1. Select **Utilities | Compare With Previous Phase**.
2. The [Compare With Previous Phase dialog box](#) will appear.

You can compare (selected) systems on [Phase level](#) and (selected) elements of systems on [System level](#).

By default, **Compare With Previous Phase** carries out the following three actions (you can customize these actions by editing the [customization file](#) *compare_rules*):

- Comparisons to check the existence of [diagrams](#)
- Comparisons to check the contents of diagrams
- Comparisons to check the [properties](#) of components

You can change the default command used for comparing phases with the option [Options | Compare Command](#).

Merge Tool - Menubar

In: Browser

Below , the default menus of the Merge Tool are listed.

[File] [Edit] [View] [Options] [Version]

- **File** menu
 - Reload
 - Open Source
 - Open Target
 - Info
 - Exit
- **Edit** menu
 - Select All
 - Invert Selection
- **View** menu
 - Refresh
 - Show All Merge Objects
 - Context Area
 - Toolbar
 - Message Area
- **Options** menu
 - Font ...
 - Compare Command...

Version

- Merge
 - Overwrite
 - Import
 - Create Merge Link
 - Compare
 - Version Browser
- **Help** menu
 - About ...

Merge Tool (Merging Systems)

The Merge tool appears when you have selected

- a target Phase in the browser
- Utilities | Merge From Other Phase... and a source Phase in the Select Import Source dialog box, or
- Utilities | Merge From Previous Phase...

The Merge Tool Window shows the source phase with its objects from which you can select the objects you want to copy to the selected target phase.

When you merge systems into another phase, ObjectTeam actually carries out three tasks:

- It copies system data from one phase to the other
- It freezes the source system (needed to create a merge link)
- It creates merge links

You use the ObjectTeam Merge Window to merge data from one phase into the next. In the Merge Window, you select which system(s) you want to merge.

See also Merge Links and Merge Rules.

Compare Command dialog box

In: Browser and Version Browser

file Select the version of the object you want to the compare the active object to.

Merge From Other/Previous Phase

In: [Merge Systems](#)

You can merge system data from a previous phase, using Utilities | Merge From Previous Phase.

You can merge system data from a Phase other than the previous, using Utilities | Merge From Other Phase. You select the source Phase from the [Select Merge Source dialog box](#), after which the [Merge Window](#) appears.

To merge systems from another or the previous phase:

1. In the ObjectTeam Browser, move to Phase into which you want to merge system data from the previous phase.
2. Select **Utilities | Merge From Previous Phase**
or Select **Utilities | Merge From Other Phase**
The [Merge from Other Phase dialog box](#) appears. Select another Phase from this dialog box
The [Merge Window](#) appears.
If the Merge Window is empty, there is nothing to merge: the source object(s) are then identical to the target object(s).
3. In the Information Area of the ObjectTeam Merge Window, select the system(s) you want to merge.
If you want to preserve any changes made in a system in the current phase, do only select non-conflicting systems. A system is non-conflicting if it has a No in the Conflict field.
See Dealing with Merge Conflicts in the Project Management Guide for details.
4. Select **Version | Merge**
The data from the source system(s) is copied to the target phase, the source system(s) are frozen, and merge links are created.

Invert (selection)- Versionbrowser, Merge Tool and Import Tool

In Version browser, Merge Tool and Import Tool

Selecting **Edit | Invert** from the menubar in the Version Browser will invert the selection of versions.

Selecting **Edit | Invert Selection** from the menubar in the Import or Merge Tool will invert the selection of objects in the information area.

Reload (Merge Tool)

In Version browser

Use **File | Reload** to load the original data to the Merge Tool.

If changes were made, the Reload dialog box appears, requesting confirmation of the *Reload* action.

Invert - Versionbrowser

In [Version browser](#)

Select Edit | Invert in the Version Browser when you want to invert your selection of versions.

Merge (systems) Tool Window

In: Browser

Title bar

The title bar lists the source and target phases

Display area

The Information Area lists the contents of the object selected in the Navigation Area. It contains the following columns:

Name Object name.

Type Object type.

Property Name Property name (only applicable for item and file properties)

Conflict A value of No or Yes (see Dealing with Merge Conflicts on page 227)

Base Version Version of Name from which both the From Version and the To Version were derived.

Tip : If there is no Base Version, the From and To Versions are not derived from the same version.

From Version Version of Name (if any) that is selected in the source version of the object selected in the Navigation area.

Tip : If the Base Version and From Version are the same (or from the same configuration), and the To Version is different (or from a different configuration), the To Version is the most recent version.

To Version Version of Name (if any) that is selected in the target version of the object selected in the Navigation area.

Smalltalk Properties

In Browser, Edit Properties dialog box

You can set the following properties for a system in Smalltalk:

- Default Smalltalk Class Category Name.
In the list box you can select either DiagramName or SystemName as Default Smalltalk Class Category Name.
- Smalltalk Generate Print Methods.
Switch the checkbutton on to enable this option.

Creating a Corporate Group

In: Browser, Working with Groups

Note . You must have created a Saved Group before you can create a Corporate Group.

To create a corporate group:

1. Select the SavedGroupVersion in a <saved groups> pseudo object.
2. Select Version | Make Corporate....
The Make Corporate dialog box appears.
3. Fill in this dialog box and press OK.
The SavedGroupVersion is promoted to Corporate Group.
You can view the CorporateGroupVersion in the <corporate groups> pseudo object.

Creating a Saved Group

In: Browser, Working with Groups

Note . You must have created a Group before you can create a Saved Group.

To create a saved group:

1. On system level, select a group in the information area. Select Version | Snapshot.
2. The Snapshot dialog box appears in which you can enter a comment, such as the reason the saved group was created. This comment appears in the Info dialog box.
3. Enter a comment, and then click OK.
In the <saved groups> pseudo object, ObjectTeam creates a saved group version with the same name as the group version and the contents of the group version are frozen.

Working with (Saved) (Corporate) Groups

In: Browser Note: You must have installed the Corporate Modeling Module before you can work with groups.

Working with groups involves the following steps:

1. Create a Group
2. Create a SavedGroup
3. Create a CorporateGroup
4. Copy the Corporate into your system

Create (Corporate) Group

In: Browser

The process of create a corporate groups involves three steps:

1. Create a Group
2. Create a SavedGroup
3. Create a CorporateGroup

To create a group:

1. Move to System level.
2. Select File | New | Group Version.
The New Group Version dialog box appears.
3. Enter the group name.
4. Do you want to specify the contents of the group at this time?
 - If yes, select Edit.
The Edit Group Structure dialog box appears.
 - If no, select OK.
ObjectTeam creates an empty group. At some later time, use the Edit Group Structure dialog box to specify the contents of the group.

To create a saved group:

1. On the system level, select a group in the information area. Select Version | Snapshot.
2. The Snapshot dialog box appears in which you can enter a comment, such as the reason the saved group was created. This comment appears in the Info dialog box.
3. Enter a comment, and then click OK.
In the <saved groups> pseudo object, ObjectTeam creates a saved group version with the same name as the group version and the contents of the group version are frozen.

To create a corporate group:

- 1.
- 2.

Version - Create Merge Link

In: [Version Browser](#)

To delete a Merge Link in the Version Browser:

1. In the Version Browser, display the object version that has the Merge Link.
2. Select the source version of the Merge Link.
3. Select the target version of the Merge Link.
4. Select **Version | Delete Merge Link**.
ObjectTeam removes the Merge Link that points from the source object version to the target object version .

Version - Create Merge Link

In: [Version Browser](#)

To create a Merge Link in the Version Browser:

1. In the Version Browser, display the object for which you want to create a Merge Link.
2. Select the source object version.
3. Select the target object version.
4. Select **Version | Create Merge Link**.
ObjectTeam adds to the target object version a Merge Link that points from the source version to the target version.

New Razor File

In: Browser

File | New | Razor File...

This option is available at System Level in the Implementation Phase when:

- The Razor integration module has been installed.
- The user environment is configured properly.
- The required Razor folders exist.

To create a new Razor File:

1. Select File | New | Razor File..
The New Razor File dialog box appears.
2. Select the ObjectTeam file type that maps to the new Razor file type, enter a file name, and click OK.
3. If the file is created successfully, select it in the Information Area.
4. Select File | Info.
The Info dialog box appears. The Razor Type in this box indicates the new Razor file type.

New Continuous File

In: Browser

To create a Continuous file:

1. Select **File | New | Continuous File...**
The New Continuous File dialog box appears.
2. Fill in the **New Continuous File dialog box** and click OK

Type Select a file type from the Type list

The set of available file types depends on your target language. The target language is set through the M4 variable M4_target_lang.

These ObjectTeam file types are mapped to Continuous file types through an ObjectTeam customization file (see Remapping File Types on page 111).

Name Enter a name for the new file in the Name field.

Comment Specify a Comment, if desired.

Type Specify a Task number, if desired.

All file version associated with a particular task can later be checked in all at once (see How to check in a Continuous task on page 214).

If you specify an invalid task number the file will not be created.

The new file is created in the Continuous database and added to the Information area of the ObjectTeam Browser.

If the selected file type is mapped to a file type that is unknown by Continuous, the new file will not be created.

New ClearCase File

In: Browser

To create a ClearCase file:

1. Select **File | New | ClearCase File...**

The New ClearCase File dialog box appears.

2. Fill in the **New ClearCase File dialog box**:

Type Select a file type from the Type list

The set of available file types depends on your target language. The target language is set through the M4 variable M4_target_lang.

These ObjectTeam file types are mapped to ClearCase types through an ObjectTeam customization file (see Customizing File Type Mapping on page 36).

Name Enter a name for the new file in the Name field.

Creation Comments

Entering Creation Comments is optional.

3. Click OK.

After a successful creation, an information box appears showing you the actions that were performed.

4. Click OK to close the Info box.

The new file is created in the appropriate ClearCase VOB and added to the Information area of the ObjectTeam Browser.

New External File Version

In: Browser Use **File | New | External File Version** to create a new External File Versions. On this level these are source files which are stored as versionable File Versions in the Repository and written to the <user_environment/system-name> directory,

Conditions :

- The browser must be on implementation system level.
- The status of the current object must be *working*.

New Model

In: Browser

File | New | Model...

Conditions :

- The browser must be on corporate level.

You specify the new model in the New Model dialog box.

Merge From Other Phase dialog box

In: Merge Tool

In this dialog box you can select the source Phase from which you want to Merge.

Note . The Implementation Phase will not be displayed here. Use Generate <language> to import systems into the Implementation Phase.

Closing a class

In: Class Browser

Use **File | Close** to close the most recently opened class, returning to the previously opened class. For example, if you open Class01 and then Class02 from anywhere in the display area, closing Class02 returns you to Class01.

If you select File | Close, and only one class or no class is open, ObjectTeam displays a warning message asking if you would like to exit from the Class Browser.

Opening a Class Diagram

In: Class Browser

Use **Utilities | Edit Class Diagram** to start the Class Diagram Editor for the currently open class. If the opened class appears in several CDs, a dialog box appears allowing you to select the desired diagram.

Finding a Class

In: [Class Browser](#)

Use **File | Find Class...** to open a class that matches a search pattern. You specify the pattern in the [Find Class dialog box](#).

Finding a Class Feature

In: [Class Browser](#)

Use **File | Find Feature...** to open a class that contains features (attributes or operations) that match a search pattern. You specify the pattern in the [Find Feature dialog box](#).

Switching between Flat view and Non-Flat view

In: Class Browser

Use **View | Flat** to switch between a display area that contains:

- only features of the opened class
(the flat view is **off**)
- features of the opened class itself and its *inherited* features
(the flat view is **on**)

If the opened class has no superclasses, only its own features are displayed when flat view is **on**.

Filtering Class Features

In: [Class Browser](#)

Use **View | Filter Features...** to narrow down the [operations](#) and [attributes](#) in the [display area](#) of the class browser.

You can change the current filter settings in the [Filter Features dialog box](#).

Opening a Class

In : Class Browser)

Use **File | Open** to display information about classes in the Class Browser.

You can open classes from any field in the display area, provided that the selected object contains a reference to a class. When you open a class, ObjectTeam updates the information in all fields accordingly.

Printing the Display Area of the Class Browser

In: [Class Browser](#)

Use **File | Print View** to make a print of the information currently available in the [context area](#) and the [display area](#) of the class browser.

To change the print command that **Print View** uses, select [Options | Printer Setup | Text](#).

The output looks like this:

```
+-----+
|View:
| Project:          docu_proj
| Configuration Version: docu_conf.1
| Phase Version:   ObjectDesign.1
| System Version:  docu_sys.1
| Class:           Account
|
|
| -- Superclasses --
|   ...
|
| -- Subclasses  --
|   ...
|
| -- Features --
|   ...
|
| -- Associations --
|   ...
|
| :   ...
```

- Superclasses:** The [class\(es\)](#) the opened class is derived from.
- Subclasses :** The class(es) that are derived from the opened class.
- Features :** The [data attributes](#) and [operations](#) of the opened class.
- Associations :** The [associations](#) in which the opened class participates.

Reloading Classes

In : Class Browser

Use **File | Reload Classes** to update the information in the display area of the Class Browser.

This is especially useful after you make any changes to the Class Diagrams involved.

Showing the scope of an item

In: Class Browser

Use **Item | Show Scope...** to obtain scope information about the item currently selected in one of the class browser list boxes.

The scope information is displayed in the Show scope dialog box.

Sorting information in the Display Area

In: [Class Browser](#)

Entries in the [display area](#) are sorted alphabetically. You can choose to make the sorting case sensitive by using **Options | Sort Case Sensitive**.

Menu Bar - Class Browser

In: [Class Browser](#)

Below, the default menus of the [class browser](#) are listed. You can add new menus and redefine existing ones by using the [Customization Editor](#).

[[File](#)] [[Edit](#)] [[Item](#)] [[View](#)] [[Options](#)] [[Utilities](#)] [[Help](#)]

- **File** menu
 - [Reload Classes](#)
 - [Open](#)
 - [Close](#)
 - [Find Class...](#)
 - [Find Feature...](#)
 - [Print View](#)
 - [Exit](#)
- **Edit** menu
 - Copy
- **Item** menu
 - [Show Properties](#)
 - [Show Scope](#)
- **View** menu
 - [Toolbar](#)
 - [Context Area](#)
 - Message Area
 - [Flat](#)
 - [Filter Features...](#)
- **Options** menu
 - [Sort Case Sensitive](#)
 - [Font ...](#)
 - [Printer Setup...](#)
- **Utilities** menu
 - [Edit Class Diagram](#)
- **Help** menu
 - [What 's This?](#)
 - [On Help...](#)
 - [Help Topics](#)
 - About Class Browser

Class Browser

Use the Class Browser tool to display the following information about classes that are in the current Phase and appear in a CD:

- Features (attributes and operations)
- Relations (generalizations and associations)

Besides getting an overview of classes, you can also use the class browser to search for classes within a PhaseVersion.

You can start the class browser by selecting **Utilities | Class Browser** from the following locations:

- Browser on phase level
- Browser on system level
- Any diagram editor

Window sections

The default class browser window contains the following areas. You can use the *View* menu to hide and show the Message Area, Context Area, and Tool Bar.

- **Menu Bar** - You select menu entries from here.
- **Tool Bar** - You can select frequently used menu entries from here.
- **Context Area** - Displays information about the current context.
- **Display Area** - Contains the list boxes that display class information.
- **Message Area** - Displays system messages. On Unix, you can browse through the message history with the arrow-up and arrow-down symbols at the right.

Context Area

In: Class Browser

The *Context Area* in the class browser is located below the tool bar and above the display area.

It shows the following information about the context (if applicable):

<u>Project</u>	<u>SystemVersion</u>
<u>Configuration Version</u>	<u>Class</u>
<u>PhaseVersion</u>	

Display Area

In: Class Browser

The *display area* of the Class Browser displays information about classes of the current phase. This information is presented in a number of list boxes:

- Superclasses
- Classes
- Subclasses
- Features
- Associations

Superclasses If the currently open class is a subclass in a generalization, its superclasses are listed here.
Classes This list box contains the class names of all classes that are known in the current PhaseVersion and that appear in a CD.
Subclasses If the currently open class is a superclass in a generalization, its subclasses are listed here.
Features This list box contains the attributes and operations of the currently open class.

Attributes are displayed as follows:

`<attribute name>:<data type>`

Operations are displayed as follows:

`<operation name>(<parameter list>):<operation type>`

`<parameter list>` is a sequence of comma separated parameters, each having the following format:

`<parameter name>:<data type>`

Associations This list box contains the binary and n-ary associations in which the currently open class participates. They are displayed as follows:

binary association:

`[qualifier]---association_name--<role>---class_name`

n-ary association:

`---association_name--<role>---class_name, ---<role>---class_name ...`

Filter Class Features dialog box

In: Class Browser

You can change the current filter settings in the *Filter Features* dialog box:

Filter Case Sensitive

Switch this button on only if you want the *Values* of the *Filter Elements* to be considered case sensitive.

Select Filter

Choose a class feature here: Attributes or Operations

Filter Enabled

Switch this button on to enable the filter settings in the list box

List box

Filter Element	Enabled	Value
-----	-----	-----

This list box shows the current settings for the filter elements belonging to the class feature selected in the field *Select Filter*. These settings can be changed by pressing the *Set Filter Element* button.

Set Filter Element

Press this button to change the current setting for a Filter Element in the *List box*.

Enabled

Switch this button on to enable the filter setting for the selected Filter Element

Value

Enter a filter value here. You can use glob-style pattern matching. For details on this type of pattern matching, refer to your Tcl documentation.

Buttons :

OK Use this button to confirm the information in the Filter Features dialog box.

Apply Use this button to make the filter effective in the class browser. The Filter Feature dialog box is not discarded.

Reset Use this button to cancel the changes made to the Filter Feature settings.

Cancel Use this button to discard the operation.

Find Class dialog box

In: Class Browser

You specify the pattern in the **Find Class dialog box**:

Class Name Here you can specify a glob-style search pattern. For details on this type of pattern matching, refer to *Tcl and the Tk Toolkit* by John K. Ousterhout.

Case Sensitive Switch this button on if the case of the characters is important in the pattern you specified in the field *Class Name*. The default value of this button is stored in the M4 variable **M4_find_case_sensitive**.

Buttons :

OK Use this button to open only the first class matching the specified search pattern. The dialog box then disappears.

Next This button can be used to open the first class or the next class in line matching the specified pattern. The dialog box does not disappear, so that you are still able to open another class matching the pattern.

Find Class Feature dialog box

In: Class Browser

Specify the pattern in the *Find Feature dialog box*:

Feature Name

Here you can specify a glob-style search pattern. For details on this type of pattern matching, refer to your TCL documentation.

Find Here you can specify whether you want to search on attributes, operations, or both. At least one of the buttons must be switched on.

Case Sensitive Switch this button on if the case is important in the pattern you specified in the *Feature Name* field. The default value of this button is stored in the M4 variable **M4_find_case_sensitive**.

Use Feature Filters

Switch this button on to restrict the search to only those features that have already met the filter criteria specified by **View > Filter Features**. The default value of this button is stored in the M4 variable **M4_find_using_filters**.

Note : If this button is off and a filter is on, ObjectTeam can locate a feature that does not meet the filter criteria. ObjectTeam opens the class containing that feature, but only those features that meet the filter criteria are displayed in the Class Browser.

Buttons

OK Use this button to open only the first class matching the specified search pattern. The dialog box then disappears.

Next Use this button to open the first class or the next class matching the specified pattern. The dialog box does not disappear, so that you are still able to open another class matching the pattern.

Tool Bar

In: Class Browser

The *tool bar* is a window area that contains icons for quick activation of frequently used menu entries. Instead of executing such a menu entry from the menu bar, you can just click on the corresponding icon in the tool bar.

The tool bar is located between the menu bar and the context area. You can find out which menu entry a toolbar button represents by dragging your mouse pointer over the button in the toolbar and keeping it there for a few seconds. A ToolTip with the name of the menu entry appears.

The tool bar in the class browser contains the following buttons by default:

- File | Open
- File | Close

Show Scope dialog box

In: Class Browser

Name : The name of the item.

Type : The type of the item.

Scope : The scope of the item: *phaseDef*, *phaseRef*, *system*, or *file*.

Qualified by: The item that qualifies the selected item (for example, the class name of a selected attribute).

The scope of an item can be changed in a diagram editor using Item | Edit Scope.

Menu Customization, CustMenu[button_type]Button '[name]' dialog box

In: [Customization Editor](#)

You can only edit the properties of objects defined on the current [browser level](#): the ones that are have a trailing asterisk (*) in their *Level* indication.

The **CustMenu[button_type]Button '[name]' dialog box** contains the following property pages:

- [Storage Page](#)
- [Interface Page](#)
- [Command Page](#)
- [Enable /disable Page](#)
- [Arbiters Page](#) (MenuBarButton and CascadeButton only)

ObjectType Customization - ObjectObject '[name]' dialog box

In: [Customization Editor](#) The **Edit Properties dialog box** contains the following property pages:

- [Storage Page](#)
 - [Interface Page](#)
 - [Command Page](#)
-

Storage page

Specifies for which [browser levels](#) the objecttype is stored. This page corresponds with the scope column in the Objecttype customization editor. Depending on the browser level at which you are editing the objecttype, one or more of the fields are grayed out.

- Project - This input field defines the project for which the objecttype is valid. This field supports wild cards.
- Configuration - This input field defines the configuration for which the objecttype is valid. This field supports wild cards.
- Phase - These four check buttons allow you to select a subset of phases in which the objecttype is valid.
- System - These two check buttons allow you to one or two system types in which the objecttype is valid.
- Read Only - Enabling this check button inhibits redefinition of this objecttype at lower browser levels.

Interface page

The Interface page lets you select icons and a filename extension for storage of the object in the filesystem.

Command page

Specifies the action taken if the button is pushed.

- Command - The command itself. You can select one of three options:
 - Predefined Procedures
 - Object Operations
 - Command - custom made for the embedded Tcl interpreter.
- Command kind - Choose one of four command types:
 - Internal - command for the internal Tcl interpreter.
 - External Output Only - command using the [Monitoring Window](#) as output device.
 - External Input/Output - command using the [Execution Window](#) as output device.
 - External Own Interface - command using its own graphical user interface.
- Interface - Control various parts of the behavior of the command.

Open Strategy Definition Customization - Search Strategy '[name]' dialog box

In: Customization Editor

Which fields appear in the **Edit Properties dialog box** depends on the *Type* of the selected strategy definition:

search _____ Name | Decomposition Flags | Diagram Types
createFile _____ Name | Diagram Qualifier | Diagram Name | Diagram Type
createSystemAndFile _____ Name | Diagram Qualifier | Diagram Name | Diagram Type | System Name
userDefined _____ Name | TCL Procedure
group _____ Name | Members

Name A name to identify the strategy definition uniquely

Decomposition Flags

These flags specify the various subcategories of items to be searched for. The possible decomposition flags are:

- **decompSystems** - Search for systems with the same name as the current item
- **decompFiles** - Search for files with the same name as the current item.
- **decompComponents** - Search for diagrams having components with the same name as the current item
- **decompParents** - Search for diagrams having parent components with the same name as the current item
- **decompLeafs** - Search for diagrams having leaf components with the same name as the current item.

Diagram Types

- __cad Class Diagram
- __cod Collaboration Diagram
- __etd Sequence Diagram
- __std State Transition Diagram
- __ucd Use Case Diagram

Diagram Qualifier

- **\$item** - Substituted by the name of the item to be opened
- **\$ itemQual** - Substituted by the name of the qualifier of the item to be opened
- **\$ diagItem** - Substituted by the item referred to by the current diagram
- **\$ diagQual** - Substituted by the qualifier item of the current diagram

Diagram Name / System Name

- **\$itemName** - Substituted by the name of the item to be opened
- **\$ itemQualName** - Substituted by the name of the qualifier of the item to be opened
- **\$ dataType** - Substituted by the data_type property of the item to be opened
- **\$ diagName** - Substituted by the name of the current diagram
- **\$ diagQualName** - Substituted by the name of the qualifier of the current diagram
- **\$ diagType** - Substituted by the type of the current diagram

Diagram Type

- **\$diagType** - Substituted by the type of the current diagram
- __cad (Class Diagram)
- __cod (Collaboration Diagram)
- __etd (Sequence Diagram)
- __std (State Transition Diagram)
- __ucd (Use Case Diagram)

TCL Procedure Here you can specify your own navigation strategy, written in Tcl. You specify the name of the Tcl procedure here.

The procedure itself can be stored for instance in the user customization file `u_desk.tcl`.

Members This field specifies the names of the member strategies (or other strategy groups). Through strategy groups you can combine the navigation strategies **search**, **createFile**, and **createfileAndSystem**

Buttons:

OK Store the current property values and leave the dialog box

Apply Now Store the current property values and do not leave the dialog box

Cancel Do not store the current property values and leave the dialog box.

Open Strategy Availability Customization - Usage of [strategy_type]Strategy '[name]' dialog box

In : Customization Editor

The **Usage of <strategy_type>Strategy '<strategy_name>' dialog box** contains one property page with the following fields:

<i>Strategy</i>	Uniquely identifies the open strategy
<i>Item Type</i>	Select an <u>item</u> here.
<i>Context</i>	Select a <u>phase</u> here.
<i>Diagram Type</i>	Select a <u>diagram type</u> here
<i>Component Type</i>	Indicates the type of components for which a property can be manipulated.
<i>Label Type</i>	Indicates the type of labels within a <u>diagram</u> for which the property can be manipulated. Examples of label types are: <code>cad_class</code> , <code>data_store</code> , <code>con_message</code> .
<i>Condition</i>	The values <i>KEY</i> and <i>NO_KEY</i> can be specified for properties of key and non-key <u>attributes</u> of a class. <i>TOP</i> , <i>MIDDLE</i> and <i>LEAF</i> can be specified for components in a hierarchical diagram technique.

Buttons :

OK Store the current property values and leave the dialog box

Apply Now Store the current property values and do not leave the dialog box

Cancel Do not store the current property values and leave the dialog box.

Property Definition Customization, Property Definition '[name]' dialog box

In: Customization Editor

The **Property Definition '[name]' dialog box** contains only one property page with the following fields:

Name A name to uniquely identify the property (not editable)

Long Name the title of the property in the user interface, e.g. the Edit Properties dialog box in diagram editors.

Interface Class Select an entry from the drop-down list of Tcl dialog elements. These elements are used to present the property and its value in the Edit Properties dialog box.

Examples: `CheckButton`, `ComboBox`, `DropDwnComboBox`

Interface Class Options

Options to be used when the interface class is initialized in the property dialog box, e.g.: `-entrySet {yes no}`

Interface Class Members

This field is only applicable if you edit the properties of a property *group*: here you can select the properties you want to include in your group.

Buttons :

OK Store the current property values and leave the dialog box

Apply Now Store the current property values and do not leave the dialog box

Cancel Do not store the current property values and leave the dialog box.

Property Location Customization, Property Location '[name]' dialog box

In: Customization Editors

The **Property Location '[name]' dialog box** contains only one property page with the following fields:

<i>Container Kind</i>	The kind of browser object for which the property can be manipulated. You can select a browser object from the drop-down list. The entry <i>Clear</i> indicates that all locations for the definition type specified so far are removed from the list of locations where a value for the property can be supplied. This prevents the accumulation of all locations specified in location files at higher levels than the current customization level.
<i>Container Type</i>	The type of object for which the property can be manipulated. Examples of container types for the Container Type <i>Item</i> are: <i>cl</i> and <i>de</i> . Examples for the Container Type <i>Component</i> are: <i>association</i> , <i>parameter</i> and <i>qualif_aggr</i> .
<i>Phase Type</i>	Indicates the <u>Phase</u> in which the property is available.
<i>Diagram Type</i>	Indicates the <u>file version</u> in which the property is available.
<i>Component Type</i>	Indicates the type of components for which a property can be manipulated. Specifying this is only useful if the current Container Kind is <i>Item</i> .
<i>Label Type</i>	Indicates the type of labels within a <u>diagram</u> for which the property can be manipulated. Examples of label types are: <i>cad_class</i> , <i>data_store</i> , <i>con_message</i> .
<i>Condition</i>	The values <i>KEY</i> and <i>NO_KEY</i> can be specified for properties of key and non-key <u>attributes</u> of a class. <i>TOP</i> , <i>MIDDLE</i> and <i>LEAF</i> can be specified for components in a hierarchical diagram technique.

Buttons :

OK Store the current property values and leave the dialog box

Apply Now Store the current property values and do not leave the dialog box

Cancel Do not store the current property values and leave the dialog box.

View Customization, ViewObject '[name]' dialog box

In: View Customization Editor

The **ViewObject '[name]' dialog box** contains the following property pages:

- Storage Page
- Interface Page
- Properties Page
- Sort Command Page

New Customization File - dialog box

In: Browser

Customization File:

You can select the desired customization file from the list box displayed here:

- * .mnu
- * .vie
- * .pnl
- objtype .objtype
- opendefs .opendefs
- openlocs .openlocs
- propdefs .propdefs
- proplocs .proplocs
- checkconfig .checkconfig
- copyspecs .copyspecs

If you want to create a customization file that is not listed in the list box, enter its name in this field:

- * .gdr (group definition rules file)

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the information area.

Edit Create the new object and start the tool to edit the new object with (a Customization Editor, the default text editor or the Edit Control Panel dialog box).

Cancel Do not create the object and leave the dialog box.

New Button/Separator - dialog box

In: Menu Customization Editor

Parent Select the button under which you want to arrange the new button or separator
Name (Not for Separators)
 Enter the name of your choice here. This name will be added to the hierarchy in the display area
 of the customization editor. An asterisk behind the new button indicates that the button was
 created on the current browser level

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the alphabetical list in the display area of the customization editor.

Edit Create the new object, leave the dialog box and start the Edit <Object> Properties dialog box

Cancel Do not create the object and leave the dialog box.

New Object Type - dialog box

(In: ObjectType Customization Editor)

This dialog box appears when you select **Edit | New...** from the menu bar.

Repository Type Select the repository type for which you specify the new object type.

Browser Type Some repository types are subdivided into browser types. For example, the repository type *Graph* has browser types *cad*, *ccd*, *dfd*, and so on. The new object type will be added to the alphabetical list in the display area of the customization editor.

Buttons :

OK Create the new object and leave the dialog box

Edit Create the new object, leave the dialog box and start the Edit <Object> Properties dialog box

Cancel Do not create the object and leave the dialog box.

New Navigation Strategy Definition - dialog box

(In: Opendefs.opendefs Customization Editor)

Name A name to uniquely identify the open strategy

Type One of the following types of strategies:

- **search** - The result of this strategy is a list of diagrams to be opened
- **createFile** - The result of this strategy is a diagram to be opened after it is created in the current system
- **createSystemAndFile** - The result of this strategy is a diagram to be opened after the specified system and a diagram within that system is created
- **userDefined** - Select this type if you want to create your own open strategy in a Tcl procedure. You can specify the name of the procedure with Edit | Edit Properties.
- **group** - Strategy groups are combinations of strategies of the above three types. They allow you to combine all strategies that are applicable at the same location in one definition.

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the list in the display area of the customization editor.

Edit Create the new object, leave the dialog box and start the Edit <Object> Properties dialog box

Cancel Do not create the object and leave the dialog box.

New Navigation Strategy Location - dialog box

(In: Openlocs.openlocs Customization Editor)

This dialog box appears when you select **Edit | New...** from the menu bar.

Strategy Select a navigation strategy from this drop down list

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the list in the display area of the customization editor.

Edit Create the new object, leave the dialog box and start the Edit <Object> Properties dialog box

Cancel Do not create the object and leave the dialog box.

New Property Definition - dialog box

(In: Propdefs.propdefs Customization Editor)

This dialog box appears when you select **Edit | New...** from the menu bar.

Name A name to uniquely identify the property

Long Name The title of the property in the user interface, e.g. the Edit Properties dialog box in diagram editors.

Interface Class The Tcl name of the dialog element used to present the property and its value in the Property dialog box.

Examples : CheckButton, ComboBox, DropDownComboBox

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the list in the display area of the customization editor.

Edit Create the new object, leave the dialog box and start the Edit <Object> Properties dialog box

Cancel Do not create the object and leave the dialog box.

New Property Location - dialog box

(In: Proplocs.proplocs Customization Editor)

This dialog box appears when you select **Edit | New...** from the menu bar.

Property Select the property you want to create the new location for.

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the list in the display area of the customization editor.

Edit Create the new object, leave the dialog box and start the Edit <Object> Properties dialog box

Cancel Do not create the object and leave the dialog box.

New View - dialog box

(In: View Customization Editor)

This dialog box appears when you select **Edit | New...** from the menu bar.

Repository Type Select the repository type for which you specify the new view

Browser Type Some Repository Types are related to more than one Browser Type, e.g. if you have selected the Repository Type *SystemVersion* you can select the Browser Types *SystemVersion* or *DocumentVersion*.

Name If you select the Browser Type *all*, you specify all possible Browser Types that are related to the Repository Type.

Buttons :

OK Create the new object and leave the dialog box. The new object is added to the list in the display area of the customization editor.

Edit Create the new object, leave the dialog box and start the Edit <Object> Properties dialog box

Cancel Do not create the object and leave the dialog box.

Edit Control Panel - dialog box

With this dialog box you can edit the customization files **<diagram_name>.pnl**. These files affect the lay-out and contents of control panels in diagram editors: the control panel of the Class Diagram Editor for instance.

You can select which symbols you want to be available for the users of a particular diagram editor and which not.

<i>New ...</i>	Use this button to add a symbol to the current selection, which is displayed under <i>Contents</i> :. You can only select symbols from a fixed list in the <u>Select Panel Entry dialog box</u> .
<i>Test ...</i>	Use this button to see how the symbols listed under <i>Contents</i> : will be arranged in the control panel. The result depends on the number of columns selected.
<i>Delete</i>	Use this button to delete a symbol from the current selection of symbols, listed under <i>Contents</i> :
<i>Contents :</i>	This box shows the current selection of symbols.
<i>columns</i>	Select the number of columns (one, two or three) you want the selected symbols to appear in in the control panel of the particular diagram editor. Use the button <i>Test...</i> to see how the arrangement of symbols in the control panel will look.

You can change the order of the diagram symbols in the control panel by using the drag and drop facility of ObjectTeam.

Press the *OK* button to confirm the current selection of symbols or the *Cancel* button to discard the operation.

Select Icon - dialog box

Use this dialog box to select a different icon for the *menu button* that you are customizing in the [customization editor](#). You can start it from the **Interface** page of the [Menu Customization dialog box](#).

The icon selected here is used if the menu button is put in the [tool bar](#) of the tool you are customizing.

The icons you can select refer to bitmap files that are stored in the directory **<M4 home>/bitmaps**.

Deleting menus, object types and views

(In: [Customization Editor](#))

Use **Edit > Delete** to delete objects from the customization environment. For the various editors this means the following:

Menu Customization

The menu entry will disappear from the menu. All defined properties are lost. If the object had any sub-menu entries, these are also lost.

Objecttype Customization

The link between the browser object type and the repository object type is removed. All defined properties are lost.

View Customization

The view will disappear from the **View** menu.

Note that changes will take effect after a restart of *ObjectTeam*.

Editing Menu Properties

(In: Customization Editor)

Use **Edit > Edit Properties** from the menu bar to modify user interface objects.

Whether a menu entry in the cascading menu **Edit > Edit Properties** is available, depends on the type of Customization Editor that is started:

Edit > Edit Properties Menu customization

Edit Properties for a menu component leads to the Menu Customization dialog box.

Edit > Edit Properties Objecttype customization

Edit Properties for a object leads to the Objecttype Customization dialog box.

Edit > Edit Properties View customization

Edit Properties for a view leads to the View Customization dialog box.

Filtering Menu Entries

(In: [Customization Editor](#))

Use the filters from the [menu bar](#) to control the number of visible elements in the display area.

Filter On Active Entries - menu entry

Use **Filter > Filter On Active Entries** to toggle between two display modes:

- Display all entries defined at the current and higher [browser levels](#)
- Display only the entries available at the current browser level

The check button indicates the current setting.

Filter Out Separators - menu entry

This option appears only in the Menu Customization Editor.

Use **Filter > Filter Out Separators** to toggle the visibility of the separators in the display area.

The check button indicates the current setting.

Filter On Current Active File Entries - menu entry

This option appears only in the Objecttype Customization Editor and the View Customization Editor.

Use **Filter > Filter On Current Active File Entries** to toggle the view between:

- Display all entries defined at the current and higher browser levels
- Display only the active objects defined at this browser level

The check button indicates the current setting.

Creating New Customization Object

(In: Customization Editor)

Use Edit > New... to create a new customization object. Depending on the Customization Editor you can create the following customization objects with this menu entry:

Menu Customization Editor

New Menu

Menu Object Type Customization Editor

New Objecttype

View Customization Editor

New View

Property Definition Editor

New Property Definition

Property Availability Editor

New Property Location

Open Strategy Definition Editor

New Navigation Strategy

Open Strategy Availability Editor

New Navigation Strategy Usage

Redefining an object

(In: Customization Editor)

Use **Edit > Redefine** in the menu bar to copy the nearest higher browser level specification to the current browser level and modify it. For example a specification on corporate level can be redefined on project level. When a specification is read-only no redefinition can be made and a warning will be generated.

If the copy is successful, the appropriate Edit Properties dialog box is invoked automatically.

Restoring Settings

(In: Customization Editor)

Use **File > Reload** in the menu bar to restore the last saved configuration settings.

Saving a Customization File

(In: Customization Editor)

Use **File > Save** in the menu bar to save any changes made in the customization editor or one of the customization dialog boxes. These changes are stored in a customization file.

Note that changes will take effect after a restart of the ObjectTeam product.

Menu Bar - Customization Editor

Below, the default menus of the customization editors are listed.
The differences between the editors (menu, objecttype or view) are indicated in *Italics*.

[File] [Edit] [Options] [Filter]

- **File** menu
 - Reload
 - Save
 - Exit
- **Edit** menu
 - New
 - Edit Properties
 - Redefine
 - (*Not in Property Definition Editor*)
 - (*Not in Property Availability Editor*)
 - (*Not in Open Strategy Definition Editor*)
 - (*Not in Open Strategy Availability Editor*)
 - Info
 - Delete
- View menu
(*Not in the Menu Customization Editor*)
 - ToolBar
 - ContextArea
 - MessageArea
 - Icon
 - Small Icon
 - Detail
- **Options** menu
 - Font ...
- **Filter** menu
 - Filter On Active Entries
 - (*Not in Property Definition Editor*)
 - (*Not in Property Availability Editor*)
 - (*Not in Open Strategy Definition Editor*)
 - (*Not in Open Strategy Availability Editor*)
 - Filter Out Separators
 - Only in Menu Customization Editor*
 - Filter On Current File Entries
 - Not in Menu Customization Editor*
- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

Customization Files - Overview

Customization files can be defined as:

- Customization file on a browser level
 - User Customization file
 - Customization file in an ObjectTeam module
- Customization files on browser levels and user customization files can be edited with

- Customization Editor
- Control Panel Dialog Box
- Text editor

Module customization files can only be edited with a text editor.

You can define the following types of customization files:

- User Interface Customization Files
to customize menus, control panels, views, objects and properties
- Check Configuration Customization File
to customize diagram and model checking
- Document Generation Customization Files
to customize files that are used during Document Generation
- Code Generation Customization Files
to customize files that are used during Code Generation
- User Customization Files
any customization file residing in the <user customization files> object
- Environment Customization files
the M4env, Meta4UserEnv and other environment files
- Miscellaneous Customization Files

See the *ObjectTeam Customization Guide* for more information on customization files.

Changing the position of a menu option

(In: Customization Editor)

Changing an existing menu option requires different actions for different situations:

If the option is defined at the current browser level (indicated by an asterisk)...
select **Edit > Edit Properties**

If the option is defined at a higher browser level and not set to read only...
select **Edit > Redefine**

Changing a menu option

(In: Customization Editor)

Changing an existing menu option requires different actions for different situations:

If the option is defined at the current browser level (indicated by an asterisk)...
select **Edit > Edit Properties**

If the option is defined at a higher browser level and not set to read only...
select **Edit > Redefine**

Customizing the browser

1. Make sure the browser is on the appropriate level.
2. Open the browser object <customization files> on that level.
You can do that by:
 - double -clicking the object in the information area
 - Selecting the object in the information area and using File > Edit
 - Selecting the object in the information area and using File > Open
The child objects of the opened browser object (i.e. the customization files) become visible in the information area
3. Open the appropriate customization file to start the customization editor.
If the customization file does not exist, you can create it with **File > New > Customization File Version**
These are your options:
 - desk .menu - for customizing the menu structure.
 - objtype .objtype - for customizing the browser objects
 - desk .view - for customizing the browser views
4. Use the Customization Editor to make the desired changes.

The other files visible in the Customization Files folder also customize parts of the ObjectTeam product but not via a customization editor.

Adding a menu option

(In: Customization Editor)

You need to edit a customization file of type ***.mnu** to add a menu option to the browser or to another ObjectTeam tool

1. Open the appropriate menu customization file in the browser.
Be aware of the browser level you select the customization file from
2. Select Edit > New > MenuBarButton from the Menu Customization Editor
3. Use Edit > Properties to specify the new menu button further

Customization Editors - Overview

Customization Files -----

Customization files can be edited with the following Customization Editors/Tools:

*.mnu	<u>Menu Customization Editor</u>
*.vie	<u>View Customization Editor</u>
*.pnl	<u>Control Panel dialog box</u>
objtype.objtype	<u>Object Type Customization Editor</u>
proplocs.proplocs	<u>Property Availability Editor</u>
propdefs.propdefs	<u>Property Definition Editor</u>
opendefs.opendefs	<u>Navigation Strategy Definition Editor</u>
openlocs.openlocs	<u>Navigation Strategy Availability Editor</u>
copyspecs.copyspecs	<u>Copy Specifications Customization Editor</u>
checkconfig.checkconfig	<u>Check Configuration dialog box</u>
modules.modules	<u>Module Availability Editor</u>

If you want to edit a customization file of another type than those listed above, you can use the default text editor.

When you have changes to the customization files, you should restart the Browser to activate your changes. Use the option Utilities | Clone to start a second browser and close the first one.

Display Area (general)

(In: [Customization Editor](#))

[\[menus\]](#) [\[property availability\]](#)
[\[objecttypes\]](#) [\[open strategy availability\]](#)
[\[views\]](#) [\[open strategy definition\]](#)
[\[property definition\]](#)

The display area of a customization editor lists customization objects of a certain type.

Most of the customization editors have three View modes:

- Large Icons
- Small Icons
- Details

You can change the current view through the View menu.

In *Details* view, the properties of these objects are represented in columns. These properties are different for every type of customization object. You can change them using **Edit > Edit Properties** from the menu bar. You can only edit the properties of objects defined on the current [browser level](#). The ones that are have a trailing asterisk (*) in their *Level* indication.

In addition, you can filter out certain objects from the display area using the commands under the [Filter menu](#).

Display Area (menu customization)

The default Menu Customization display area shows all the menu bar entries. You can navigate by:

- unfolding menus
- folding menus
- edit menu properties by:
 - selecting and using [Edit > Edit Properties](#)
 - double -clicking the intended menu object

The Menu Customization Editor supports **Drag and Drop** of menu objects. The following rules apply:

- If a Menu object is dropped on another Menu object it is placed above that object.
- If a Menu object is dropped on another Menu object **below** a MenuBar or Cascade object, the object is placed in the MenuBar menu or Cascaded menu.

Display Area (objecttype customization)

The default display area of the Object Type Customization Editor shows all the Repository Types and, in *Detail* view, also their object properties. You can edit object properties by:

- selecting and using [Edit > Edit Properties](#)
- double -clicking the intended object

Display Area (view customization)

The default display area of the View Customization Editor shows all the Repository Types and, in *Detail* view, also their view properties. You can edit view properties by:

- selecting and using [Edit > Edit Properties](#)
- double -clicking the intended object

Display Area (Property Definition customization)

The default display area of the Property Definition Editor shows all the Properties and, in *Detail* view, also their definition properties. You can edit property definition properties by:

- selecting and using Edit > Edit Properties
- double -clicking the intended object

Display Area (Property Availability customization)

The default display area of the Property Availability Editor shows all the Properties and, in *Detail* view, also their location properties. You can edit property location properties by:

- selecting and using Edit > Edit Properties
- double -clicking the intended object

Display Area (Open Strategy Availability customization)

The default display area of the Open Strategy Availability Editor shows all the Navigation Strategies and, in *Detail* view, also their location properties. You can edit strategy location properties by:

- selecting and using Edit > Edit Properties
- double -clicking the intended object

Display Area (Open Strategy Definition customization)

The default display area of the Open Strategy Definition Editor shows all the Navigation Strategies and, in *Detail* view, also their definition properties. You can edit strategy definition properties by:

- selecting and using Edit > Edit Properties
- double -clicking the intended object

Module Availability - Info box

In: Customization Editors

The Module Availability Info box contains the following information:

<i>Long Name</i>	The long name of the selected module
<i>Select State</i>	
<i>Level</i>	Shows at which browser level the module is defined.
<i>Location</i>	.
<i>Name</i>	Short name
<i>Version</i>	The module version.
<i>Type</i>	the type of the module
<i>RequiredLicences</i>	A license feature that is needed for this module.
<i>RequiredModules</i>	List of modules required for this module.
<i>IsSupportingModule</i>	Indicates whether the module is supporting other modules or functions autonomically.

Select Module Directory Dialog box

In: Customization Editors

When you press the button Browse in the New Module dialog box the Select Module Directory dialog box appears.

In this dialog box you can select the directory from which you want to select a new module.

Buttons :

OK Press OK to accept the changes and leave the Select Modules dialog box.

Cancel Press Cancel to discard the changes and leave the Select Modules dialog box.

Add Required Modules

In : [Customization, Module Availability Editor](#)

When you have selected a module to add to your ObjectTeam software, it is possible that more (supporting) modules must be added. The option Edit | Add Required Modules adds these automatically for you.

To add required modules

1. In the Modules Availability Editor, select Edit | Add Required Modules.
ObjectTeam checks the installed modules and adds supporting modules if necessary.

Check Module Requirements

In : Customization, Module Availability Editor

To check the module specifications:

1. In the Modules Availability Editor, select **Check | Requirements**. ObjectTeam checks the required and conflicting modules of each module specification and displays the results in a dialog box.
 1. If the module specifications meet all requirements, an info box appears.
Press OK to leave this box
 2. If required modules are missing, the info box lists them.
Activate the required modules.
 3. If conflicting modules are active, the info box lists them.
Deactivate the conflicting modules.
2. If you modified the module specifications, return to step 1.

Delete Module

In : [Customization, Module Availability Editor](#)

To delete a module specification:

1. Open the Modules Availability Editor
2. Select the module specification. Select **Edit | Delete**.
ObjectTeam deletes the module specification from the modules.modules customization file and updates the Display Area.

Module Info

In : Customization, Module Availability Editor

To view additional information about a module specification:

1. In the Modules Availability Editor, select the module specification. Select **File | Info** or click on the Info button in the Tool bar.
An Info dialog box appears

Module Location

In : Customization, Module Availability Editor

The Location field provides the full path of the module directory. If you move a module directory, you must edit the Location field of any module specification that refers to the module.

To change the Location of a module specification:

1. Open the Modules Availability Editor
2. Select the module specification. Select Edit | Change | Location.
The Select Directory dialog box appears.
3. Select the new module directory, then select OK.
ObjectTeam updates the module specification.

Install New Module

In : Customization, Module Availability Editor

To create a module specification:

1. Open the Modules Availability Editor
2. Select **Edit | New**.
The Select Directory dialog box appears.
3. Select the module directory, then select OK.
ObjectTeam adds the module specification to the modules.modules customization file and updates the Display Area.
4. Check the module specifications
5. Select **File | Save** to save the modules.modules customization file.
6. Select **File | Exit** to exit from the Modules Availability Editor.
7. In the Browser, select **Utilities | Clone** to open a new Browser. ObjectTeam opens a new Browser with the customization file in affect.

Select State

In : Customization, Module Availability Editor

The Select State field determines whether the module specification activates (on, the default) or deactivates (off) the module. Edit this field to activate or deactivate a module.

To change the Select State of a module specification:

1. Open the Modules Availability Editor
2. Select the module specification.
3. Select **Edit | Change | Select State.**
ObjectTeam changes the select state from on to off, or from off to on.

User Interface Customization files

In : Customization Editors

Menu Customization Editor

- **desk .mnu** - Browser menus
- **class .mnu** - Class Browser menus
- **diagram .mnu** - Generic diagram menus
- **cad .mnu** - CD menus
- **cod .mnu** - COD menus
- **etd .mnu** - SD menus
- **std .mnu** - STD menus
- **ucd .mnu** - UCD menus

View Customization Editor

- **desk .vie** - Browser views

Object Type Customization Editor

- **objtype .objtype** - Browser object types

< Diagram Control Panel> dialog box

- **cad .pnl** - CD control panel
- **cod .pnl** - COD control panel
- **etd .pnl** - SD control panel
- **std .pnl** - STD control panel
- **ucd .pnl** - UCD control panel

Diagram Navigation Strategy Customization Editors

- **opendefs .opendefs** - Open Strategy Definition Customization file
- **openlocs .openlocs** - Open Strategy Availability Customization file

Property Customization Editors

- **propdefs .propdefs** - Property Definition customization file
- **proplocs .proplocs** - Property Availability customization file

User Customization Files

In : Customization Files

The user customization files reside on root level (above corporate level), under the pseudo-object *<user customization files>*.

These files are user-dependent and are stored in the file system: in the **icase** directory under the user's home directory:

- **< user_home>/icase** (Unix-based systems)
- **< user_home>\icase** (Windows NT 3.51)
- **< home_drive>:\Windows\Profiles*<user>*** (Windows 95/NT 4.0)

The user customization file are:

- the objecttype customization file (**objtype.objtype**)
- all the menu customization files (***.mnu**)
- all the diagram control panel files (***.pnl**)
- the view customization file **desk.vie**
- the **Meta4UserEnv** file
This file contains the initial M4 variables. This file is not stored in the icase directory, but in the user's home directory as:
 - **. Meta4UserEnv** (Unix-based systems)
 - **Meta4UserEnv .txt** (Windows-based systems)

This name is overruled by the shell environment variable `Meta4UserEnv`.

• the following `u_*.tcl` files :

- **u_desk .tcl** - Browser customization
- **u_mtool .tcl** - Monitoring Window customization
- **u_uce .tcl** - Customization Editor customization
- **u_ude .tcl** - Diagram Editor customization
- **u_clbrowse .tcl** - Class Browser customization

In these files, you can include user-defined Tcl procedures, with which you can extend the functionality of the corresponding ObjectTeam tools

- every other file you have put in the **icase** directory yourself

Activating Modules

In : [Customization, Module Availability Editor](#)

You can activate a module in the following ways:

- Create a module specification in the modules.modules customization file. By default, the new module specification activates the module. This is the most common way of activating a module.
- If a customization file contains a module specification that deactivates a module, you can edit the module specification so that it activates the module.

Deactivating Modules

In : [Customization, Module Availability Editor](#)

You can deactivate a module in the following ways:

- If the current customization file contains a module specification that [activates a module](#), you can edit the module specification so that it deactivates the module. Alternatively, you can delete the module specification.

Start Module Availability Editor

In : [Customization Editors, Module Availability Editor](#)

To start the Modules Availability Editor:

1. In the Browser, open the <customization files> pseudo-object on the level on which you want to edit the customization file version.
The list of customization file versions appears in the Display Area.
2. Is there a *modules.modules* customization file version?
 1. If yes, double-click on it.
 - The **Modules Availability Editor** starts up.
 2. If no, create it:
 - If you are on Corporate or User Customization level, select **File | New | External File**; otherwise, select **File | New | Customization File Version**.
The [New External File dialog box](#) appears.
 - Select *modules.modules*, then select Edit.
 - The Modules Availability Editor starts up.

Menu Customization Editor

Customization Editors -----

With this editor you can edit menu customization files (*.mnu). Menu customization allows you to modify menu bars, menu buttons, tool bar buttons and the operations they initiate. You can edit these menu settings for the browser and for other tools of ObjectTeam.

You can perform the following customization tasks on menus and menu buttons:

- Create new menus, menu buttons and separators
- Redefine existing ones
- Edit their properties
- Select a corresponding item for the Tool Bar

See also:

- Delete Menu-item
- Change Position of Menu-item
- Change Menu-items
- Filter Menu-items

ObjectType Customization Editor

Customization Editors -----

With this editor you can edit the customization file **objtype.objtype**.

Objecttype customization allows you to:

- Customize the mapping of objects to repository objects
- Redefine existing mappings
- Edit their properties

See also

- Delete ObjectType

Navigation Strategy Definition Editor

Customization Editors

With this editor you can edit the customization file **opendefs.opendefs**.

Open Strategy Definition customization allows you to customize navigation strategies between diagrams. You can customize the options offered to end users when they use File > Open for diagram objects.

You can perform the following customization tasks on open strategy definitions:

- Create new navigation strategies
- Edit properties of navigation strategies

Navigation Strategy Availability Editor

Customization Editors

With this editor you can edit the customization file **openlocs.openlocs**.

Open Strategy Availability customization allows you to customize the location of navigation strategies between diagrams.

You can perform the following customization tasks on open definitions:

- Create new navigation strategy locations
- Edit properties of navigation strategy locations

Property Definition Editor

Customization Editors

With this editor you can edit the customization file **propdefs.propdefs**.

Property Definition customization allows you to customize properties and property groups.

You can perform the following customization tasks on property definitions:

- Create new properties
- Edit properties of properties

Property Location Editor

Customization Editors

With this editor you can edit the customization file **proplocs.proplocs**.

Property Availability customization allows you to determine where end users are enabled to edit which properties and which property groups.

You can perform the following customization tasks on property locations:

- Create a new location for a property (group)
- Edit properties of property (group) locations

View Customization Editor

Customization Editors

With this editor you can edit view customization files (***.vie**).

View customization allows you to control the way browser objects are presented in the information area of the browser on various levels.

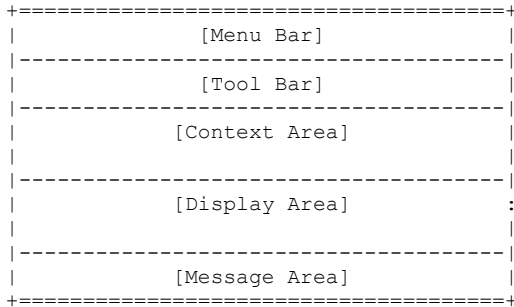
You can perform the following customization tasks on views:

- Create new views
- Redefine existing ones
- Edit their properties

Customization Editor - Window

Customization Editors

The default customization editor window contains the following sections.



- **Menu Bar** - You select menu entries from here. The available menu entries per target file can differ.
- **Tool Bar** - you can select frequently used menu entries from here.
- **Context Area** - Displays information about the current context, such as current file, etc.
- **Display Area** - displays the customizable objects.
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Modules

In: Customization Editors

A modules.modules customization file contains one or more module specifications. Each module specification activates or deactivates a module. A modules.modules customization file can exist at any Browser level, from corporate level through user level.

Each module is a directory of files that implement an optional ObjectTeam feature. The directory name is the module name.

- Each module directory must contain (see the module directories in *M4_home\modules*) a specially formatted file named `properties.properties` that defines the module's properties.
- Each module can contain the following four subdirectories (`bin`, `config`, etc, and `tcl`) that contain the files that implement the feature.

Copy Specifications Customization - Copy Specification '[name]' dialog box

In: Customization Editor

- Select gdr file* You can either select a gdr file from the list or type a filename.
Note . If the file is not in the list, it does not exist yet. You must create the gdr file first.
- Level* Type the browser level to which the copy specification applies.
- Description* Type a description of the copy specification
- Apply Now button* This button applies the change to the Copy Specification without leaving the Copy specification dialog box

Edit <name>.gdr dialog box

In: Customization Editor)

Select File | New | Customization File Version.

Type a name followed by .gdr in the textbox and press Edit.

The Edit <name> dialog box appears.

- File selector
- Item selector

New

File selector: The New File Selector dialog box appears

Item selector The New Item selector dialog box appears

Delete

File Selectors

Description

New File selector dialog box

Customization Files

New Item selector dialog box

Customization Files

Create New Button/Separator

(In: Customization Editor)

Edit > New Menu allows you to create the following objects:

- Menu Bar Button - a button in the menu bar
- Cascade Button - a button leading to a sub menu
- Push Button - a button in a menu
- Check Button - a button with an on-off indicator
- Radio Button - a button coupled with other radio buttons. Only one of the coupled buttons can be switched on.
- Separator - a line separating the button above and below it.

The New Button dialog box appears.

Create Copy Specifications Customization file

(In: Customization Editor)

1. Select **Edit > New...** from the menu bar.
The New Copy Specification dialog box appears.
2. Select a File Type from the list or type a new name.
3. Press Edit to edit the new copy specification
The Copy Specifications <name> dialog box will start.

Create Group Definition Rules

(In: Customization Editor)

1. Select File | New | Customization File Version.
2. Type a name followed by .gdr in the textbox and press Edit.
3. The Edit <name> dialog box appears.

Copy Specifications Customization Editor

In: Customization Editors

In ObjectTeam, you can copy diagrams from one phase to the other. Diagrams can be copied to a system residing in: a different phase in the same configuration a phase in a different configuration in the same project a phase in a different project in the same repository Properties may have been specified for the graphical objects in a diagram, item properties as well as component properties.

When a diagram is copied, the following elements are always copied along with the diagram:

- the graphical objects
- the component properties
- the item properties having scope File

There is a part of the behavior that you can customize. You can customize what must be done with:

Elements	File Type
item properties having scope System or Phase	all
CDMs	Class Diagram
decompositions	State Transition Diagram
	Use Case Diagram

The way these copy rules are specified is through item and file selectors. These selectors are stored in Group Definition Rules files. Selectors are a generic way of specifying a closure over a set of diagrams and items. The same mechanism of item and file selectors is used in Group Versions, hence the name Group Definition Rules.

The item selectors and file selectors are stored in customization files called Group Definition Rules files.

New Copy Specifications dialog box

In : Objtype.objtype Customization Editor

File Type This has the following options:

Text box Type a name if you want to create a new file type

List Select a name from the list if you want to use an existing file type that is not already defined

Edit When this button is pressed the Copy Specifications dialog box appears.

Test Control Panel - dialog box

With this dialog box you can test the edited Control Panel.

Press each button to test it.

Press OK when you are done to leave this box.

Copy Specifications Info box

In: Customization Editors The Copy Specifications Info box reflects the following information:

File Type The type of the file for which the copy specifications are defined.
Reflects the column File Type.

Group Definition Rules
The name of the Group Definition Rules file that is used.
Reflects the column Group definition rules.

Specification Level
Shows at which browser level the copy specification is defined.
Reflects the column Level.

Description Reflects the column Description.

Menu Info box

In: Customization Editors

The Menu Info box contains the following information:

Name The name of the selected menu

Type The type of the selected menu

Specification Level

Shows at which browser level the menu is defined.

Scope Displays the settings of Scope on the Storage Page of the Edit Properties dialog box.

ReadOnly Indicates whether you can edit this menu on this level.

UserDefined Indicates whether this menu-button is a default or userdefined button.

Visible Displays the settings of Visible on the Storage Page of the Edit Properties dialog box

MenuPath Shows the name of the menu together with the menus it belongs to.

Note . The name is a short name generated from the long name that you assign to the menu on the Interface page at Name. Uppercase becomes lowercase and spaces are removed.

Example : When you created a new menu under File called Open By Name at Name on the Interface page, the Menupath will show: file.menu.openbyname.

index Position of the menu in listing (the first menu-item has value 0). Example:

Edit

Cut: index = 0

Copy: index = 1

Paste: index = 2

Object Type Info box

In: Customization Editors

The Object Type Info box contains the following information:

Name

Type

Specification Level

Scope

ReadOnly

UserDefined

Open Strategy Definition Info box

In: Customization Editors

The Open Strategy Definition Info box contains the following information:

Name

Name

The following information reflects the settings of the Definition page in the Edit Properties dialog box:

Type

Specification Level

Shows at which browser level the menu is defined.

Decomposition Flags

Diagram Types

Open Strategy Availability Customization - Info box

In: Customization Editors

The Open Strategy Availability Info box contains the following information:

Name : The following information reflects the names of the Open Strategy Availability.

Name

Location : The following information reflects the settings of the Location page in the Usage of
<strategy_type>Strategy <strategy_name> dialog box:

Type Strategy Type

Specification Level

Shows at which browser level the menu is defined.

Item Type

Phase

Diagram Type

Component Type

Label Type

Condition

Property Definitions Info box

In: Customization Editors

The Property Definitions Info contains the following information:

Definition : The following information reflects the settings of the Definition page in the Edit Properties dialog box:.

Name

Long Name

Interface Class

Interface Options

Interface Members

Level The following information shows at which browser level the definition is defined.

Specification Level

Property Locations (proplocs) Info box

In: Customization Editors

The Property Locations Info contains the following information:

Name : The following information reflects the names of the Property Locations.

Short Name

Long Name

Location : The following information reflects the settings of the Location page in the Edit Properties dialog box:

Container Kind

Container Type

Specification Level

Phase Type

Diagram Type

Component Type

Label Type

Condition

View Info box

In: Customization Editors

The View Info contains the following information:

Name Displays the View's name

Type Displays the object type: ViewObject

Specification Level

Shows at which browser level the view is defined.

Scope Displays the settings of Scope on the Storage Page of the Edit Properties dialog box.

ReadOnly Indicates whether you can edit this menu on this level.

UserDefined Indicates whether this view is default or userdefined.

repositoryType

browserType

Edit 'checkconfig' dialog box

You can modify the output of diagram checking operations through the Edit 'checkconfig' dialog box. This dialog box appears when you edit the customization file *checkconfig*.

These modifications apply to the following check operations:

- Check > Check Contents
- Check > Global Model
- Check > Local Model

The checks are collected in groups, of which the message type can be modified. The message type controls the way the message is presented in case one of the checks in the group fails. The options are:

- **warning** - The messages in the group are presented as warnings
- **default** - The messages in the group are presented as their default type
- **error** - The messages in the group are presented as errors
- **off** - The messages in the group are not presented

You can change the message type through the Check Value dialog box. This dialog box appears if you double-click on the group line in the group list box.

The dialog box contains a page for each of the phases. The group list box contain all the groups defined for this phase. The columns describe the following:

<i>Group</i>	The name of the group
<i>Check value</i>	The current message type of the group. You can modify this with the <u>Check Value dialog box</u> .
<i>Description</i>	The description of the group.

The Messages in selected group listbox contain the messages in the currently selected group. The message texts are stored in the `etc` directory of `<M4_home>`. Note that modifications in this file are corporate-wide.

The CheckConfig Customization Editor (dialog box)

In: Customization Editors

The top half of the CheckConfig Customization Editor consists of a set of four tabbed pages, one for each phase. Each phase page shows a list of all groups and their output settings in the selected phase. Note: If you have added a user-defined phase, it will appear as an extra tabbed page.

The list box has the following columns:

Field	Meaning
Group	The name of the check group. Which checks belong to each group is defined in the checkmap.checkmap file.
Check value	The value of the check. Values are: Warning - the message is issued as a warning Default - the message is issued using the default tool setting. This can be error, warning or off. Error - the message is issued as a warning Off - the message is not issued
The chosen value applies to all checks in the group.	
Description	A description of the types of checks in this group

To configure your output settings:

1. In the CheckConfig Customization Editor, click on the tab for the phase in which you will be carrying out checking.
2. Select the name of the group whose output settings you want to configure. In the messages list box, a list of all checks in the selected group is displayed.
3. Double -click on the name of the group.
The **Group <group name> dialog box** appears in which you can select the check value for the group.
4. Select the new check value and click OK.
The new value is applied in the CheckConfig Customization Editor.
5. Click OK to apply your change and leave the Editor, or click Apply to apply your changes and keep the window open.

View Customization, ViewObject '[name]' dialog box

In: [View Customization Editor](#),
[ViewObject '\[name\]' dialog box](#)

Specifies the object types visible in the view.

- Name - the name of this view
- Associations - The object sets available at the current level. An object set consists of one or more types of objects. The objects of the selected associations will show in the Types list box.
- Types - The object types shown by the view. If none of the types belonging to an association are selected, all are shown. If one or more are selected, only those are shown.
- ToolBarEntries - The predefined tool bar entries that are available at the current level. Only the icons of the selected entries will appear in the toolbar. Note that the entries with the Place in toolbar checkbox switched on in the [Menu Customization Editor Interface Page](#) will appear always in the toolbar.

View Customization, ViewObject '[name]' dialog box

In: [View Customization Editor](#),
[ViewObject '\[name\]' dialog box](#)

Specify the properties of the viewed objects and how they are represented. Only the properties in the Visible Properties dialog box are shown. You can drag the properties from one listbox to the other.

- Name - An enabled check button means the property is shown.
- Width - The width of the column.
- Sort Kind - The sort direction; none, increasing or decreasing.
- Sort Policy - The sort strategy; ascii, int or float.
- Sort Order - The order in which the objects are shown. The property with the lowest number is sorted first.

Double clicking on a property displays the **View Property dialog box**, which allows you to change several properties:

Width	The width of the column in characters.
Sort Kind	The type of sorting that is done. Options are: ascii, integer and float.
Sort Policy	The sort direction. Options are: none, increasing and decreasing.
Sort Order	The priority of the sort. The lowest number gives highest priority. The Sort Order setting is interpreted only if the Sort Policy is set to increasing or decreasing.

Sort Command Page, ViewObject '[name]' dialog box

In: [View Customization Editor](#),
[ViewObject '\[name\]'](#) dialog box

Allows you to insert or edit the sorting script for this view. No script means the default order is used. Normally a name of a Tcl procedure is inserted which takes care of the sorting. This procedure must be placed in a file that is always sourced. An appropriate place is the u_desk.tcl file in the users icase directory.

Storage Page, ViewObject '[name]' dialog box

In: View Customization Editor,
ViewObject '[name]' dialog box

Specifies for which browser levels the view is stored. This page corresponds with the scope column in the View customization editor. Depending on the browser level at which you are editing the view, one or more of the fields are grayed out.

- Project - This input field defines the project for which the view is valid. This field supports wild cards.
- Configuration - This input field defines the configuration for which the view is valid. This field supports wild cards.
- Phase - These four check buttons allow you to select a subset of phases in which the view is valid. If all phases are enabled this is represented by a * in the scope column of the View customization editor information area.
- System - These two check buttons allow you to one or two system types in which the view is valid. If both system types are enabled this is represented by a * in the scope column of the View customization editor information area.
- Read Only - Enabling this check button inhibits redefinition of this view at lower browser levels.

Menu Customization, CustMenu[button_type]Button '[name]', Arbiters page

In: [Menu Customization Editor](#)

Allows you to specify up to three arbiters for the button. The Arbiter page is available for:

- MenuBar buttons
- Cascading buttons

CustMenu[button_type]Button '[name]', Command page

In: [Menu Customization Editor](#)

Specifies the action taken if the button is pushed.

The Command page is available for:

- Push Buttons
- Check Buttons
- Radio Buttons

The options on the Command page are:

- **Command**
The command itself. You can select one of three options:
 - **Predefined Procedures**
A list of predefined actions that can be performed irrespective of the enabled object
 - **Object Operations**
A list of predefined actions that can be performed on the enabled object(s). Note: This option is only available for Browser menus, and not for Diagram Editor menus.
 - **Command**
custom made for the embedded Tcl interpreter.
- **Command kind**
Choose one of four command types:
 - **Internal**
command for the internal Tcl interpreter.
 - **External Output Only**
command using the [Monitoring Window](#) as output device.
 - **External Input/Output**
command using the [Execution Window](#) as output device.
 - **External Own Interface**
command using its own graphical user interface.
- **Interface**
Control various parts of the behavior of the command.

Menu Customization, CustMenu[button_type]Button '[name]', Enable/disable page

In: [Menu Customization Editor](#)

Controls the conditions on which the menu button is enabled.

The Enable/disable page is available for:

- Push buttons
- Check buttons
- Radio buttons

The options on the Enable/disable page differ for Browser menus and diagram Editor menus.

Browser menus:

- **disabled / enabled for classes**
Select one or more repository classes from this list box and press the Add-> button to enable them. The selected classes are moved from the *disabled...* to the *enabled...* list box. Similarly, you can disable enabled classes by selecting them in the *enabled...* list box and press the button <- Remove. Disabled classes are moved from the *enabled...* to the *disabled...* list box. Bear in mind that if the *enabled...* list box does not contain any entries, the menu object is enabled for *all* repository types.
- **disabled / enabled for browser types**
Some repository classes have more than one browser type attached to them. The repository type SystemVersion for instance, has the browser type SystemVersion and DocumentVersion attached. Using these list boxes you can specify for which browser type the menu object is enabled. Bear in mind that if the *enabled...* list box does not contain any entries, the menu object is enabled for *all* browser types attached to the selected repository type.
- **Selection count**
Controls the number of objects that have to be selected for the menu button to be enabled. The options are:
 - don 't care
 - 0
 - 1
 - many
- **Check enable on**
Specifies on what changes the enable is checked. Options are:
 - Nothing
 - Selection change
 - Level change
- **Script for extra checking**
This script is executed every time a change as selected with **Check enable on** occurs.

Diagram Editor menus:

A Diagram Editor has a set of basic operations that can be carried out depending on what is selected in the Editing area:

- Open
- Undo
- Move
- Copy
- Delete
- Replace
- Change
- Center

- Save
- Read
- Edit
- Item
- Check
- Diagram

Menu Customization, CustMenu[button_type]Button '[name]' dialog box, Interface page

In: [Menu Customization Editor](#)

Specifies the name and ways to use the menu button.

The Interface page is available for:

- MenuBar Buttons (Name, Mnemonic, Pinnable)
- Cascade Buttons (Name, Mnemonic, Pinnable)
- Push Buttons (Name, Mnemonic, Accelerator, Toolbar)
- Check Buttons (Name, Mnemonic, Accelerator, Toolbar, Initial State)
- Radio Buttons (Name, Mnemonic, Accelerator, Toolbar, Initial State, Radio Arbiter)

The options on the Interface page are listed below. Not all options are available to all menu objects.

- **Name** - (All)
The name of this view. As the mnemonic can only use characters appearing in the name, it is useful to have it editable here.
- **Mnemonic** - (All)
Determines which character in the name in combination with the meta key will be used as keyboard shortcut.
- **Hint Text** - (All; Windows only)
Specifies the text that appears in the message area if a user drags the mouse pointer over the menu (button).
- **Pinnable** - (MenuBarButton and CascadeButton; Unix only)
Specifies if the menu can be torn off.
- **Accelerator** - (PushButton, CheckButton and RadioButton)
This defines the accelerator key combination for the menu button.
- **ToolBar** - (PushButton, CheckButton and RadioButton)
Enabling the check button *Place in toolbar* will put the selected icon in the tool bar. Otherwise it can be selected per view in the [View Customization Editor Interface Page](#).
With the button *Select Icon* you can select the icon from the [Icon Selection dialog box](#).
- **Popup menu** - (PushButton, CheckButton and RadioButton)
This lets you place a menu option in a context-sensitive popup menu that is selected in the [information area](#) of the [Browser](#) by holding down the right mouse button.
- **Initial State** - (CheckButton and RadioButton)
(Check buttons and Radio Buttons only) Determines the default state of the button.
- **Radio Arbiter** - (RadioButton)
(Radio buttons only) Here you can select from the arbiters defined for the corresponding menubar button. An arbiter allows only one of the radiobuttons assigned to it to be enabled.

Editing Menu Object Properties - dialog box

(In: [Customization Editor](#))

Storage page

Specifies for which [browser levels](#) the menu is stored. Depending on the level at which you are editing the menu, one or more of the fields are grayed out. The storage page is available for all browser menu objects.

- **Project**
This input field defines the project for which the menu is valid. This field supports wild cards.
- **Configuration**
This input field defines the configuration for which the menu is valid. This field supports wild cards.
- **Phase**
These four check buttons allow you to select a subset of phases in which the menu is valid.
- **System**
These two check buttons allow you to specify if the menu object applies to [systems](#) only, [documents](#) only or to both
- **Read Only**
Enabling this check button makes redefinition of this menu at lower levels impossible.
- **Visible**
These check buttons allow you to specify on which browser level the menu object must be (in)visible.

Editing Objecttype Properties - dialog box

(In: [Customization Editor](#))

Command page

Specifies the action taken if the button is pushed.

- Command - The command itself. You can select one of three options:
 - Predefined Procedures
 - Object Operations
 - Command - custom made for the embedded Tcl interpreter.
- Command kind - Choose one of four command types:
 - Internal - command for the internal Tcl interpreter.
 - External Output Only - command using the [Monitoring Window](#) as output device.
 - External Input/Output - command using the [Execution Window](#) as output device.
 - External Own Interface - command using its own graphical user interface.
- Interface - Control various parts of the behavior of the command.

Editing Objecttype Properties - dialog box

(In: [Customization Editor](#))

Interface page

The Interface page lets you select icons and a filename extension for storage of the object in the filesystem.

Editing Objecttype Properties - dialog box

(In: [Customization Editor](#))

Storage page

Specifies for which browser levels the objecttype is stored. This page corresponds with the scope column in the Objecttype customization editor. Depending on the browser level at which you are editing the objecttype, one or more of the fields are grayed out.

- Project - This input field defines the project for which the objecttype is valid. This field supports wild cards.
- Configuration - This input field defines the configuration for which the objecttype is valid. This field supports wild cards.
- Phase - These four check buttons allow you to select a subset of phases in which the objecttype is valid.
- System - These two check buttons allow you to one or two system types in which the objecttype is valid.
- Read Only - Enabling this check button inhibits redefinition of this objecttype at lower browser levels.

Editing Menu Object Properties

(In: Customization Editor)

Use **Edit | Edit Properties** in the Menu Customization Editor to specify a menu button further. The Edit Properties dialog box appears.

Editing Objecttype Properties

(In: Customization Editor)

Use **Edit | Edit Properties** to create new user interface objects.

You can only edit the properties of objects defined on the current browser level. The ones that are have a trailing asterisk (*) in their *Level* indication.

The Edit Properties dialog box appears

Editing Navigation Strategy Definition Properties

(In: Customization Editor)

Use **Edit | Edit Properties** to specify properties for a navigation strategy. The Edit Properties dialog box appears.

You can only edit the properties of objects defined on the current browser level. The ones that are have a trailing asterisk (*) in their *Level* indication.

Editing Navigation Strategy Location Properties

(In: Customization Editor)

Use **Edit > Edit Properties** to specify properties for a navigation strategy location. The Edit Properties dialog box appears.

You can only edit the properties of objects defined on the current browser level. The ones that are have a trailing asterisk (*) in their *Level* indication.

Editing Property Definition Properties

(In: Customization Editor)

Use **Edit | Edit Properties** to specify properties for a property definition. The Edit Properties dialog box appears.

You can only edit the properties of objects defined on the current browser level. The ones that are have a trailing asterisk (*) in their *Level* indication.

Editing Property Location Properties

(In: [Customization Editors](#))

Use **Edit | Edit Properties** to specify properties for a property location. The [Edit Properties dialog box](#) appears.

You can only edit the properties of objects defined on the current [browser level](#). The ones that are have a trailing asterisk (*) in their *Level* indication.

Editing View Properties

(In: [Customization Editors](#))

Use **Edit | Edit Properties** in the View Customization Editor to edit the properties of (new) views. The [Edit Properties dialog box](#) appears.

You can only edit the properties of objects defined on the current [browser level](#). The ones that are have a trailing asterisk (*) in their *Level* indication.

Module Availability Customization, Select Module (New Module) dialog box

In: Customization Editors

In this dialog box you can select the module you want to install.

Select the button Show Supporting Modules to see the supporting modules.

Buttons :

Browse	Press the Browse button to select a different Modules directory. The <u>Module Select Directory</u> dialog box appears
OK	Press OK to accept the changes and leave the Select Modules dialog box.
Cancel	Press Cancel to discard the changes and leave the Select Modules dialog box.

Module Availability - Select Module (Change Location) dialog box

In: Customization Editors

In this dialog box you can select a module of which you want to change its location.

Buttons :

Browse	Press the Browse button to select a different Modules directory. The <u>Module Select Directory</u> dialog box appears
OK	Press OK to accept the changes and leave the Select Modules dialog box.
Cancel	Press Cancel to discard the changes and leave the Select Modules dialog box.

Group <group name > dialog box

The Group <group name> dialog box is used to modify the message type or check value of a group of check messages. The box contains the group description and a option button for the check value. The options are:

- **warning** - The messages in the group are presented as warnings
- **default** - The messages in the group are presented as their default type
- **error** - The messages in the group are presented as errors
- **off** - The messages in the group are not presented

Checkconfig Customization Editor

In: Customization Editors

The Checkconfig Customization Editor is started by creating a checkconfig.checkconfig. file.

To create a checkconfig.checkconfig file:

1. In the Browser, open the pseudo-object <customization files> on any browser level or the <user customization files> object.
2. Select File | New | Customization File Version on any browser level, or File | External File on <user customization files> level.
3. In the dialog box, select checkconfig.checkconfig and click Edit.
The Check Config Customization Editor appears.

Customization Editors Tool Bar

The *tool bar* is a window area that contains icons for quick activation of frequently used menu entries. Instead of executing such a menu entry from the menu bar, you can just click on the corresponding icon in the tool bar.

The tool bar is located between the menu bar and the context area. You can find out which menu entry a toolbar button represents by dragging your mouse pointer over the button in the toolbar and keeping it there for a few seconds. A ToolTip with the name of the menu entry appears.

The tool bar in the Customization Editor contains the following buttons by default:

- File > Save
- Edit > Redefine
- File > Info
- Edit > Delete

Control Panel Customization Editor

In: [Customization Editors](#)

The Control Panel Customization Editor is started by creating a *.pnl file.

To create a *.pnl file:

- 1.
2. In the Browser, open the pseudo-object <customization files> on Project, Configuration or Phase level, or the <user customization files> object.
3. Select File | New | Customization File Version.
4. In the dialog box, select a *.pnl file and click Edit.
The [Control Panel Customization Editor](#) appears.

Select Module Dialog box

In: Customization Editors

In this dialog box you can select a required module you want to add to an existing module.

Buttons :

- | | |
|--------|--|
| Browse | Press the Browse button to select a different Modules directory.
The <u>Module Select Directory</u> dialog box appears |
| OK | Press OK to accept the changes and leave the Select Modules dialog box. |
| Cancel | Press Cancel to discard the changes and leave the Select Modules dialog box. |

Select Panel Entry - dialog box

In this dialog box you can select a new Panel Entry from the list.

Press OK to confirm your choice and leave this box, press Cancel to leave this box without changes.

Browser - document level

The browser on *document* level can be used for creating and formatting project documentation. From here, you can:

- navigate the repository using the navigation area
- execute actions on objects available on this level using menus

Note : Your access rights determine which operations you are allowed to perform on which objects.

Window sections

The default browser window contains the following areas. You can use the *View* menu to hide and display the Context Area, Tool Bar and Message Area.

- Menu Bar
- Tool Bar
- Context Area
- Navigation Area
- Information Area
- **Message Area** - Displays system messages. In Unix, you can browse through the message history with the arrow-up and arrow-down symbols at the right.

Edit Document Structure dialog box

Section

- New...* Use this button to add a file section, a local section or a property section to the document structure as defined in the Document Structure Matrix. You can *select* sections that already exist in the repository. You can *enter* the name of new sections. Sections that didn't exist in the information area before initially get the type *None*.
- Edit* Use this button to change the parent section of the selected section.
- Delete* Use this button to delete the selected section from the document structure. Deleting a section doesn't delete it from the browser. It is still there in the browser, so you could add it to the structure again by using the *New* button.

List box

The list box displays the document structure. This structure is reflected in the information area of the browser on Document level. You select file or local sections from here that you want to edit.

Save / OK

Use these buttons to save the changes made in the structure to the information area of the browser. With *OK* the dialog box is closed, with *Save* it remains displayed.

New Document dialog box

Document Name: Enter the name of the new document here.

Documented System:

Enter the name of the system you want to create the document for. This has to be an existing system.

Editor : Select the word processor or Desk Top Publishing package you are using to create the new document:

UNIX :

- **fm40**: FrameMaker 4.0
- **fm50** : FrameMaker 5.0
- **il60** : Interleaf 6.0

Windows :

- **word**: Microsoft Word for Windows 7.0

Press the *OK* button to confirm the information in the dialog box. Press the *Cancel* button to discard the operation.

New Local Section dialog box

Local Section Name:

Enter the name of the new local section here.

Local Section Type:

The file types listed here are supported by the specified word processor or Desk Top Publishing package you are using for your document. Select the file type of your choice.

Press the *OK* button to confirm the new object. Press the *Cancel* button to discard the operation. Press the *Edit* button to create the new object and then open it for editing.

New Property Section dialog box

Use this dialog box to create or define a new property section . You can create file property sections (*Fileprop*) and item property sections (*Itemprop*) in this box.

Property Section Name:

Fill in the name of the new property section. If you are *defining* a property section, this name is already known.

Files

This list box displays all the file versions of the documented system. Select the one of your choice. If you create an item property section, the item has to be defined in this file.

Items

This list box displays all the items that are defined in the selected file. Skip this list box if you are creating or defining a file item property. In case of a item property section, select the item of your choice here.

Properties

This list box displays all the properties of either one of the following:

- the selected file: if you have only selected a file
- the selected item: if you have selected a file and an item

Press the *OK* button to confirm the information in the dialog box or the *Cancel* button to discard the operation.

Defining a Local, File or Property Section

(In: Browser on document level)

When you create a new section in the Edit Document Structure dialog box, ObjectTeam creates a section with type *None*. Use **File > Define** to define the *Type* of the section, which appears in the Information area on Document level.

You can define the following types of sections:

- file section using the File Section dialog box
- local section using the Local Section dialog box
- property section using the Property Section dialog box

Generating the structure and contents of a local section

1. Make sure the browser is on Document level.
2. Select a local section in the information area of the browser.
3. Select **File | Generate | Structure** or **File | Generate | Structure and Contents**.

If you select **Structure**, the file sections or property sections specified for the selected local section are exported from the repository, converted to a graphic file format supported by your DTP-package and copied to the user environment.

If you select **Structure and Contents**, the local section itself is generated as well.

You can specify contents generators and structure generators for a local section by editing its properties. After the structure of a local section is generated, the information area is updated with the appropriate file sections or property sections. These are grouped hierarchically under the selected local section.

Generating a default Document

1. Make sure the browser is on Phase level.
2. Select File > New > Document Version.
3. Complete the New Document Version dialog box.
The new object of type *DocumentVersion* is created on Phase level. If you are unable to create a new document, you may not have the necessary access rights.
4. Select the new document object in the information area.
5. Select File > Generate > Document.
6. Complete the Generate Document dialog box.
The default document is generated. Reports on document generation are displayed in an Execution Window. When the generation has finished, the following message appears in the Execution Window:

```
Generating document finished  
Done
```

```
-> (E)xit (P)revent Reuse
```

7. Open the generated document in the information area to see its generated structure.
8. From the information area, edit a local section to start the configured word processor and refine the generated draft document.

Previewing a File Section

(In: Browser on document level)

Use **File > Preview** to view file sections with a preselected file viewer. You can use it with objects of the following type:

- none
- **DoctextSection** : Plain ASCII text
- **EpsfSection** : Encapsulated PostScript with TIFF preview
- **EpsSection** : Encapsulated PostScript without preview
- **PropertySection** : Plain ASCII text
- **PsSection** : PostScript
- **EpsiSection** : Encapsulated PostScript with bitmap preview

You can change the current tool that is used for showing objects with Options > Previewer....

Updating Document Directory

(In: Browser on Document level)

Use **File > Update Document Directory** to force an update of the selected document (object). You can update the entire document by selecting it in the information area on phase level or one or more document objects by selecting them in the information area on document level.

If you update the document directory, for instance for a local section which contains one or more file sections, the diagrams the file section(s) refer to are exported again from the repository.

A report of the updating process appears in a Monitoring Window.

Menu Bar - Browser (document level)

Below, the default menus of the browser on document level are listed. You can add new menus and redefine existing ones by using the Customization Editor.

[File] [Edit] [View] [Options] [Version] [Security] [Utilities] [Help]

- **File** menu
 - New
 - Define
 - File section...
 - Local section...
 - Property section...

 - Delete
 - Close
 - Edit
 - Open
 - Show
 - Generate
 - Structure
 - Structure and Contents

 - Preview
 - Uppdate Document Directory
 - Change
 - Link Status...

 - Properties
 - Edit ...
 - Delete ...
 - Show ...

 - Intro ...
 - Print
 - Print View
 - Exit
- **Edit** menu
 - Undo
 - Cut
 - Copy
 - Paste
 - Select All
 - Deselect All
- **View** menu
 - Refresh
 - ToolBar
 - ContextArea
 - MessageArea
 - Large Icons

- Small Icons
- Details
- Filter
 - Edit ...
 - Delete
- Default
- Pseudo
- **Options** menu
 - Compare Command...
 - Editor ...
 - Font ...
 - Previewer ...
 - Printer Setup
 - Edit ...
 - Delete ...
 - Show ...
 - Viewer ...
 - Copy User Environment
 - To Corporate Environment...
 - To Project Environment...
 - To ConfigVersion Environment...
 - To PhaseVersion Environment...
 - To SystemVersion Environment...
- **Version** menu
 - Freeze ...
 - Unfreeze
 - New
 - Copy ...
 - Delete ...
 - Select
 - New ...
 - Selected ...
 - Deselect
 - Compare ...
 - Make Fixed
 - Select Fixed...
 - Make Selected...
 - Make Current
 - Make Snapshot
- **Security** menu
 - Show Access Rights...
 - Effective Roles...
 - Activate Role...
 - Deactivate Role...
 - Role Rights
 - Edit

- Show
- **Utilities** menu
 - Clone
 - Class Browser
 - Monitoring Window
 - Execution Window...
 - Delete Unreferenced Items...
 - Reports
- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

DocumentVersion - browser object

The browser object DocumentVersion appears in the information area on:

- Phase level

A Document is used to generate project documentation. It consists of a structured set of:

- Files with a format the specified word processor or Desktop Publishing package understands.
- Links to diagrams from the productivity environment.

The browser object DocumentVersion is actually a special type of SystemVersion. However, the possible child objects that a DocumentVersion object can have are different to those of a SystemVersion object. The child objects of a DocumentVersion are all related to the documentation facilities of ObjectTeam:

- Document Structure Matrix - represents the structure of the document
- Local Section - represents a type of document that is compatible with the specified word processor or DTP-package
- File Section - represents a link to a graphic file or a text file from the ObjectTeam repository
- Property Section - represents a link to one or more item or file properties from the ObjectTeam repository

The browser object DocumentVersion is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- DocumentVersion
        |
        +- {dsm| file section| property section| local section}
```


Document Structure Matrix - browser object

The Document Structure Matrix (dsm) is used in the browser on Document level. It is used to define the structure of the document. The dsm contains information as to which local sections, file sections and property sections are part of the document, and how the hierarchical structure of these sections is defined.

You can edit the structure of your document in the Edit Document Structure dialog box.

The browser object DocumentVersion is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- DocumentVersion
        |
        +- {dsm | file section | property section | local section}
```


File Section

A file section is a link to a diagram in the repository. It is used in the browser on Document level. When a document containing such a reference is generated, the referred diagram is exported to the appropriate format and incorporated in the document.

You can choose to make a File Section **fixed** or **selected** by using the following menu entries:

- Version > Make Fixed
- Version > Select Fixed...
- Version > Make Selected

The status *fixed* implies a fixed link to a specific version of the diagram, whereas the status *selected* implies a link to the version that is currently the *working* version.

The type of a file section can be:

- diagram type
- type of a (generated) source file

A file section is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- DocumentVersion
        |
        +- {dsm| file section | property section| local section}
```


Local Section

A local section is used in the browser on Document level.

The main purpose of local sections is to provide a framework in which file sections can be placed. Local sections refer to files that are supported by the word processor or DTP the document is created for. For instance:

- Documents in the proprietary word processor format
- Specific graphic formats
- ASCII files

What the available section types for a local section are, depends on the defined word processor or DTP:

Book FrameMaker book file

Cat Interleaf Catalog File

Data Data file

Doc

- FrameMaker document
- Interleaf Document
- MS Word Document

DocText ASCII text file

Eps Encapsulated PostScript file without preview image

Epsf Encapsulated PostScript file with TIFF preview image

Epsi Encapsulated PostScript file with bitmap preview image

Mif File in FrameMaker MIF format

Mml File in FrameMaker MML format

Ps PostScript file

Str Interleaf Structure File

Toc

- FrameMaker table of contents
- Interleaf table of contents

Ximage Graphical file in Ximage format

Local sections are part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- DocumentVersion
        |
        +- {dsm| file section| property section| local section}
```


Property Section

A property section is a link to one or more property values defined for a file or an item in the repository.

A property section is used in the browser on Document level. When a document containing such a reference is generated, the appropriate property values are retrieved from the repository and incorporated in the document.

You can create or define a property section in the Property Section dialog box.

The type of a property section can be:

- Fileprop : referring to file properties
- Itemprop : referring to item properties

A property section is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- DocumentVersion
        |
        +- {dsm| file_section| property section | local_section}
```


Creating a new Document - task

1. Make sure the browser is on Phase level.
2. Select File | New | Document Version.
3. Complete the New Document Version dialog box.
The new object of type DocumentVersion is created on Phase level.

If you are unable to create a new document, you may not have the necessary access rights.

Defining the structure of a Document manually

When you use a Document Structure File, the document structure is automatically defined according to this structure file. Every line in a Document Structure File represents a local section. Every column represents control information for the local sections.

However, if you want to define the document structure manually, you do the following:

1. Make sure the browser is on Document level.
2. Select File > New > Document Structure Matrix.
The new browser object Document Structure Matrix (dsm) is created. If you are unable to create a new dsm, you may not have the necessary access rights.

To add a local section, file section or property section to the Document Structure Matrix, do the following:

1. Double -click on the Document Structure Matrix in the information area, or select it and then select **File > Edit**.
2. Use the Document Structure dialog box to:
 - add new sections to the structure
 - change the position of sections within the structure
 - delete sections from the structure
3. Press the **Apply** or the **OK** button to confirm your changes to the structure.

The new structure of the document is reflected in the information area on Document level. You can generate the structure and the contents of every local section using File > Generate on Document level.

Creating a Section

To create a local section, a file section or a property section, do the following:

1. Make sure the browser is on Document level.
2. Select **File > New > Local/File/Property Section(s)**.
3. Enter the required information in the New Local Section dialog box, the New File Section dialog box or the New Property Section dialog box.
The new section object is created on Document level.
4. Add the section to the document structure.

If you are unable to create a new section, you may not have the necessary access rights.

Generate Document dialog box

In: Document Generator (Generate Document)

Document Structure File:

Select a Document Structure Definition file from here. These definition files contain information used for the generation of default documents. They consist of structured sets of local sections and a set of commands to create file sections for these local sections.

By default, the following files can be selected:

- < M4_home>/etc/<editor>_mod.str
This file is suitable for the documentation of a system within the following phases:
 - Analysis
 - System Design
 - Object Design
- < M4_home>/etc/<editor>_imp.str
This file is suitable for the documentation of a system within the phase:
 - Implementation

You can also use your own Document Structure Definition File. If you want to do that, you must first define your definition file as value for the Document property Document Structure File. The next time you select File > Generate... for the document, your definition file will be part of the list of Document Structure Definition Files in the Generate dialog box.

Press the *OK* button to confirm the selected Document Structure Definition file. Press the *Cancel* button to discard the operation.

Help Tool - dialog box

In: [Help Tool](#)

Current Page The current page contains one of the following items:

- [Contents](#)
- [Index](#)

Within the Help Tool dialog box, you can flip the pages by:

- Selecting another page tab
- Select another page number

Page Tab You can switch between the **Contents** and the **Index** page by selecting the appropriate page tab from here.

Page number You can switch between the **Contents** and the **Index** page by clicking on the arrow pointing right or left.

OK If you press this button, the topic that is currently selected in the hierarchical tree is loaded in the help tool. The dialog box is dismissed.

Apply Now This has the same effect as the **OK** button. The only difference is that the dialog box remains displayed.

Cancel If you press this button, the dialog box is dismissed without anything happening to the contents of the help tool.

Not Saved dialog box

This dialog box reminds you of the fact that the current diagram (or other object) is not saved. It offers you the following options for the question:

Save changes to <Object> first ?

Yes Saves the object before the requested action is carried out.

No Doesn't save the object. If the operation you selected requires a saved object, it carries out the requested action on the object as it was last saved.

Cancel Discards the operation without saving the object.

New Dialog box

Name Enter the name of the new object here.

Buttons:

OK Press the *OK* button to confirm the *Name* of the new object.

Cancel Press the *Cancel* button to discard the operation.

Edit Press the *Edit* button to create and then edit the object.

Delete Object dialog box

Press the *OK* button to confirm the removal of the specified object(s). Do this only if you are very sure!
Press the *Cancel* button to discard the operation.

Delete Properties dialog box

Select the object of which you want to delete the properties and press the *OK* button to confirm the removal.
Press the *Cancel* button to discard the operation.

Select Object dialog box

You select the object of your choice in the list box of this dialog box. You can select only **one** object out of all the objects listed here.

Press the *OK* button to confirm the current selection or the *Cancel* button to discard the operation.

Select Object dialog box

The objects listed in the list box are the objects you can select.

You can (de)select a discontinuous range of objects by pressing the CTRL key and the left mouse button simultaneously on the objects of your choice. To (de)select a continuous range of objects, select the first file in the usual way and the last file while pressing the SHIFT key.

Press the *OK* button to confirm the current selection or the *Cancel* button to discard the operation.

Opening a second Browser

In: Browser

Use **Utilities | Clone** to open a second copy of the browser.

This way you can see browser objects at two levels simultaneously.

Note that changes made in one copy are not updated automatically in the other copy. Use View | Refresh to refresh the navigation area and the information area of a browser.

Specifying the default text editor

Use **Options | Editor** to specify the default ASCII editor used for editing text files such as:

- some customization files
- generated source code files

Exiting a tool

[[Browser](#)] [[Diagram Editor](#)] [[Monitoring Window](#)] [[Customization Editor](#)] [[Class Browser](#)] [[Help Tool](#)]

Use **File | Exit** to quit the current tool.

This menu entry can be used to quit the following tools:

Browser By using **File | Exit** in the browser on any level you quit ObjectTeam. As long as there are still Monitoring Windows or Execution Windows active that were started from the browser, you cannot exit. Editors and Class Browsers are not dependent on the browser, so they do not prevent you from exiting the browser.

Diagram editor If you try to exit a diagram editor without having saved the diagram, a message box appears. This box gives you the option to save the diagram before exiting, to exit without saving the diagram, or to cancel the exit action.

Monitoring Window You can only exit a Monitoring Window if there are no commands active.

Customization Editor If you try to exit a diagram editor without having saved the customization file, a message box appears. This box gives you the option to save the file before exiting, to exit without saving or to cancel the exit action.

Class Browser Exiting from the Class Browser is always possible, as it does not modify any data.

Help Tool Exiting from the Help tool is always possible.

Opening files in the Help Tool (Unix only)

In: [Help Tool](#)

Use File | Open File to open an ObjectTeam help file in the [help tool](#). You specify the file in a File Selection dialog box.

The default ObjectTeam help files are stored in the directory <M4_home>/help. These files are text files with certain tags added to them. The tags are interpreted by the the help tool.

The set of tags that is used is a subset of the *HyperText Markup Language* (HTML). The following tags are supported:

```
<!--... -->
<A HREF="...">...</A>
<A NAME="...">...</A>
<A HREF="..." NAME="...">...</A>
<B>...</B>
<BR>
<DD>
<DL>...</DL>
<DT>
<Hn>...</Hn>
<I>...</I>
<LI>...</LI>
<OL>...</OL>
<P>
<PRE>...</PRE>
<TITLE>...</TITLE>
<TT>...</TT>
<UL>...</UL>
```

For more details on these tags, refer to HTML documentation.

Setting Printer Options

Use **Options | Printer Setup** to specify the following:

- Options | Printer Setup | Graphical...
to specify the Graphical printer settings (printer command, page width, page height)
- Options | Printer Setup | Text...
to specify the Text printer settings (printer command, line length, page length)

The Printer Setup dialog box appears

(De)Selecting all objects in the information area

In: Browser

Use **Edit | Select All** to select all (visible) objects in the information area. Use **Edit | Deselect All** to deselect all currently selected objects.

Starting an Execution Window

Use **Utilities | Execution Window** to execute a (system) command in an Execution Window, i.e. an xterm on UNIX platforms or a DOS box on Windows platforms.

The Execution Window Command(s) dialog box appears.

Menu Bar

In: [Help Tool \(Unix only\)](#)

Below , the default menus of the [help tool](#) are listed. You can add new menus and redefine existing ones by using the [Customization Editor](#).

[[File](#)] [[Navigate](#)] [[View](#)] [[Options](#)] [[Help](#)]

- **File** menu
 - [Open File](#)
 - Print
 - [Exit](#)
- **Navigate** menu
 - [Contents ...](#)
 - [Index ...](#)
 - Previous Entry
- **View** menu
 - ToolBar
 - MessageArea
- **Options** menu
 - [Printer Setup...](#)
- **Help** menu
 - About ...

Table of Contents

In: [Help Tool](#)

- Selecting **Help | Help Topics** from any window in ObjectTeam
- Selecting **Navigate | Contents...** from the [help tool](#)
- Press the **TOC** button in the tool bar of the help tool
- Selecting the *Contents* tab in the [Help Tool dialog box](#)
- Browsing back from page 2 to page 1 in the Help Tool dialog box

The Table of Contents in the help tool consists of an hierarchical tree of objects which can be considered as chapters, sections, subsections and so on. The child objects of an object in this tree can be made visible by unfolding the parent object. They can be hidden again by folding the parent object. You can unfold an object by clicking on the preceding arrow pointing right. You can fold one by clicking on the preceding arrow pointing down.

Objects at the bottom of a branch (leaf objects) don't have a preceding arrow and cannot be unfolded. As they represent the actual help topic, you can open them to make the corresponding help topic appear in the drawing area of the help tool.

You can use the buttons **OK** or **Apply Now** in the [Help Tool dialog box](#) to confirm your selection.

Index

In: [Help Tool](#)

You can invoke the Index of Help topics by:

- Selecting **Navigate | Index...** from the [help tool](#)
- Press the **Index** button in the tool bar of the help tool
- Selecting the *Index* tab in the [Help Tool dialog box](#)
- Browsing from page 1 to page 2 in the Help Tool dialog box

The index is an alphabetical list of help topics with first level entries and second level entries. You can navigate this list by:

- Using the scroll bar at the right of the list
- Using the name search option
You can enter the first letters of the topic you are looking for in the field that is positioned above the actual index. The first topic in the list that comes the closest to the entered string is selected.

Some of the first level entries are suffixed by:

(-)

This indicates that there is no help topic available for this particular entry. However, for all its second level topics there is.

You can use the buttons **OK** or **Apply Now** in the [Help Tool dialog box](#) to confirm your selection.

Help Tool

This topic is only applicable to UNIX-based platform. On Windows-based platforms, the Windows help system is used to offer on-line help topics.

You can use the Help tool to find information about a certain topic of ObjectTeam. You can start the help tool from every window using **Help > What's This?** or **Help > Help Topics**.

What 's This? initially offers information about the current context. Depending on the location and the situation, it gives you:

- General information on the current tool:
 - browser : there is a help topic for every browser level
 - diagram editor
 - customization editor
 - Monitoring Window
 - class browser
 - repository tool
 - help tool
- Specific information on the object that is selected in the current tool:
 - **browser** : if a browser object is selected in the information area, help information is offered about that particular object
 - **diagram editor**: if a diagram object is selected in the drawing area, help information is offered about that particular object

When the help tool is activated, there are several ways to find information about ObjectTeam:

- You can ask again for context sensitive help using the menu entry **Help | What's This?** in the tool you want information about.
- You can select a topic from the (alphabetically sorted) Help Index.
- You can use the Table of Contents.
- You can activate a new help topic by clicking on a hyperlink in the displayed help topic. Every hyperlink is underlined. To activate a hyperlink, drag the mouse cursor over to an underlined word or a set of underlined words and click the left mouse button when the mouse pointer changes into a little hand. The new help topic is displayed. You can return to the previous topic by pressing the **Back** button.

Window sections

The default help tool window contains the following areas:

- Menu Bar - You select menu entries from here.
- Tool Bar - You can select commonly used menu items here.
- **Display Area** - The help actual help information is displayed here.
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Help Tool Bar

In: Help (UNIX-based platforms only)

The *tool bar* is a window area that contains icons for quick activation of frequently used menu entries. Instead of executing such a menu entry from the menu bar, you can just click on the corresponding icon in the tool bar.

The tool bar is located between the menu bar and the context area. You can find out which menu entry a toolbar button represents by dragging your mouse pointer over the button in the toolbar and keeping it there for a few seconds. A ToolTip with the name of the menu entry appears.

The tool bar in the help tool contains the following buttons by default:

- Navigate | Contents...
- Navigate | Index...
- Navigate | Previous Entry

You can change the buttons in the tool bar by using the Customization Editor.

The following tools also have tool bars:

[Browser]

[Diagram Editor]

[Customization Editor]

[Class Browser]

Text editor dialog box

Text Editor Enter the name of the preferred Ascii editor here. Keep in mind that you define this editor only for the file type currently selected in the *Context* field. The default Ascii editor is defined in the M4_variable **M4_editor**.

Text Window Editor Switch this option *on* if the selected text editor creates its own window. If it doesn't, switch this button *off*.

Context Select the file type here for which the currently specified *Ascii Editor Command* must be used. Keep in mind that a separate editor has to be specified for every *Context*. You can select the following file types:

- none
- **DocTextSection** : Plain ASCII text

Printer Setup dialog box

There are two kinds of Printer Setup:

- [Graphical Printer Setup](#)
- [Text Printer Setup](#)

Graphical Printer:

(Unix-based platforms only)

You can enter a new value for any of the fields listed here if you do not want to use a default value.

Printer Command E.g.:

```
lpr -Pps
```

Page Width The default value is retrieved from the M4 variable **M4_ps_printer**.
Page Height The default value in inches is retrieved from the M4 variable **M4_ps_page_w**.
 The default value in inches is retrieved from the M4 variable **M4_ps_page_h**.

Text Printer:

You can enter a new value for any of the fields listed here if you do not want to use a default value.

Printer Command E.g. for Unix-based platforms:

```
lpr -Pline
```

E.g. for Windows-based platforms:

```
notepad.exe /p
```

Line Length The default value is retrieved from the M4 variable **M4_a_printer**.
Page Length The number of characters on a line. The default is retrieved from **M4_a_printer_llen**.
 The number of lines on a page. The default is retrieved from **M4_a_printer_plen**.

Reports On Actors...

In: Browser and Use Case Diagram Editor

Report on actors is available on two Browser levels:

- phase version
- system version

The report includes, for each actor, the communication associations to or from the actor.

To create a report on Actors:

1. Select Utilities | Reports | On Actors...
The Report Options dialog box for Actors appears.
2. Edit the dialog box.
3. Press OK to start the Report
The results of the generated Report can be viewed in the Monitoring Window.

Reports On CDMs

In: Browser

Report on CDMs is available on two Browser levels:

- phase version
- system version

The report includes:

- CDM name
- Diagrams in which the associated class appears
- System version that contains the CDM

To create a report on CDMs:

1. Select Utilities | Reports | On CDMs.
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Classes

In: [Browser](#) and [Class Diagram Editor](#)

Report on classes is available on two Browser levels:

- [phase version](#)
- [system version](#)

The report includes:

- Class names
- Class properties
- Attributes
- Operations

To create a report on Classes:

1. Select Utilities | Reports | On Classes
The results of the generated Report can be viewed in the [Monitoring Window](#).

Reports On Class Generalizations

In: [Browser](#), and [Class Diagram Editor](#)

Report on class generalizations is available on two Browser levels:

- [phase version](#)
- [system version](#)

The report includes:

- Superclass name and type
- Generalization type
- Subclass name and type

To create a report on CDMs:

1. Select Utilities | Reports | On Class Generalizations.
The results of the generated Report can be viewed in the [Monitoring Window](#).

Reports On Communications

In: Browser and Collaboration Diagram Editor

Report on Collaborations is available on two Browser levels:

- phase version
- system version

The report lists all objects of the Collaboration Diagram with the messages they send and receive.

To create a report on Communications:

1. Select Utilities | Reports | On Communications.
The results of the generated Report can be viewed in the Monitoring Window.

Report On Components and Properties...

In : Browser and all Diagram Editors

Report on components and properties is available only at the system level.

The report includes:

- Component name and type
- Property names and values

To create a report on Components and Properties:

1. Select **Utilities | Reports | On Components and Properties...**
The Report Options dialog box for Properties appears.
2. Fill in the dialog box.
3. Press OK to start the Report
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Corporate Groups

In : Browser

Report on corporate groups is available only on Corporate level.

The report includes:

- The corporate group name
- Saved group name and version
- Where the corporate group is referenced

To create a report on Corporate Groups:

1. Select Utilities | Reports | On Corporate Groups.
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Documents

In : [Browser](#)

Reports on [Documents](#) are available on four levels:

Corporate level All selected document versions are listed per selected [project](#), [configuration](#) and [phase version](#).

Project level All selected document versions in the project are listed per configuration version and phase version.

Configuration level All selected document versions are listed per selected phase version.

Phase level All selected document versions are listed.

The report includes:

- The selected [version](#) of the document
- The status of the selected document version
- The [link status](#) of the selected document version

To create a report on Documents:

1. Select Utilities | Reports | On Documents.
The results of the generated Report can be viewed in the [Monitoring Window](#).

Reports On Files

Reports on files is available on all browser levels. Only files defined within the current level are reported. Files include:

- All diagram types
- Document Structure Matrices
- Document Sections (local sections, file sections, property sections)
- User -defined Tcl files
- (Generated) source files

The report includes:

- The name of the selected file version
- The type of the selected file version
- The selected version of the file
- The status of the selected file version
- The link status of the selected file version

Reports On Groups

In : Browser

Report on groups is available on all browser levels.

Only corporate groups and groups defined within the current level are reported. The group versions are listed per selected project, configuration and phase version.

The report includes:

- The name of the selected group version
- The selected version of the group version
- The status of the selected group version
- The link status of the selected group version

To create a report on Groups:

1. Select Utilities | Reports | On Groups.
The results of the generated Report can be viewed in the Monitoring Window.

Report On Items and Properties...

In: Browser and all Diagram Editors

Reports on items and properties is available on all browser levels.

The report includes:

- Item name and type
- Property names and values

To create a report on Components and Properties:

1. Select Utilities | Reports | On Items and Properties...
The Report Options dialog box for Properties appears.
2. Fill in the dialog box.
3. Press OK to start the Report
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Phases

In : [Browser](#)

Report on [phases](#) is available on three levels:

Corporate level All phases are listed per [project](#).

Project level All phases in the project are listed per [configuration version](#).

Configuration level

All selected phase versions in the configuration version are listed.

The report includes:

- The type of the phase version
- The selected [version](#) of the phase
- The status of the selected phase version
- The [link status](#) of the selected phase version

To create a report on Phases:

1. Select Utilities | Reports | On Phases.
The results of the generated Report can be viewed in the [Monitoring Window](#).

Reports On Projects

In : Browser

Report on CDMs is available on Corporate level. All projects in the corporate environment are listed.

The report includes:

- The configurations in the projects
- The selected versions of the configurations
- The status of the configurations

To create a report on Projects:

1. Select Utilities | Reports | On Projects.
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Saved Groups

In : Browser

Report on saved groups is available on all browser levels.

The report includes:

- The name of the saved group
- The corporate group name, if promoted
- The contents
- The creator and time created

To create a report on Saved Groups:

1. Select Utilities | Reports | On Saved Groups.
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Systems

Report on systems is available on four levels:

Corporate level All selected system versions are listed per selected project, configuration and phase version.

Project level All selected system versions in the project are listed per configuration version and phase version.

Configuration level

All selected system versions are listed per selected phase version.

Phase level All selected system versions are listed.

The report includes:

- The selected version of the system
- The status of the selected system
- The link status of the selected system

To create a report on Systems:

1. Select Utilities | Reports | On Systems.
The results of the generated Report can be viewed in the Monitoring Window.

Report On Use Cases...

In: Browser and Use Case Diagram Editor

Report on use cases is available on two levels:

- phase version
- system version

The report includes, for each use case, the communication associations to or from the use case and any use case generalizations associated with the use case.

To create a report on Use Cases:

1. Select Utilities | Reports | On Use Cases...
The Report Options dialog box for Use Cases appears.
2. Edit the dialog box
3. Press OK to start the Report
The results of the generated Report can be viewed in the Monitoring Window.

Reports

Use **Utilities | Report** to create reports on various repository components and items. Different reports are available at different Browser levels. Following is a complete list of the default reports:

Note

1. All reports are accessible from the browser, diagram related reports are also accessible from the diagram editors.
2. When options mentioned below are not be visible in your menu: check if the appropriate Module is installed.

- Projects
- Configurations
- Phases
- Systems
- Files
- Cdms
- Unreferenced Cdms
- Items and Properties...
(also accessible from all Diagram Editors)
- Component and Properties...
(also accessible from all Diagram Editors)
- Classes
(also accessible from the Class Diagram Editor)
- Class Generalizations
(also accessible from the Class Diagram Editor)
- Missing Operations
(also accessible from the Class Diagram Editor)
- Communications
(also accessible from the Collaboration Diagram Editor)
- Events and messages
(Events also accessible from State Transition Diagram Editor)
(Messages also accessible from Collaboration and Sequence Diagram Editors)
- Actors ...
(also accessible from Use Case Diagram Editors)
- Use Cases...
(also accessible from Use Case Diagram Editors)
- Roles and Users
(only available when Security Module is installed):
- Security ...
(only available when Security Module is installed):
- Groups
(only available when Corporate Modeling Module is installed):
- Corporate Groups
(only available when Corporate Modeling Module is installed):
- Saved Groups
(only available when Corporate Modeling Module is installed):
- Documents
(only available when Document Module is installed):

The Report output is sent to a Monitoring Window.

Reports On Configurations

In: Browser

Report on Configurations is available on two Browser levels:

- corporate
- project

The Report includes

- Object
 - Corporate
 - Projects
 - Configurations
- Type
- Owner
- Version
- Status
- Creator

To create a report on CDMs:

1. Select Utilities | Reports | On CDMs.
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Unreferenced CDMs

In: Browser

Report on Unreferenced CDMs is available on two Browser levels:

- phase version
- system version

The report includes Class Definition Matrices whose corresponding class does not exist anymore.

The CDMs are listed per selected project, configuration, phase and system version.

To create a report on unreferenced CDMs:

1. Select Utilities | Reports | On Unreferenced CDMs.
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Missing Operations

Reports on missing operations is available on all [browser levels](#).

The report includes the classes for which no operations were found.

Report On Events and Messages...

Report On Events and Messages in: [Browser](#)

Report on Messages: in [Collaboration Diagram Editor](#), and [Sequence Diagram Editor](#),

Report on Events: in [State Transition Diagram Editor](#),

Report on Events and Messages is available on two levels:

- **phase version**

- **system version**

Note : When the Report is started from the browser all events and messages of the selected Phase are shown. When the Report is started from a Diagram Editor, only the Events or Message of that Diagram are shown. The report includes:

- Source component of the event or message
- Event or Message name
- Destination component of the event or message
- Diagram in which the event appears

To create a report on Event and Messages / Events / Messages:

1. Select Utilities | Reports | On Events and Messages / Events / Messages.
The results of the generated Report can be viewed in the [Monitoring Window](#).

Reports On Roles and Users

In: Browser

Reports on Roles and Users is available on all browser levels.

The report includes two sections:

Role , listing:

- Role name
- Security level
- User
- Use

User , listing:

- User name
- Security level
- Activated Roles

To create a report on Roles and Users:

1. Select Utilities | Reports | On Roles and Users.
The results of the generated Report can be viewed in the Monitoring Window.

Reports On Security

In: Browser

Reports on missing operations is available on all browser levels.

Note . ObjectTeam reviews all elements downwards from the level you start the report on.

Example . If you start the report on Corporate level the entire repository is searched and this may take a while.

The report includes:

- Controlled objects:
 - on Role Guest
 - on Role Superuser
 - on <role name> of active role
- Controlled Actions (an explanation of the abbreviations can be found on the first page of the report).

To create a report on Security:

1. Select Utilities | Reports | On Security..
The Report Options dialog box for Security appears.
2. Edit the dialog box.
3. Press OK to start the Report
The results of the generated Report can be viewed in the Monitoring Window.

Report Options dialog box, On Actors

In: Browser, and Diagram Editors

The Report Options dialog box for actors (COD actors or UCD actors) contains one checkbutton: Show Free Text.

When you check this button, the Free Text that you typed in the Item Properties dialog box for actors will be included in the Report.

Report Options dialog box, On Missing Operations

In: Browser, and Diagram Editors

Class Name(s) Specify the class or classes for which you want to run a report on missing operations.

If you want to specify more than one class, leave a space between the entered class names. You can also use glob style pattern matching to specify class names. Details on this type of pattern matching can be found in general Tcl documentation.

During a Report on Missing Operations, ObjectTeam traces back events in the following diagrams and reports if the corresponding operation is missing in the selected class:

- Sequence Diagram
- Collaboration Diagram
- State Transition Diagram

Report Options dialog box, On Items/Components and Properties

In: Browser, and Diagram Editors

Property Name(s)

Specify the properties whose name and value you want to appear in your Report on Items/Components and Properties.

If you leave this field empty, all the property names and values that have been edited are printed in the *Property Name* and *Property Value* columns of the report.

You specify a property by entering its **type**. These types are listed in the customization file proplocs.proplocs. To specify the property **Free Text** for instance, you enter **freeText**.

If you want to specify more than one property, leave a space between the entered property types. You can also use glob style pattern matching to specify properties. Details on this type of pattern matching can be found in general Tcl documentation.

Report Options dialog box, On Security

In: Browser, and Diagram Editors
(Utilities | Reports | On Security...)

Role Name(s) Specify the role for which you want to run a report on Security. If you want to specify more than one role, leave a space between the entered roles names. You can also use glob style pattern matching to specify class names. Details on this type of pattern matching can be found in general Tcl documentation.

Report Options dialog box, On Use Cases

In: Browser, and Diagram Editors

The Report Options dialog box for Use Cases contains two checkbuttons:

Show Use Case Properties

When this button is checked, the properties of the Use Cases are added to your Report on Use Cases.

Show Decomposition Information

When this button is checked Decomposition information is added to the Report.

Execution Window Command(s) dialog box

Command(s) Enter the (system) command(s) you want to execute in the Execution Window that will be started. The output of the command(s) will be displayed in the Execution Window.

UNIX To start a C-shell for instance, enter the following command:

`cs`

Windows To start a DOS box for instance, enter the following command:

`command.com`

Reverse Engineer - dialog box

In: [Browser](#)

See your Code Generation Guide for language specific information.

Diagram The name of the [Class Diagram](#). If no name is entered no diagram is created.

Generate Select the generated types:

- [CDs](#) and [CDMs](#)
- Only CDs
- Only CDMs

Overwrite Existing Diagrams

Switch this on if you want to overwrite an existing diagram with the same name as the one specified.

Create Reference Diagrams

Switch this on if you want to create separate CDs for classes that exceed the maximum size.

Maximum Size Specify the maximum sizes in pixels of a [class](#), class tree, or [diagram area](#). If all source file information does not fit in a single diagram, extra diagrams are created. These diagrams have names derived from the specified name, but with the suffix *_n* where *n* is an incremented number starting at zero.

Input Filter Command (C++ only)

Command used to preprocess the header files. Specify the command using the format: `myfilter %1 %2`.

It is then executed as: `myfilter input-file output-file`.

Skip Identifiers File (C++ only)

File containing a list of identifiers to be ignored by reverse engineering. The file must be an ASCII file that contains the identifiers; separate the identifiers using white space.

(*Note*: The `M4_reveng_skipfile` variable also specifies a skip identifiers file. The file specified in this field is used in place of the file specified by the `M4_reveng_skipfile` variable.)

Case (NewEra only)

Identifiers such as class names and variable names are case-sensitive in ObjectTeam, whereas in NewEra they are not. To circumvent conflicts that may arise, you can specify the way in which the reverse engineering tool must handle upper case and lower case:

- First case
The identifier names are created in the case that is encountered first by the reverse engineering tool.
- Upper Case
The identifier names are created in upper case letters
- Lower Case
The identifier names are created in lower case letters

Creating a Database

In: [Browser on implementation system level](#)

Note : This menu entry is **not** available if your RDBMS is Oracle.

Use **Database | Create Database** to create the database your application will work with. The [New Database dialog box](#) will appear. You only need a database if you designed persistent classes.

When you create a database, some initial actions are executed such as the creation of empty system tables and the registration of the database in the repository.

You can create more than one database. The existence of several databases makes it possible to develop and test more than one application for a database, and to develop and test an application on different databases.

If another (active) database is selected at the moment you want to create a new one, use [Database | Deselect Database](#) to deselect it.

Destroying a Database

In: Browser on implementation system level

Note : This menu entry is **not** available if your RDBMS is Oracle.

Use **Database | Destroy** to delete a database created with Database > Create Database. The database that is currently selected will be deleted, including all database tables.

If you only want to delete database *tables* from the selected database, you should run an SQL *drop** script.

Executing SQL scripts

In: [Browser on implementation system level](#)

Use **Database | Execute SQL** to run SQL scripts, which are created during SQL code generation.

To create database tables (and procedures) for the selected database for instance, you need to run the SQL script *create*.

The SQL scripts *drop** can be used to drop database tables and procedures in the selected database.

Generating a Makefile

In: [Browser on implementation system level](#)

Use **Target | Generate Makefile** to create a makefile on the basis of the include sections of the source code files. A *makefile template* is used to generate a makefile.

(*Unix only*): You can generate the dependency list of the makefile by executing the shell script **mkdep**.

Running an Executable

In: Browser on implementation system level

Use **Target | Run** to run an executable. The executable is started in a separate Execution Window.

Selecting/Deselecting a Database / (Schema)

In: Browser on implementation system level

Use **Database | Select Database** (for Oracle databases: **Database | Select Schema**) to select the database your application will work on. The Select Database/Schema dialog box appears.

You only need a database if you make use of persistent classes.

If another (active) database is selected at the moment you want to select a new one, use **Database | Deselect Database** (for Oracle databases: **Database | Deselect Schema**) to deselect it, or press the *Cancel* button to discard the operation.

Setting Password to access Database (Server)

In: [Browser on implementation system level](#)

Some RDBMSs need to validate if a user is allowed to connect to a particular database or database server. The user needs to enter a valid password in order to be able to carry out database-related tasks, such as [executing SQL scripts](#).

With **Database | Set Password** you can enter the required password once, without having to enter it every next time password validation is required. The password is remembered by ObjectTeam until you exit the browser.

Dialog Box

<i>Server</i>	If it is the database server that requires a password, specify it in this field.
<i>Schema</i>	If it is the Oracle schema that requires a password, specify it in this field.
<i>Database</i>	If it is the database that requires a password, specify it in this field.

Updating User Environment

In: [Browser on implementation system level](#)

Use **Utilities | Update User Environment** to make your user environment consistent with the repository.

Round Trip Selected

In: [Browser](#)

You can use **Utilities | Round Trip Selected** to move your changes in source files back into the CDMs of the Object Design phase. This ensures that your changes are preserved when you regenerate the source files.

Conditions :

- You must have installed a [Code Generator Module](#).
- Round -trip engineering is only available for certain code generators and certain types of source files. See your Code Generation Guide for the options available for you Code Generator.
- See your Code Generator Guide for information on changes that can be captured by Round-trip engineering.

To run round-trip engineering:

1. Move to the System level of the Implementation phase.
2. Select the header files (for non-persistent classes) on which you want to run round-trip engineering.
3. Select Utilities | Round Trip Selected.
*ObjectTeam opens the [Round Trip Selected Files dialog box](#), then compares the classes in the header files to the classes in the Object Design phase.
For each difference it finds, ObjectTeam proposes an action and prompts you for confirmation.
Based on your responses, ObjectTeam updates the CDMs of the Object Design phase.*

Note : The [customization file](#) **roundtrip.roundtrip** in the **/m4_home/etc** directory controls what actions round-trip engineering proposes, as well as the default answers it supplies. To examine or modify the current behavior of round-trip engineering, examine or modify the customization file.

Import Into Implementation Phase, New Dialog Box

In: [Browser](#)

Note : This dialog box is only available in persistent code generator modules. See your code generation documentation.

Specify the kind of code generation you want in this dialog box:

- OOPL Model** Check this if you want to generate the source files for your target language from the non-persistent classes in your class model. If you have persistent classes in your model, and your target language supports persistent code generation, embedded SQL files are generated from these classes.
Persistent classes are classes for which the property **persistent** is switched on.
- SQL Model** Check this if you want to generate SQL *create* and *drop* scripts from the persistent classes in your class association model.
You need to run the *create* script, using [Database | Execute SQL](#), in order to create the structure of your target database.
You can use the *drop* script to drop the structure of your target database.

New Database dialog box

In: Browser on implementation system level

Note : This dialog box is **not** available if your RDBMS is Oracle.

Server Fill in the name of the server on which the database will reside.

Database Fill in the name of the database. Keep the restrictions in mind that your RDBMS imposes on the length of the database name and the use of special characters.

Press the *OK* button to confirm the current information or the *Cancel* button to discard the operation.

Select Database dialog box

In: Browser on implementation system level

The dialog box contains:

Server / (Oracle SID)

Fill in the name of the database server machine. If this server is not the same one you are developing on, make sure the server is accessible.

If your RDBMS is Oracle, you specify an *Oracle SID* instead of a *Server*.

Apply

Press this button after you have entered a server name. All the available databases on the server will be listed. It may take a while before the list of databases appears.

Database / (Schema):

Select a database from the list box or enter one in the entry field to select a database.

If your RDBMS is Oracle, you specify a *Schema* instead of a *Database*.

Press the *OK* button to confirm the current information or the *Cancel* button to discard the operation.

Password dialog box

In: Browser on implementation system level

The dialog box contains:

<i>Server</i>	If it is the database server that requires a password, specify it in this field.
<i>Schema</i>	If it is the Oracle schema that requires a password, specify it in this field.
<i>Database</i>	If it is the database that requires a password, specify it in this field.

Round Trip Selected Files dialog box

In: [Browser](#)

Note . See your Code Generation Guide for more language specific information on the actions below.

Use the buttons to respond to the proposed actions:

Button	Action
Yes	Make the proposed change.
No	Do not make the proposed change.
<i>Default to All</i>	Use the default response for all remaining questions.
Stop	Cancel all changes
OK	Leave this dialog box. <i>ObjectTeam updates the model based on your responses.</i>

Reverse Engineering <language>...

In: Browser

Use **Utilities | Reverse Engineer...** to create classes from source files.

Conditions :

- You must have installed a Code Generator Module.
- Round -trip engineering is only available for certain code generators and certain types of source files. See your Code Generation Guide for the options available for you Code Generator.
- See your Code Generator Guide for information on changes that can be captured by Reverse engineering. Reverse Engineering allows you to do the following:

- Make class libraries available inside the ObjectTeam environment.
- Understand structure and functionality of class libraries.

Warning : Reverse engineering is not designed to capture all the information in the source files. Therefore, do not use reverse engineering to import classes in order to maintain them in ObjectTeam.

To run Reverse Engineering:

- Move to System level in Object Design phase.
Typically , you reverse engineer into an empty system. This prevents name conflicts with existing classes. Classes are reverse engineered with scope `Phase' by default, and, therefore, sometimes an empty phase may be needed.
- Select **Utilities | Reverse Engineer [language]**.
- A File Selection dialog box appears prompting you to select the files to be reverse engineered.
- Select the files that you want to reverse engineer, then select OK. A Reverse Engineer [language] dialog box appears prompting you to select the options that you want to use.
- Select the options, then select OK.
ObjectTeam reverse engineers the selected file, reporting the results in a Monitoring window. The selected source files are now converted to a Class Diagram with CDMs using the parameters specified in the Reverse Engineer [language] dialog box.

Generating Code

In: Browser Before you can generate code, you must have installed the required Code Generator's Module. All Code Generator's Modules add a menu-item *Generate <language name>* to the Utilities menu.

To (re)generate code:

1. Select the level and the objects at that level that you want (re)generated:

- To generate code at Phase Level:
Select the Implementation Phase.
- To generate code at System level:
Select the System in the Implementation Phase.
- To regenerate code at System level:
Select the File(s) that you want regenerated in the Information Area.

2. Select **Utilities | Generate <language name>**.

A cascading menu appears.

3. Select one of the following:

- Select **New** to generate code for all classes that are defined in this system and that have not yet been generated into code.
- Select **Selected** to regenerate the files that you have selected in the Information area of the browser.
- Select **All** (some Code Generators only) to (re)generate code for all classes that are defined in this system.

ObjectTeam opens a Monitor window for displaying log messages, then generates the files.

In the Monitoring Window you can monitor the Code Generation process.

Compile With Options Dialog Box, Java

In: Browser, and Modules

To specify compiler options

1. Select **Target | Compile With Options**
The Compile With Options dialog box appears.
2. Specify the compiler options in the dialog box and press OK.

Compile With Options Dialog Box

In: [Browser](#)

The *Compile With Options Dialog Box* contains a text box in which you can enter options for the compiler.

For details on compiler options, refer to the compiler documentation.

Stopping a system process

(In: Monitoring Window)

The commands in the **Process** menu are used to stop a process running in the Monitoring Window, like a report that is running longer than expected.

The commands are, in order of severity:

Command	Result
Terminate	Running process is terminated through its own handler. (UNIX signal: SIGTERM)
Abort (UNIX only)	Running process is terminated through its own handler and a core is dumped. (UNIX signal: SIGQUIT)
Kill (UNIX only)	Running process is terminated. No locks on tables or files are removed (UNIX signal: SIGKILL)

WARNING. Use the **Process** commands with caution, as they can create incomplete or inconsistent objects, locks or (on UNIX-based platforms) core dumps. Not all taken actions are automatically undone. You must delete (partially) created objects yourself.

Saving Monitoring Window output

(In: Monitoring Window)

Use **File > Save Output** to save output displayed in a Monitoring Window window to a text file. The Save Output dialog box appears.

Menu Bar - Monitoring Window

The following menus are available in the Monitoring Window:

- **File** menu
 - Save Output...
 - Print
 - Exit
- **Edit** menu
 - Copy
- **View** menu
 - ToolBar
 - MessageArea
- **Options** menu
 - Font ...
 - Printer Setup...
 - Clear Screen
 - Reuse
 - Suspend Output
- **Process** menu
 - Terminate
 - Abort
 - Kill
- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

Locking Monitoring Window output

(In: Monitoring Window)

Sometimes the output of a process is very big. The result is a continuous flow of lines that you can hardly keep track of. To stop the flow of lines, you can turn on the radio button **Suspend Output**, which can be found under the **Options** menu.

You can then view all the output that was sent to the display area before the button was pressed.

The remaining part of the output is displayed when you turn **Suspend Output** off again.

Preventing a Monitoring Window from being replaced

(In: Monitoring Window)

Usually when a process starts a new Monitoring Window, it replaces the last Monitoring Window on the screen, if there is one. If you want to prevent a certain Monitoring Window from being replaced, you can turn off the radio button **Reuse**. This button can be found under the **Options** menu in the Monitoring Window.

Printing the displayed Monitoring Window output

(In: Monitoring Window)

1. Select **Options > Printer Setup...** to check the Text printer command.
2. Enter another printer command in the Text Printer dialog box if needed.
3. Confirm the print command with the *OK* button.
4. Select **File > Print** from the menu bar.

The displayed output is now printed to the specified Text printer.

Monitoring Window

A *Monitoring Window* is used for displaying output of certain tools, such as:

- Report commands
- Check commands
- Generate commands
- Import commands
- Compare commands

If you start a Monitoring Window with **Utilities > Monitoring Window...** you can enter a command in the Monitoring Window dialog box. The output of this command is then printed to the display area of the Monitoring Window.

Window Areas

The default Monitoring Window contains the following areas. You can use the *View* menu to hide and display the Tool Bar and Message Area.

- **Menu Bar** - You select menu entries from here.
- **Tool Bar** - You select commonly used menu entries from here.
- **Display Area** - The actual output from the command is shown here.
- **Message Area** - Displays system messages. In UNIX, you can browse through the message history with the arrow-up and arrow-down symbols at the right.

Saving Output dialog box

(In: Monitoring Window)

You can specify a new file name here if you do not want to use the displayed default file name.

Warning . If the specified file already exists, it is overwritten without any warning.

State Transition Diagram Editor - Control Panel

In: Diagram Editors

The following symbols are available in the control panel of the State Transition Diagram Editor. (Arranged according to their position in the control panel.)

<u>State</u>	<u>Super State</u>
<u>Start State</u>	<u>Final State</u>
<u>STD Class</u>	<u>Note</u>
<u>Vertex</u>	<u>Transition</u>
<u>Event Message</u>	<u>Note connector</u>
Select	

By default, all the symbols are included in the control panel. You can remove and add symbols, and change their order by editing the customization file **<diagram_name>.pnl**. You can edit this file using the Edit Control Panel dialog box

Centering a diagram

In: Diagram Editors

Use **Edit | Center** to move a diagram to the middle of the drawing area.

When you center a diagram, the editor ensures that the boundaries of the diagram extend to 800 pixels beyond the furthest symbol. If necessary, the size of the drawing area is extended. The maximum size of a drawing area is 16,384 pixels. So you can use **Edit | Center** to increase the size of the drawing area.

Centering a diagram is considered as a change to the diagram, since the coordinates of the diagram and its components are changed. Therefore, you cannot center a diagram in read-only mode.

Renaming an Item

In : Diagram Editors

Use **Item | Change Name...** to rename an item that is referred to in a diagram object.

To change the name of an item:

1. Select in the drawing area the diagram object that refers to the item you want to rename.
If the selected diagram object refers to more than one item, a dialog box appears in which you can select the item name of your choice.
2. Select **Item | Change Name...**
The Change Name dialog box appears.
3. Enter the new name and click on OK.
If the specified name already exists, the Name Conflict dialog box appears.

Checking the contents of a diagram

In: [Diagram Editors](#))
and [Browser on System level](#)

Use **Check | Contents** to check if a diagram is correctly drawn.

The *symbols* in the diagram are checked on:

- Name
- Definition
- Connection with other symbols

The *connectors* are checked on:

- Direction
- Number
- Name

For a comprehensive list of checking rules, refer to the *ObjectTeam Customization Guide*.

You must save a diagram first before selecting **Check | Contents**. A [Monitoring Window](#) window reports on the checking actions.

Checking a local model

In : [Diagram Editors](#)

Use **Check | Local Model** to check if a diagram is correctly drawn. What is checked, depends on the diagram objects selected in the [drawing area](#):

- **Nothing selected:** All the events received by all [classes](#) in the current Class Diagram are checked. The result is the same as when Check Local Model is invoked from the browser with the *cad* or the *cdm* selected.
- **Class (es) selected:** Only the events received by the selected classes are checked.

The messages received by classes can be:

- [COD Messages](#)
- [SD Messages](#)
- [STD Event Messages](#)

For a comprehensive list of checking rules, refer to the *ObjectTeam Customization Guide*.

You must [save](#) a diagram first before selecting **Check | Local Model**. A [Monitoring Window](#) reports on the checking actions.

Starting the Class Browser

In : Diagram Editors

Use **Utilities | Class Browser** to start the class browser.

With the class browser you can:

- get an overview of classes
- view the contents of classes
- view relations between classes in the current system

Closing a diagram

In : [Diagram Editors](#)

Use **File | Close** to close the current diagram. If you have made any changes to the diagram and you haven't saved it yet, you get the chance to save the diagram or to ignore the changes.

Setting the color in the drawing area

In : Diagram Editors

Use the **Options | Color** to change the colors used in the drawing area of a diagram editor. You can change the color of the following objects:

- **Foreground** : used to display the diagram objects
- **Background** : used to display the drawing area itself
- **Selection** : used to display the diagram object(s) that are currently selected

On black and white monitors, you can use the **Color** menu to set the "blackboard" mode: white objects on a black background.

Copying diagram objects

In: [Diagram Editors](#)

Use **Edit | Duplicate** to copy an object or a group of objects in a diagram. To copy one or more objects, do the following:

1. Select the object(s) in the drawing area. You can do that by dragging a rectangle around the objects of your choice or by selecting objects with the CTRL-key pressed.
2. Select **Edit | Duplicate**.
A copy of the selected object(s) is attached to your cursor.
3. Move the cursor to the intended spot.
4. Click with the left mouse button to fix the object(s).

If a suggested copy action conflicts with the syntax of the diagram editor, the message area will display a message saying that an illegal copy was attempted. The copy action will not be carried out.

To copy a complete diagram into the current diagram, use [File | Read](#).

Deleting diagram objects

In : [Diagram Editors](#)

Use **Edit | Delete** to delete an object or a group of objects from the diagram.

If a selected object is a node (not a vertex), the connectors connected to the object will disappear as well.

If the selected object is a vertex in a connector, the connector straightens.

Editing the scope of an item

In : [Diagram Editors](#)

Use **Item | Edit Scope** to change the extent to which the definition of an item is valid inside the project. The following scopes of items are available:

- Phase
- System
- File

You can change the scope of an item in the [Edit Scope dialog box](#).

What the available options for a scope change are depends on the following criteria:

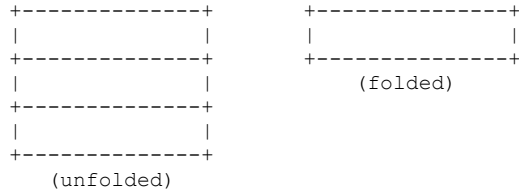
- the current scope of the selected item; this is indicated in the dialog box.
- the scopes the selected item is allowed to adopt (see the *ObjectTeam Modeling Guide*).

You can change the scope **downwards** (for example, from *Phase* to *System*) or **upwards** (for example, from *System* to *Phase*).

Folding and unfolding a class

In: Class Diagram Editor

Use **Edit | Fold** to reduce the size of a CD class its attribute section and operation section:



The information of the attribute and operation section is not lost when you fold a class; you can bring the hidden sections back by just using **Edit | Unfold**.

Initial Fold

Whether a class symbol must appear folded or unfolded when you place it in the drawing area, is determined by the M4 variable *M4_initial_fold*. You can change the value of this M4 variable from the diagram editor by using the switch **Options | Initial Fold**.

Manipulating File Properties

In: [Diagram Editors](#)

Use **File | Properties** to *edit*, *show* and *delete* property values of the current diagram. The same properties are involved as when you use the [Properties menu in the browser](#) for a diagram object (FileVersion).

To edit, show or delete properties for items within the current diagram, use the options under the [Item](#) menu.

Setting the font of the drawing area

(In: Diagram Editors)

Use **Options | Font | Bold** to change the font used for class names in in CDs, initiators in SDs, state activities and classes in STDs, and actors and systems in UCDs.

Use **Options | Font | Normal** to change the font used for all other labels in the drawing area of any diagram editor.

Use **Options | Font | Annotation** to change the font used for Notes. If it is not changed, the Annotation font will be a smaller version of the Normal Font.

The fonts and the font properties that you can select are the font and font properties available in your windowing system (X-Windows or MS-Windows).

Setting the grid in the drawing area

In : Diagram Editors

Use **Options | Grid...** to select a grid size for the drawing area. The Grid dialog box appears.

Moving diagram objects

(In: [Diagram Editors](#))

Use **Edit | Move** to shift an object or a group of objects from one place to another in the diagram. To move one or more objects, do the following:

1. Select the object(s) in the drawing area. You can do that by dragging a rectangle around the objects of your choice or by selecting objects with the CTRL-key pressed.
2. Select **Edit | Move**.
The selected object(s) are attached to your cursor.
3. Move the cursor to the intended spot.
4. Click with the left mouse button to fix the object(s).

If a suggested move action conflicts with the syntax of the diagram editor, the message area will display a message saying that an illegal move was attempted. The move action will not be carried out.

Printing diagram

In: Diagram Editors

Use **File | Print** in a diagram editor to print off the current diagram.

The diagram is printed according to certain print and printer settings that you can change using the following menu entries:

- **Options | Printer Setup** - to change Graphical printer settings
- **Options | Print Options** - to specify the way diagrams are printed

Copy a diagram into another diagram

In: Diagram Editors

Use **File | Read** to copy an entire diagram into the current diagram. It is best to Save the current diagram before you use **File | Read**.

You select a diagram in the Read Diagram dialog box.

Redrawing a diagram

In: Diagram Editors

Use **Options | Redraw** to refresh the drawing area in a diagram editor. This may be needed after you have copied, moved or deleted many diagram objects.

Reloading a diagram

(In: Diagram Editors)

Use **File | Reload** to load the last saved version of a diagram.

If changes were made, the Reload dialog box appears, requesting confirmation of the *Reload* action.

The Reload dialog box appears.

Specifying the syntax of diagram object labels

In: [Diagram Editors](#)

Use **Options | Syntax...** to specify syntax rules for the names of objects in diagram editors. You can specify syntax rules in the [Syntax dialog box](#).

Syntax M4 variables

Syntax rules are defined in M4 variables. Which M4 variable is applicable for which diagram object depends on the [item](#) type:

M4 variable	item type
M4_class_syntax	cl
M4_data_syntax	de
M4_process_syntax	pe
M4_state_syntax	st

For example, the variable `M4_class_syntax` is applicable for [associations](#), since the item type of an association is **cl**.

If a variable is not specified, the syntax for the object names is completely unrestricted (apart from the restriction of the length of the internal buffer). Any specification, however minor, must comply completely with the syntax rules, and every field of the syntax must be present.

Saving a diagram

In: Diagram Editors

Use **File | Save** to save the current diagram.

Undoing changes applied to a diagram

In : Diagram Editors

Use **Edit | Undo** to cancel your last action. If you select Undo two times in a row, the second Undo cancels the first one and thus works as a Redo.

A right mouse button click works also as Undo. A right mouse button click also interrupts actions such as Move and Duplicate.

Zooming in/out on a diagram

In: Diagram Editors

Use **Options | Zoom In** to increase or **Options | Zoom Out** to decrease the scale factor in the drawing area. The scale factor is displayed in the message area when you zoom in or out.

Menu Bar - Diagram Editor

Below, the default menus of the Diagram Editors are listed. You can add new menus and redefine existing ones by using the Customization Editor.

[File] [Edit] [View] [Item] [Options] [Check] [Utilities] [Help]

- **File** menu
 - Reload
 - Open By Name...
 - Open
 - Close
 - Read ...
 - Save
 - Properties
 - Edit ...
 - Delete ...
 - Show ...

 - Print
 - Exit
- **Edit** menu
 - Undo
 - Cut
 - Copy
 - Paste
 - Delete
 - Move
 - Duplicate
 - Replace
 - Select All
 - Deselect All
 - Center
 - Fold
(CDE only)
 - Unfold
(CDE only)
- **View** menu
 - ToolBar
 - Context Area
 - Message Area
- **Item** menu
 - Change Name
 - Edit Scope...
 - Edit Properties...
 - Delete Properties...
 - Show Properties...
- **Options** menu
 - Redraw

- ZoomOut
- ZoomIn
- Grid ...
- Font
 - Normal ...
 - Bold ...
- Color
 - Foreground ...
 - Background ...
 - Selection ...
- Print Options...
- Printer Setup...
- Pointer Focus
- Syntax ...
- Initial Fold
(CDE only)
- **Check** menu
 - Contents
 - Local Model
 - Use Case Model
(only in UCDE)
- **Utilities** menu
 - Class Browser
 - Monitoring Window...
 - Execution Window...
 - Reports
 - Delete Unreferenced Items
- **Help** menu
 - What 's This?...
 - On Help
 - Help Topics
 - About ...

Class Definition Matrix (CDM)

In: Browser

When you enter attributes or operations for a class in a Class Diagram, the **Class Definition Matrix** (CDM) for the class is created or updated automatically.

In the browser, a Class Definition Matrix is represented by the file version object of type *cdm*. This object is a child object of the browser object SystemVersion. So the object *cdm* appears in the information area on:

- System level

You cannot edit a CDM directly. Instead, when you edit a CDM, ObjectTeam opens the CD in which the CDM's class appears. If the class appears in more than one CD, a dialog box appears prompting you to select the CD that you want to open.

Item

In: [Browser](#)

Items are not specified directly, but generated from symbol labels in diagrams. An unnamed graphical component in a diagram is just a graphical element that cannot be referred to, but a named component creates an item in the repository.

When you [define an item](#) it appears in the system of its definition under the pseudo object [<defined items>](#).

Defined Items

Defined items allow different graphical elements to refer to the same semantic element by their name.

For example: when you open a class in a CD, you get the option to:

- create or open a diagram
- create or open a system

both with the same name as the class you opened (see: [diagram navigation](#)). The diagram and system refer to the same item as the class. An item is identified by its name and its type.

<i>Item name</i>	An item name can be up to 80 characters in length. All printable ISO-Latin-1 characters, except space, tab, newline, slash, colon, and comma are allowed in item names.
<i>Item type</i>	The type of an item is determined by the symbol used to create it. The relationship between symbol and type is fixed internally in ObjectTeam and cannot be changed: see Item-Component relations for a full overview of component types.

Qualified Items

Qualified items are items that belong to another item such as operations and classes. Because an operation always belongs to a class, an operation is a qualified item.

The name of a qualified item consists of the owner item name and the qualified item name. For example, the getName operation on the Member class is named Member.getName. The operation getName may appear in several classes; however, because getName is a qualified item, each operation is unique to the class in which getName appears.

State Transition Diagram (STD)

In: [State Transition Diagram Editor](#)

Objects have a life cycle. If this is complex, it can be described in a *State Transition Diagram*, showing which messages (all objects of) a [class](#) can receive and the changes of state as a result of these messages.

In the browser, a State Transition Diagram is represented by the [file version](#) object of type *std*. This object is a child object of the browser object [SystemVersion](#). So the object *std* appears in the information area on:

- [System level](#)

By opening the object *std* from the browser on System level, you start the [State Transition Diagram Editor](#).

The browser object *std* is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- SystemVersion
        |
        +- {file version}
```


Aggregating Classes

In: Class Diagram Editor

To aggregate classes:

1. Make sure that the correct type of multiplicity indicators are selected in the control panel.
2. Select an aggregation symbol from the control panel.
3. Connect the class symbols you want to aggregate using the aggregation symbol.
The class symbol from which you start the aggregation represents the *assembly* within the aggregation. The class symbol the aggregation leads to represents a *component* within the aggregation.
4. Enter the relevant aggregation information.

To complete an aggregation:

- define its properties using Item | Edit Properties...

Associating Two Classes

In: Class Diagram Editor

To associate two classes:

1. Select the appropriate begin and end association symbols from the control panel
2. Select an association or a qualified association symbol from the control panel.
3. Move the cursor to the first class symbol
4. Press the left mouse button in the class symbol
5. Move the cursor away from the class symbol
If you want a corner in the connector, you can press the left mouse button anywhere except in another class symbol. From there, you can continue moving the cursor.
You can undo a click with the left mouse button by pressing the right one.
6. Press the left mouse button in the class symbol you want to associate the first class symbol with
The two classes are now associated.
7. Enter the relevant association information such as *name*, *role names* and *constraints*.

To complete an association:

- define its properties using Item > Edit Properties...

Creating a Class

In: Class Diagram Editor

To create a class:

1. Select a CD class symbol from the control panel.
2. Insert the symbol at the intended spot in the drawing area.
3. Enter the class name in the top section of the placed class symbol.

To define a derived class, enter a slash before the class name, e.g.:
`/Patient`

To complete an aggregation:

- define its properties using Item | Edit Properties...

To complete a class definition:

- define its attributes.
- define its operations.
- define its properties using Item | Edit Properties...

Editing Label Text in Diagram Objects

In: [Diagram Editors](#)

Inside the label of a symbol you can move the cursor using the cursor keys. Use the **Delete** key to delete characters to the right of the cursor and the **Backspace** key to delete characters to the left of the cursor.

You can also use **Edit | Cut**, **Edit | Copy**, and **Edit | Paste** to move text between a label and any text window.

In nodes, text is centered automatically. To enter multi-line text, enter a return character.

You can control how the label text is entered, e.g. a maximum number of characters, etc. Therefore, if you cannot enter text in a label, it is possible that [label settings](#) were installed and that the text you try to enter does not comply with these settings.

Associating More than Two Classes

In: Class Diagram Editor

To associate more than two classes:

1. Select an n-ary association symbol from the control panel.
 2. Place the symbol somewhere in the drawing area, near the class symbols you want to associate.
 3. Enter a name for the n-ary association symbol.
 4. Select the appropriate begin association symbol from the control panel.
 5. Select an n-ary association connector from the control panel.
 6. Connect every class symbol participating in the association with the n-ary association symbol.
 7. Enter the relevant n-ary association information, such as *role names* and *constraints*.
- Note* : N-ary associations are not supported in the standard code generation.

Resizing a Diagram Object

In: Diagram Editors

A diagram object can be enlarged or reduced, and, except for circles, its proportions can be changed.

To resize a diagram object:

1. Click on the Select symbol in the control panel of the diagram editor.
2. Click once on the object in the drawing area with the left mouse button.
Handles appear on the outer edge of the selected object to indicate that it is selected.
3. Place the cursor on one of the selection handles.
The cursor turns into a resize cursor.
4. Drag the handle out to enlarge the object, or in to reduce it.

Selecting a Diagram Object - task

In: [Diagram Editors](#)

To select a single object:

1. Click on the Select symbol in the control panel of the diagram editor.
2. Click once on the object in the [drawing area](#) with the left mouse button.
Handles appear on the outer edge of the selected object to indicate that it is selected.

To select additional objects:

1. Hold down the **Control** key and click on the additional objects.
2. Alternatively , hold down the **Control** key and drag the mouse button over the desired objects.

Diagram Editors

A diagram editor is a graphic tool with which you create diagrams. It offers a control panel with diagram objects and menus to manipulate the information stored in the diagram.

The following diagram editors are available:

- Class Diagram Editor
- Collaboration Diagram Editor
- Sequence Diagram Editor
- State Transition Diagram Editor
- Use Case Diagram Editor

State Transition Diagram Editor

In: Diagram Editors

Use the State Transition Diagram Editor (STDE) to draw State Transition Diagrams.
The objectives of this diagram technique are to:

- show the states of a class and the events it receives
- document the behavior of the class
- aid communication

The names of State Transition Diagrams must have the following form:

```
<class name>:<activity name>
```

Window Sections

The default diagram editor window contains the following sections:

- Menu Bar
- Tool Bar
- Context Area
- Drawing Area
- Control Panel
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Context Area

In: Diagram editors

The *Context Area* in a diagram editor is located between the tool bar and the drawing area. It contains the following fields:

<u>Project</u>	<u>System</u>
<u>Phase</u>	Diagram Version

Drawing Area

In: Diagram editors

The *drawing area* is located in the center of the diagram editor window between the context area and the message area.

The drawing area is the worksheet of the diagram editor: in this part of the diagram editor you actually create and edit (parts of) your diagrams. You insert symbols from the *control panel* here, connect them with connectors and add the appropriate text.

The drawing area only offers a partial view on the entire drawing space. The scroll bars at the right and at the bottom allow you to change the part that can be seen.

There are a few features in the diagram editor available that make working in the display area more comfortable, such as the option to zoom in and out and the options to move, copy, delete and replace symbols.

Use Case Diagram Editor - Control Panel

In: Diagram Editors

The following symbols are available in the control panel of the Use Case Diagram Editor. (Arranged according to their position in the control panel.)

<u>Use Case</u>	<u>Use Case Actor</u>
<u>Note</u>	<u>Vertex</u>
<u>Undirected Communication Association</u>	<u>Directed Communication Association</u>
<u>Use Case Generalization</u>	<u>Note connector</u>

Select

By default, all the symbols are included in the control panel. You can remove and add symbols, and change their order by editing the customization file **<diagram_name>.pnl**. You can edit this file using the Edit Control Panel dialog box

Changing the editing focus

In: [Diagram Editors](#)

A text field must have the *editing focus* before you can edit the text in the field. In a diagram, the label of every diagram component is a text field. Therefore, for easy editing of the diagrams, it is important to select the behavior that you are most comfortable with.

Use **Options | Pointer Focus** to determine how to give a field the editing focus.

- In Windows, you generally move the editing focus to a text field by clicking on the field. ObjectTeam works this way when **Options | Pointer Focus** is not selected. This is the default behavior.
- In UNIX, you generally move the editing focus to a text field by moving the pointer to the field. (Clicking on the field is not necessary, but the pointer must remain on the field.) ObjectTeam works this way when **Options | Pointer Focus** is selected.

Pointer Focus M4 variable

The current value of the pointer focus option is stored in the M4 variable *M4_keyboardfocuspolicy*.

- A value of *explicit* indicates the Windows-like behavior
- A value of *pointer* indicates the UNIX-like behavior.

Use Case Diagram (UCD)

In: Use Case Diagram Editor

A *Use Case Diagram* models the way in which an actor can use the system. You can then use the Sequence Diagram to further define each path through the system.

A use case represents a particular sequence of transactions between the system and an actor. The collection of all use cases, therefore, specify all the ways of using a system. You can use UCDs to analyse system requirements and to help you define system boundaries.

In the browser, a Use Case Diagram is represented by the file version object of type *ucd*. This object is a child object of the browser object SystemVersion. So the object *ucd* appears in the information area on:

- System level

By opening the object *ucd* from the browser on System level, you start the Use Case Diagram Editor.

The browser object *ucd* is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- SystemVersion
        |
        +- {file version}
```


Use Case Diagram Editor

In: Diagram Editors

Use this diagram editor to draw Use Case Diagrams.

Window Sections

The default diagram editor window contains the following sections:

- Menu Bar
- Tool Bar
- Context Area
- Drawing Area
- Control Panel
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Class Diagram Editor - Control Panel

In: Diagram Editors

The CD Control Panel contains the following symbols:

Class N-ary Association
More Classes Note
Vertex Association
Aggregation Qualified Association
Qualified Aggregation N-ary Association Connector
Constraint Association Class Connector
Generalization Connector Overlapping Generalization Connector
Note Connector Propagation
Select
Multiplicity Indicators

Control Panel - Collaboration Diagram Editor

In: Diagram Editors

The following symbols are available in the control panel of the Collaboration Diagram Editor (arranged according to their position in the control panel).

<u>Actor</u>	<u>Instance</u>
<u>N-ary link</u>	<u>Note</u>
<u>Vertex</u>	<u>Note connector</u>
<u>Link</u>	<u>Aggregation Link</u>
<u>Qualified Link</u>	<u>Qualified Aggregation Link</u>
<u>Backward messages:</u>	
<u>Nested</u>	<u>Forward messages:</u>
<u>Flat</u>	<u>Nested</u>
<u>Asynchronous</u>	<u>Flat</u>
	<u>Asynchronous</u>
<u>N-ary Link connector</u>	Select
<u>Link Start stereotypes:</u>	
association	<u>Link End stereotypes:</u>
parameter	association
local	parameter
global	local
self	global
	self

By default, all the symbols are included in the control panel. You can remove and add symbols, and change their order by editing the customization file **<diagram_name>.pnl**. You can edit this file using the Edit Control Panel dialog box

Diagram Editors - Control Panels

In: [Diagram Editors Diagram Editor](#) |
[Class Diagram](#)
[Collaboration Diagram](#)
[Sequence Diagram](#)
[State Transition Diagram](#)
[Use Case Diagram](#)

Each diagram editor has a Control Panel from which you can select the symbols for your diagram.

The Control Panel data are stored in the <diagram_name>.pnl files in the Pseudo objects <customization files> and <user customization files> and can be customized.

Sequence Diagram Editor - Control Panel

In: Diagram Editors

Diagram Editors, Control Panels

The SD Control Panel contains the following symbols:

<u>SD Initiator</u>	<u>SD Object</u>
<u>SD Timing Constraint</u>	<u>Note</u>
<u>Vertex</u>	<u>Nested Message</u>
<u>Flat Message</u>	<u>Asynchronous Message</u>
<u>Return Message</u>	<u>Note Connector</u>
<u>In Scope Region</u>	<u>Object Termination</u>
<u>Select</u>	

Edit Scope dialog box

In: Diagram Editors

Current Scope is ...

This option tells you the scope of the selected item. If this is the only option available in the dialog box, you cannot change the scope of the selected item.

Create empty definition on File/System level

With this option you **create** a new item definition, with default property values, in the current file or system. This option is available when changing the scope downwards from:

- System to File
- Phase to File
- Phase to System

Copy definition to File/System level

With this option you **copy** the existing item definition, including its property values, to the current file or system. This option is available when changing the scope downwards from:

- System to File
- Phase to File
- Phase (Ref) to System

Restrict definition to System level

With this option you do not modify item definition, just change scope. (The item is defined in the current system.) This option is available when changing the scope downwards from:

- Phase (Def) to System

Export definition to System/Phase level

With this option you **move** the existing item definition, including its property values, to the current system. The item definition in the file is deleted. This option is available when changing the upwards scope from:

- File to System
- File to Phase(Def)

Make definition available on Phase level

With this option you do not modify item definition, just change scope. (The item is defined in the current system.) This option is available when changing the scope upwards from:

- System to Phase(Def)

Refer to (not yet existing definition) on System level

With this option you **create** a new item definition, with default property values, in the current system. The item definition in the file is deleted. This option is available when changing the scope upwards from:

- File to System

Refer to (not yet existing definition) on Phase level

With this option the item definition in the file is deleted. The item is undefined. This option is available when changing the scope upwards from:

- File to Phase(Ref)

Refer to definition on Phase level

With this option the item definition in the file is deleted. The item definition exists in another system. This option is available when changing the scope upwards from:

- System to Phase(Ref)

Note : If the item is a class and the system contains the CDM, the item definition cannot be deleted; the scope cannot be changed.

Grid Dialog Box

In: Diagram Editors

Grid size

Clicking this button makes an option menu pop up. The options range from 1-64. **1** gives you the least and **64** the biggest number of squares in the grid. The more squares in the grid, the more precisely you can position the symbols, as the symbols snap to the grid.

Item Properties dialog box

In : [Diagram Editors](#)

See for Diagram specific properties:

[Properties in Class Diagrams](#)

[Properties in Collaboration Diagrams](#)

[Properties in Sequence Diagrams](#)

[Properties in State Transition Diagrams](#)

[Properties in Use Case Diagrams](#)

Dialog Box Layout

You can specify the property values of selected diagram objects in this dialog box. The list of available properties depends on:

- The current diagram
- The selected diagram object(s)
- The current phase (Properties in the Object Design Phase are used for Code Generation and are more specific than in the Analysis or System Design Phase)

The Edit/Show Properties dialog box has two sections:

- A *list box* at the left that you use to select the item whose properties you want to edit or show. Most properties are inherited by newer versions of the object.
- A *book* at the right that allows you to edit or view the properties of the selected item. These properties are different for each diagram.
The properties are arranged under different tab pages. For example, the tab page titled **Text** contains the property *Free Text*. You can view different pages by selecting a tab or a page number.

If you have created a new item that is not defined yet, you must press the **Define Item** button before you can edit the item's properties.

Storage of Properties

All the properties for every browser object and diagram object are defined in the following customization files:

```
<M4_home>/etc/<database>/<language>/propdefs.propdefs  
<M4_home>/etc/<database>/<language>/proplocs.proplocs
```

where <database> is the t_* subdirectory associated with your target database and <language> is the l_* subdirectory associated with your target language.

Name Conflict - dialog box

In: Browser, Diagram Editors, Change Name dialog box

This dialog box appears if you have used Item | Change Name in a diagram or File | Change | Name in the browser when the new name is already being used. It offers you the following options:

- | | |
|------------------|--|
| <i>Overwrite</i> | Overwrites the existing item name: the existing item is replaced by the current item. |
| <i>Refer</i> | Makes a reference to the existing item: the current item is replaced by the existing item. |
| <i>Cancel</i> | Discards the renaming operation and quits the dialog box. |

New Diagram dialog box

In: Diagram Editors

Classifier (only for Collaboration Diagram, Sequence Diagram, and State Transition Diagram)

Enter the name of the object that qualifies the new diagram (usually a Class or Use Case).

Name Enter the name of the new Diagram.

OK button creates the new diagram,

Edit button creates the new diagram and opens an editor

Cancel button discards the operation.

The name of a Class Diagram or Use Case Diagram has the format: <name>

The name of the new Collaboration Diagram, Sequence Diagram, and State Transition Diagram has the format:
<Classname>:<Name>

Open By Name dialog box

In : [Diagram Editors](#)

You have the following options in this dialog box:

Diagram

(all diagrams)

Select a diagram name here when you want to

- load a specified diagram or
- reload the current diagram

Selection

(Class and Use Case Diagram)

1. Type an existing diagram name here when you want to
 - load a specified diagram or
 - reload the current diagram
2. Type a new diagram name here when you want to
 - create a new diagram

Qualifier

Name

(Collaboration, Sequence and State Transition Diagram)

1. Type both an *existing classifier* and a *name* here when you want to
 - load a specified diagram or
 - reload the current diagram
2. Type either a *new or existing classifier* and a *new diagram name* here when you want to
 - create a new diagram

Print Options dialog box

In : Diagram Editors

Use this dialog box to specify the way a diagram is printed.

The options are:

- Title* Enter the name that is printed with the diagram.
- Landscape* Switch this option on for landscape printing, switch it off for portrait printing. This option is stored in the M4 variable `M4_ps_mode`.
- Print Box* Switch this button on to allow the Print Information Box to be printed on every sheet. This box contains information like the name of the diagram, the date it was changed last and the status. This option is stored in the M4 variable `M4_print_box`.
- Auto Scale* Autoscaling means that the diagram is made to fit within the specified number of horizontal and vertical pages without distorting the diagram. Switch this option on if your diagram has to be printed on more than one page. This option is stored in the M4 variable `M4_ps_auto_scale`.
- Scale Factor* This option is only available if Auto Scale is off. It sets the PostScript scaling factor. This option is stored in the M4 variable `M4_ps_scale`.
- Horizontal Number Of Pages* Specifies the number of sheets in horizontal direction used to print the diagram. This option is stored in the M4 variable `M4_ps_page_h`.
- Vertical Number Of Pages* Specifies the number of sheets in vertical direction used to print the diagram. This option is stored in the M4 variable `M4_ps_page_v`.
- Title*
- Orientation*
- Scale Strategy* If these are switched on, the information is saved with the diagram.

Reload diagram dialog box

In : Diagram Editors

Use this dialog box to confirm the loading of the last saved version of the current diagram. Edits made since the last time you saved are lost if you confirm.

Select Operation dialog box

Diagram Editors

The options shown depend on the diagram and item you navigate from:

- Class Diagram
- Collaboration Diagram
- Sequence Diagram
- State Transition Diagram
- Use Case Diagram

Depending on the circumstances, you may have to select which item and what type of operation (load diagram, make decomposition, create diagram, start editor) you are interested in.

Operation :

load diagram	a decomposition diagram exists and is opened
make decomposition	create a decomposition diagram with the same name of the selected item from which the Select Operation dialog box was started.
create diagram	create a diagram with the same name as the selected item, the appropriate editor is started.
start editor	a diagram of another diagram editor type exists, the appropriate editor is started and the diagram opened.

Syntax dialog box

In : [Diagram Editors](#)

The syntax rules you specify in the Syntax dialog box have the following form:

```
< >: [...] [...]  
  |   |   |  
length |   following character  
        |  
        initial  
        character
```

Example:

```
8: [A-Z] [A-Za-z0-9]
```

Here the maximum length of the object name is eight characters, every name must start with a capital letter, and after the first letter you can use any alphanumeric character.

length When **0** is filled in, the length of the internal buffer is the only restriction. It is possible to fill in a length, **12** for instance, and leave the character syntax free:

```
12: [ ] [ ]
```

character When filling in the character syntax, it is possible to specify certain letters, upper case and/or lower case, special symbols and a range of symbols:

```
[ abcde] or [a-e] : the lower case letters a to e  
[ A-Z$] : the upper case letters A to Z and the dollar symbol
```

To fill in symbols that have a function in the syntax of this variable, place a backslash (\) in front of the symbol:

```
[\-]
```

The dash actually stands for the range of ASCII values between the two letters or symbols indicated. So it is also possible to specify [*-~]. This means that all symbols are allowed between ASCII value 052 and 176.

In each editor, you can only set those variables that are relevant for that editor. In the Class Diagram for instance, you can edit the following fields:

Class Syntax defines the syntax of labels of objects with the item type *cl*
Data Syntax defines the syntax of labels of objects with the item type *de*
Process Syntax defines the syntax of labels of objects with the item type *pe*

Replacing a diagram component

In: Diagram Editors

With the *Edit | Replace* menu you can replace one component with another in your diagram.

The new symbol must be of a compatible component type as the symbol to be replaced, i.e. you cannot replace a class with an association, but you can replace an aggregation with a qualified association. See also: [changing multiplicity] \ and [changing link stereotypes].

To replace a diagram component:

1. Select the component you want to replace in the drawing area.
2. Select the new component symbol from the control panel.
3. Select **Edit | Replace**.

Note : Only one symbol can be selected for replacement. If more than one symbol is selected in the drawing area, Replace is disabled. Replace is also disabled if no symbols are selected.

Editing Item Properties

In: Diagram Editors

To edit/show/delete the currently available properties for a symbol:

1. Select or more symbols in the drawing area of the diagram.
You can select more than one object in the by dragging the mouse cursor around the objects of your choice. The objects included in the rectangle are then selected.

2. Select:

Item | Edit Properties..., and the Edit properties dialog box will appear.

- Select the item you want to change in the left pane of the dialog box.
- Change the data for the selected item.
- Repeat this until you are done.
- Leave the dialog box by pressing OK to save the changes to your items (Note: changes to your components are not yet saved)

Item | Show Properties..., and the Show Properties dialog box will appear.

- Select the item you want to see in the left pane of the dialog box.
- View the data for the selected item.
- Repeat this until you are done.
- Leave the dialog box with OK: nothing will be saved.

Item | Delete Properties..., and the Delete Properties dialog box will appear.

- Select the item of which you want to delete its properties.
- Press OK, and the properties of the item are set to default.

OMT / UML look

Diagram Editors

The menu_items **Options | UML Look** and **Options | OMT Look** let you switch between an OMT version and UML version of the Class Diagram, Sequence Diagram, and State Transition Diagram editors.

The data in these diagrams remains the same, but their appearance is adapted to the selected look.

These options work independently of the OMT module (whether it is installed or not).

Navigating - Diagram Editors

In: Diagram Editors

Navigating the Diagram Editors means that you from an item in your current diagram, you can open another diagram of either the same type (decomposition) or another type. The name of the new diagram is identical to the name of the item from which you created the new diagram

Use **File | Open** to open or create diagrams that refer to the same item. The Select Operation dialog box appears. The anchor point is the label of the diagram object that refers to one or more items.

If you select such a symbol in the drawing area and select **File | Open** next, you can open an item-related diagram.

Depending on the circumstances, you may have to select which item and what type of operation (load diagram, make decomposition, create diagram, start editor) you are interested in.

The navigation strategy between diagrams is defined in the following customization files:

- **opendefs .opendefs**
Use the Open Strategy Definition Editor to customize this file.
- **openlocs .openlocs**
Use the Open Strategy Availability Editor to customize this file.

Open By Name ...

In: [Diagram Editors](#)

Use **File | Open By Name** to load or create a diagram of the same type of the current diagram editor, or to reload the current diagram.

The [Open By Name dialog box](#) appears.

If the diagram you try to open is already being edited, it opens in read-only mode.

Setting Print Options

In: Diagram Editors)

Use **Options | Print Options...** to specify the way a diagram is printed.

You set the specifications in the Print Options dialog box.

Class Diagram (CD)

In: Class Diagram Editor

Class Diagrams describe the static and structural aspects of a system. They define:

- classes that are part of a system
- associations between these classes
- features of the classes (data attributes and operations)

In the browser, a Class Diagram is represented by the file version object of type *cd*. This object is a child object of the browser object SystemVersion. So the object *cd* appears in the information area on:

- System level

By opening the object *cd* from the browser on System level, you start the Class Diagram Editor

Collaboration Diagram (CD)

In: Collaboration Diagram Editor

The Collaboration Diagram shows a set of objects related in a particular context, and the set of messages exchanged between the objects to achieve a desired operation or result.

The COD is a graph of references to objects and the links between those objects, with message flows attached to its links. The diagram shows the objects relevant to the performance of an operation.

Diagram

In: [Diagram Editors](#)

[Class Diagram](#)

[Collaboration Diagram](#)

[Sequence Diagram](#)

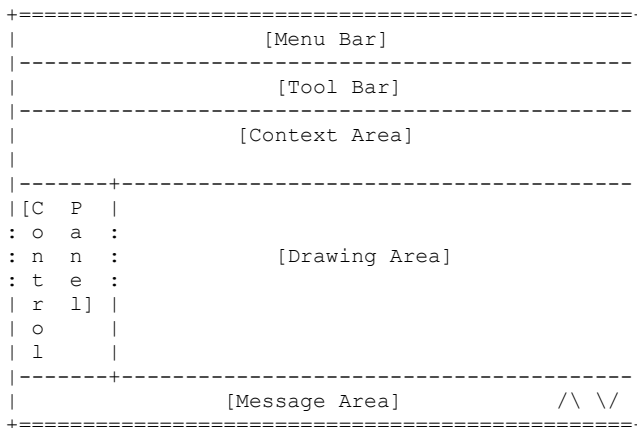
[State Transition Diagram Editor](#)

[Use Case Diagram Editor](#)

Window Sections

The default diagram editor window contains the following sections:

- **Menu Bar** - you select menu entries from here.
- **Tool Bar** - you can select frequently used menu entries from here.
- **Context Area** - displays information about the current diagram.
- **Drawing Area** - you draw the actual diagrams here.
- **Control Panel** - you select diagram symbols from here.
- **Message Area** - displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.



Item and Component properties

In : Diagram Editors

For items and components you can specify two kinds of properties:

Implicit properties:

These properties are automatically created when you add components to a diagram. For example a diamond on one side of a relation makes it an aggregation, and using a slash in an association name indicates that it is derived.

Explicit properties:

These properties are specified through the **Edit Properties dialog box**. These properties provide additional information about the items. These properties are particularly important for code generation.

Warning :

- Item properties are saved when you leave the **Item Properties** dialog box with OK.
- Component properties are saved when you leave the diagram editor with a save option.

The reason for this is that items reside in the repository and can be referred to from other diagram editors, components cannot be referred to from an editor other than that of the active diagram.

Sequence Diagram (SD)

In: [Sequence Diagram Editor](#)

A Sequence is an ordered list of [messages](#) between objects. A Sequence Diagram describes a single scenario of events that can take place in the real world. The *Sequence Diagram* usually models objects rather than classes.

In the browser, a Sequence Diagram is represented by the [file version](#) object of type *etd*.

This object is a child object of the browser object [SystemVersion](#). So the object *etd* appears in the information area on:

- [System level](#)

By opening the object *sd* from the browser on System level, you start the [Sequence Diagram Editor](#).

The browser object *sd* is part of a tree of (versions of) other objects:

```
Corporate
|
+- Project
  |
  +- ConfigVersion
    |
    +- PhaseVersion
      |
      +- SystemVersion
        |
        +- {file version}
```


Aggregation

In: [Class Diagram Editor](#)

Subsections :

[Labels](#)

[Properties](#)

An *aggregation* is a special form of an [association](#). You can [specify the multiplicity](#) of an aggregation through the multiplicity buttons in the control panel.

The name and role labels of the symbol refer to [items](#)

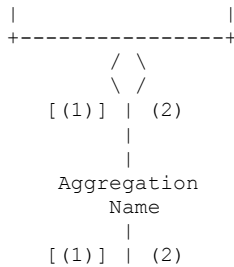
Labels

See also

[Edit Label Text](#)

[Specify Label Syntax](#)

The connector representing the aggregation has the following labels:



Aggregation Name

The name of the aggregation.

(1) = *constraint* The multiplicity of an aggregation can be restricted numerically with this label.

Properties

See also [Edit, Show, Delete properties](#)

Depending on the phase you are in, you can specify different properties for this symbol.

You can edit or see the currently available properties for a symbol with **Item | Edit Properties** or **Item | Show Properties**.

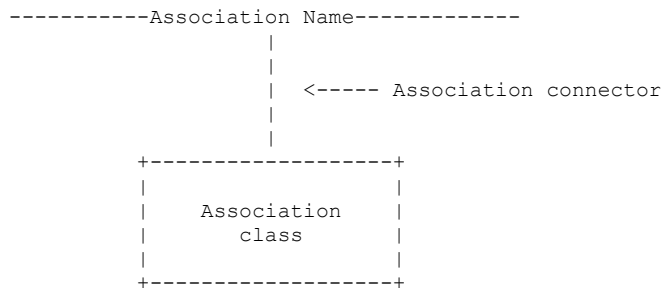
Association Class Connector

In: Class Diagram Editor

An *Association Class Connector* is used to connect a CD class symbol to an association.

The association can be of the following type:

- Association
- Qualified association
- Aggregation
- Qualified aggregation



Association

In: [Class Diagram Editor](#)

Subsections :

[Labels](#)

[Properties](#)

[Class symbols](#) can be related by the *association* symbol. You can specify the multiplicity of an association through the [multiplicity buttons](#) in the control panel.

Labels

The connector representing the association has the following labels:

```
      [ (1) ]                [ (1) ]
      -----Association Name-----
      (2)                    (2)
```

Association Name

The name of the association.

(1) = *constraint* The multiplicity of an association can be restricted numerically with this label. Depending on the multiplicity of the association, the following constraints are valid:

Constraint	Association
0,1	(optional)
0-1	(optional)
0,n	(many)
0-n	(many)
0+	(many)
1	(one)
n	(one or more)
m,p	(one or more)
m-p	(one or more)
m+	(one or more)
{ordered}	(all)

where $m > 0$, $n > 1$ and $p \geq m$

The square brackets are only visible when you are inserting something in the field.

(2) = *role* A role name is compulsory if the multiplicity of the association is mandatory. In optional multiplicity associations, at least one role name is compulsory.

The name and role labels of the symbol refer to [items](#).

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item > Edit Properties](#) or [Item > Show Properties](#).

Class - Attributes Section

In: [Class Diagram Editor](#)

Subsections : [[Labels](#)] [[Properties](#)] [[Scope](#)] [[Navigation](#)]

(In: [Class Diagram Editor](#))

The middle section of a [CD class symbol](#) is reserved for attributes. It can contain zero or more attributes. A typical attribute looks like this:

```
name:char[40]
```

When you enter attributes for a class, the [Class Definition Matrix](#) (CDM) for that class is automatically created or updated.

Labels

The Attributes section can contain one or more attributes. Every attribute is put on a separate line. A typical attribute definition contains the attribute name and its data type.

The syntax for attributes is:

```
*[$|/]attribute_name:data_type=initial value
```

Explanation:

* Indicates a key attribute in a persistent class

\$ Indicates a class attribute

/ Indicates a derived attribute

attribute_name Identifies an attribute

data_type Standard type or the name of another class

Initial value Provides an initial value for the attribute

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

Scope

Attribute The scope of an attribute is the same as the scope of the [class](#) it is qualified by.

Data Type: The initial scope of data types is *Phase*. You can change the scope with [Item | Edit Scope...](#) to *System*.

Navigation

When you select a [class symbol](#) and then select [File | Open](#), the [Item dialog box](#) that appears lists all attributes and their data types. This is useful for [opening or creating an item-related diagram](#) for a data type that is a class.

Class

In: [Class Diagram Editor](#)

Subsections :

[Labels](#)

[Properties](#)

[Scope](#)

[Navigation](#)

A *CD class* symbol represents a class object in a Class Diagram. It is represented [unfolded](#) or [folded](#).

The Class name label of the symbol refers to an [item](#).

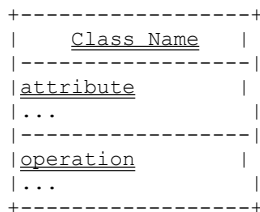
Some CD classes are used as special classes to generate supporting code for objects such as typedefs and enumerations. However, the use of special classes is different for each Code Generator, and you should read the User Guide of your Code Generator to see how special classes are supported.

Labels

[Edit Label Text](#)

[Specify Label Syntax](#)

A class symbol has three sections. Each of these has a label:



Properties

[Edit, Show, Delete properties](#)

Depending on the phase you are in, you can specify different properties for this symbol.

You can edit or see the currently available properties for a symbol with **Item | Edit Properties** or **Item | Show Properties**.

Scope

[Edit Scope](#)

The initial scope of a CD class is *Phase*. You can change the scope with **Item | Edit Scope...** to *System*.

Navigation

[Navigation - Diagram Editors](#)

You can open or create an item-related diagram by selecting an class symbol from the drawing area and select **File | Open**. This option is only available when your diagram is saved.

Class Name

In: Class Diagram Editor

Subsection :

Labels

The top section of a CD class symbol is reserved for the class name. The class name identifies a class uniquely.

Labels

The Class Name section contains only one class name, for example:

```
+-----+
|  Person  |
+-----+
```

The syntax for class names is:

```
/name
```

Explanation:

/ This can be used to indicate a derived class
name Assigning a name to a class is *compulsory*.

Constraint

In: Class Diagram Editor

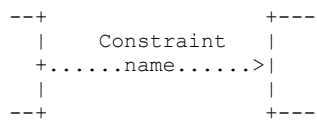
A *constraint* symbol is a connector indicating a general constraint between:

- CD class - CD class
- association - association
- CD class - association

What we call an *association* here can be an association, a qualified association, an aggregation or a qualified aggregation.

Labels

The constraint symbol has one label:



A Constraint Name can be used to specify details of the constraint. For instance, if one association linking two classes is a subset of another association linking the same classes, you can put a constraint symbol between these two associations and enter subset as the constraint name.

Generalization and Overlapping Generalization

In: [Class Diagram Editor](#)

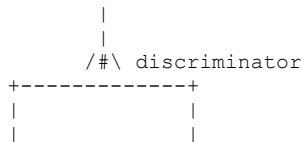
Subsections : [[Labels](#)] [[Properties](#)]

With a *Generalization* symbol you can establish a super-subtype relation between classes.

A generalization symbol consists of a node part and a connector part. The node part in a **generalization** symbol is an empty triangle, whereas in an **overlapping generalization** it is a black triangle.

Labels

The node part of the generalization symbol has the following label:



The name of the symbol refers to an item.

Properties

Depending on the phase you are in, you can specify different properties for this symbol. To see the currently available properties for a symbol, select the symbol in the drawing area of the diagram and then select Item > Edit Properties or Item > Show Properties.

More Classes (...)

In: [Class Diagram Editor](#)

A *More Classes* symbol indicates that more subclasses exist in a super-subtype relation.

Multiplicity

In: [Class Diagram Editor](#)

With the multiplicity buttons at the bottom of the control panel, you can specify the range of allowable integers for each side of an association between two classes:

[Association](#)

[Aggregation](#)

[Qualified Association](#)

[Qualified Aggregation](#)

[N-ary Association Connector](#)

The Multiplicity buttons are at the bottom of the control panel:

Begin Mandatory	1	1	End Mandatory
Begin Optional	0..1	0..1	End Optional
Begin Many	*	*	End Many

Note 1: during code generation all other multiplicity types, such as 1..*, 3..8, etc. are treated as Many multiplicity.

Note 2: For the N-ary Association Connector only the Begin <multiplicity> is used.

Note 2: the generation of code may differ for each code generator. Read the User Guide for your code generator. In general the following is true:

- If the multiplicity of the relation is mandatory, then a role name is mandatory too at that side of the relation.
- In optional multiplicity relation (optional on both sides of the relation, at least one role name is compulsory.

See also:

[Set multiplicity](#)

[Change multiplicity](#)

N-ary Association

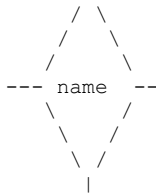
In: [Class Diagram Editor](#)

Subsections : [[Labels](#)] [[Properties](#)] [[Scope](#)]

An *N-ary Association symbol* is the node part of an n-ary association. It is connected to the appropriate [class symbols](#) by an [n-ary association connector](#).

Labels

The symbol representing the n-ary association node has the following label:



Assigning an *n-ary association* name is compulsory.

Note that n-ary associations are not supported in the standard code generation.

The name of the symbol refers to an [item](#).

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item > Edit Properties](#) or [Item > Show Properties](#).

Scope

The initial scope of an n-ary association is *File*. You can change the scope with [Item > Edit Scope...](#) to *System* or *Phase*.

N-ary Association Connector

In: Class Diagram Editor

Subsection : Labels

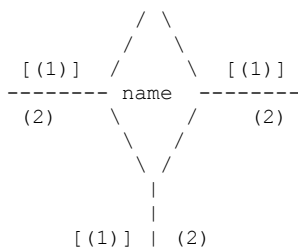
An *N-ary Association Connector* is the connector part of an n-ary association.

One side of this connector must be connected to a class symbol, the other side to an n-ary association symbol.

You can specify the multiplicity of an n-ary association through the multiplicity indicators in the control panel.

Labels

The connector representing the n-ary association connector has the following labels:



Explanation:

name The *name* of the n-ary association is specified in the corresponding n-ary association symbol.

(1) = *constraint* The multiplicity of an association can be restricted numerically with this label.

Depending on the multiplicity of the association, the following constraints are valid:

Constraint	Association
0,1	(optional)
0-1	(optional)
0,n	(many)
0-n	(many)
0+	(many)
1	(one)
n	(one or more)
m,p	(one or more)
m-p	(one or more)
m+	(one or more)
{ordered}	(all)

where $m > 0$, $n > 1$ and $p \geq m$

The square brackets are only visible when you are inserting something in the field.

(2) = *role* A role name is compulsory if the multiplicity of the association is mandatory. In optional multiplicity associations, at least one role name is compulsory.

The name and role labels of the symbol refer to items.

Class - Operations Section

In: [Class Diagram Editor](#)

Subsections : [[Labels](#)] [[Properties](#)] [[Scope](#)] [[Navigation](#)]

(In: [Class Diagram Editor](#))

The bottom section of a [CD class symbol](#) is reserved for operations. It can contain zero or more operations. A typical operation looks like this:

```
getNewNumber (name:char[80]):integer
```

When you enter operations for a class, the [Class Definition Matrix](#) for that class (CDM) is automatically created or updated.

Labels

The Operations section can contain one or more operations. Every operation is put on a separate line. A typical operation definition contains the method name, a parameter list and the method type:

```
|
|-----|
| select (p:Point):Boolean |
|-----+
|
```

The syntax for operations is:

```
$method_name(par_list):method_type{abstract}
```

Explanation:

\$ Indicates a class operation

method_name Identifies a method

(par_list) Consists of zero or more parameter definition(s). The syntax of a parameter definition is as follows:

```
(parameter_name:parameter_type)
```

The parameter_type can be a standard type or the name of another class

method_type Indicates method type: a standard type or the name of another class

{ abstract} Indicates an abstract operation in a superclass

Properties

In the Object Design phase, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

Scope

Operation : The scope of an operation is the same as the scope of the [class](#) it is qualified by.

Data Type: The initial scope of data types is *Phase*. You can change the scope with [Item | Edit Scope...](#) to *System*.

Parameter : The scope of a parameter is the same as the scope of the [class](#) it is qualified by.

Navigation

For the data types of operations and parameters you can open or create an item-related diagram by selecting a [class symbol](#) and select [File | Open](#). Select the name of the data type in the dialog box.

Propagation

In: Class Diagram Editor

Subsection :

Labels

A *propagation* is the automatic, successive application of an operation to an object associated to the object on which the operation was originally applied. Propagation is typically used in aggregations where a operation on the whole object results in operations on the parts.

The propagation symbol can be placed on or near an connector, and will move with the connector it belongs to. Before inserting a propagation symbol, you must connect two objects with an connector and then insert the propagation symbol by clicking near the connector. You can move the propagation symbol during insertion, to prevent it from overlapping its connector.

A propagation symbol can be attached to the following connectors:

- Association
- Qualified association
- Aggregation
- Qualified aggregation

Labels

The propagation symbol has the following label:

```
  +- operation  +--  
  | ----->  |  
  +-----+  
  |          |  
  |          |  
  +-+      +--
```

Assigning an *Operation* as label is *optional*.

The direction of the propagation leads from the original class to the subsequent class.

The name of the symbol refers to an item.

Qualified Aggregation

In: [Class Diagram Editor](#)

Subsections :

[[Labels](#)]

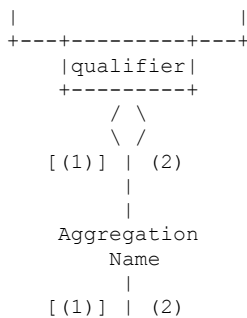
[[Properties](#)]

A *Qualified Aggregation* is an aggregation of which the effective multiplicity is reduced by a qualifier.

You can specify the multiplicity of a qualified aggregation through the multiplicity indicators in the control panel.

Labels

The connector representing the qualified aggregation has the following labels:



Explanation:

qualifier The qualifier is compulsory in a qualified aggregation. Note that the property data type for qualifiers is compulsory too.

Aggregation Name The name of the qualified aggregation.

(1) = *constraint* The multiplicity of a qualified aggregation can be restricted numerically with this label. Dependent on the multiplicity of the aggregation, the following constraints are valid:

Constraint	Association
0,1	(optional)
0-1	(optional)
0,n	(many)
0-n	(many)
0+	(many)
1	(one)
n	(one or more)
m,p	(one or more)
m-p	(one or more)
m+	(one or more)
{ordered}	(all)

where $m > 0$, $n > 1$ and $p \geq m$

The square brackets are only visible when you are inserting something in the field.

(2) = *role* A role name is compulsory if the multiplicity of the aggregation is mandatory. In optional multiplicity aggregations, at least one role name is compulsory.

The name and role labels of the symbol refer to items.

Properties

Depending on the phase you are in, you can specify different properties for this symbol. To see the currently available properties for a symbol, select the symbol in the drawing area of the diagram and then select Item | Edit Properties or Item > Show Properties.

Qualified Association

In: [Class Diagram Editor](#)

Subsections :

[[Labels](#)]

[[Properties](#)]

The *Qualified Association* consists of an association and a qualifier reducing the effective multiplicity of the association. It can be used to relate [class symbols](#) in Class Diagrams.

You can specify the multiplicity of a qualified association through the [multiplicity indicators](#) in the control panel.

Labels

The connector representing a qualified association has the following labels:

```
---+
  +-----+ [(1)]
  |qualifier|-----Association Name-----
  +-----+ (2)
  |
  +-----+ (2)
```

Explanation:

qualifier The qualifier is compulsory in a qualified association.

Association Name

The name of the qualified association.

(1) = *constraint* The multiplicity of a qualified association can be restricted numerically with this label. Depending on the multiplicity of the association, the following constraints are valid:

Constraint	Association
0,1	(optional)
0-1	(optional)
0,n	(many)
0-n	(many)
0+	(many)
1	(one)
n	(one or more)
m,p	(one or more)
m-p	(one or more)
m+	(one or more)
{ordered}	(all)

where $m > 0$, $n > 1$ and $p \geq m$

The square brackets are only visible when you are inserting something in the field.

(2) = *role* A role name is compulsory if the multiplicity of the association is mandatory. In optional multiplicity associations, at least one role name is compulsory.

The name and role labels of the symbol refer to [items](#).

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item > Edit Properties](#) or [Item > Show Properties](#).

Actor - symbol

In: [Collaboration Diagram Editor](#)

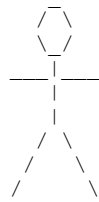
Subsection :

[Labels](#)

An *Actor* in a Collaboration Diagram represents the person, software, hardware, or other agent external to the system that is interacting with the system.

Labels

The Collaboration Actor is represented by a stick man figure and has the following label:



actorName

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

Qualified Aggregation Link

In: [Collaboration Diagram Editor](#)

Subsections :

[[Labels](#)]

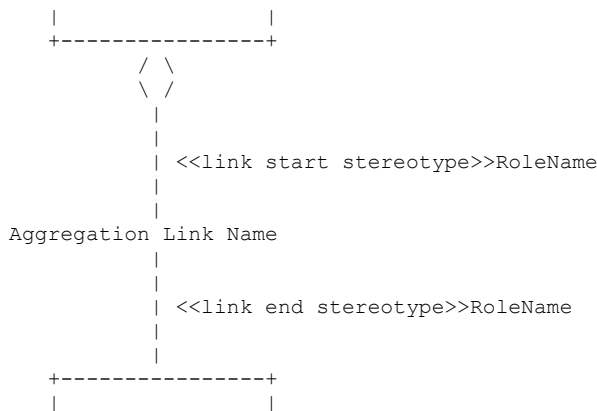
[[Properties](#)]

The *Aggregation Link* is an instance of an aggregation in a Class Diagram.

You can specify the stereotypes of an aggregation link through the [link stereotypes](#) in the control panel.

Labels

The connector representing an aggregation link has the following labels:



Explanation:

Aggregation Link Name

The name of the aggregation from which this aggregation link is instantiated. This name has class syntax.

An aggregation link can hold [messages](#) showing the information that flows along the link.

Link Stereotype

A [link stereotype](#) indicates a kind of implementation of the role.

Role Name

The name of the link roles. This label contains the name of a qualified data element. This name has data syntax:

RoleName :ClassifierName

RoleName You can specify a name for the role

ClassifierName For binary links the RoleName is qualified by the type of the Instance at the opposite side of the link.
For n-ary links the RoleName is qualified by the type of Instance it is linked to.

The name and role labels of the symbol refer to [items](#).

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol.

To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

Instance

In: [Collaboration Diagram Editor](#)

Subsections :

[[Labels](#)]

[[Properties](#)]

[[Scope](#)]

[[Navigation](#)]

An Instance is represented by a rectangle. It is an instantiation of a class in a [Class Diagram](#) or a Use Case in a [Use Case Diagram](#).

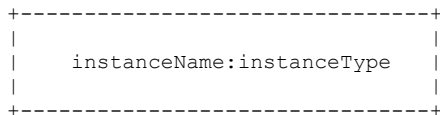
The name of the instance is underlined and has class syntax.

Labels

[Edit Label Text](#)

[Specify Label Syntax](#)

A class symbol has three sections. Each of these has a label:



instanceName the name of the instance

instanceType The instanceType can be [defined as an item](#) to correspond to a [class](#) or a [use case](#). (see [Properties](#) below)

Properties

[Edit , Show, Delete properties](#)

Depending on the phase you are in, you can specify different properties for this symbol.

You can edit or see the currently available properties for a symbol with **Item | Edit Properties** or **Item | Show Properties**.

Scope

[Edit Scope](#)

The initial scope of an instance is *Phase*. You can change the scope with **Item | Edit Scope...** to *System*.

Navigation

[Navigation - Diagram Editors](#)

You can open or create an item-related diagram by selecting an instance symbol from the drawing area and select [File | Open](#). This option is only available when your diagram is saved.

Link

In: [Collaboration Diagram Editor](#)

Subsection :

[Labels](#)

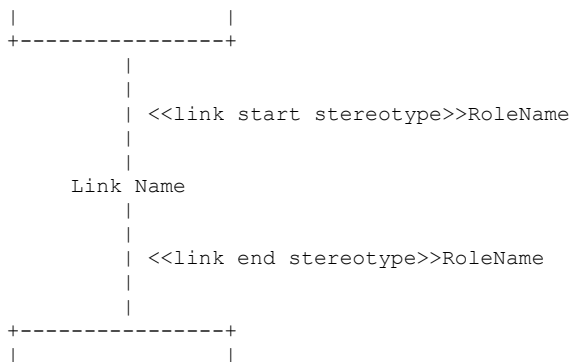
[Properties](#)

The link is an instance of an association in a Class Diagram [Instances](#) can be related to other instances and to and [Actors](#) by the *link* symbol.

You can specify the stereotype of a link through the [link stereotypes](#) in the control panel.

Labels

The connector representing the Link has the following labels:



Link Name The name of the association from which this link is instantiated. This name has class syntax. A Link can hold [messages](#) showing the information that flows along the link.

Link Stereotype A [link stereotype](#) indicates a kind of implementation of the role.

Role Name The name of the link roles. This label contains the name of a qualified data element. This name has data syntax:
`RoleName :ClassifierName`

RoleName You can specify a name for the role

ClassifierName For binary links the RoleName is qualified by the type of the Instance at the opposite side of the link.

For n-ary links the RoleName is qualified by the type of Instance it is linked to.

The name and role labels of the symbol refer to [items](#).

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

Message flow

In: Collaboration Diagram Editor

Subsection :

labels

A message flow carries a message from one object to another along a link (link, qualified link, aggregation link, qualified aggregation link, and n-ary link connector).

A message flow in a Collaboration Diagram has two characteristics:.

Message direction:

- Backward Messages
- Forward Messages

The direction in which the link was drawn determines the direction of the message.

Type of Messages)

- Flat Message
- Nested Message
- Asynchronous Message

Labels

----- (Type) Message Name ----- \

The default syntax for a message name is:

```
<predecessor> <guard-condition> <sequence-expression>  
<return-value> [ = | := ] <message-name> (<argument-list>)
```

Explanation:

< predecessor> <guard-condition>

A comma separated list of sequence numbers followed by a slash. The clause is omitted when the list is empty

< sequence-expression>

A dot separated list of number or names optionally followed by a condition or an iteration (both enclosed in square brackets, the iteration preceded by an asterisk) and separated from the rest of the label by a colon.

< return-value> [= | :=]

A data element with scope file. If the message does not return a value, then the return value and the assignment operator are omitted.

< message-name> A process element with scope system.

< argument-list>

A comma separated list of data elements with scope file. This list is enclosed in brackets and can be empty.

N-ary Link

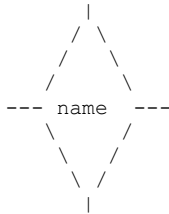
In: Collaboration Diagram Editor

Subsection :

[Labels]

An *N-ary Link symbol* is the node part of an n-ary link. It is connected to the appropriate instances and actors by an n-ary link connector.

Labels



The symbol representing the n-ary association node has the following label:

name the name of the n-ary link.

COD N-ary Link Connector

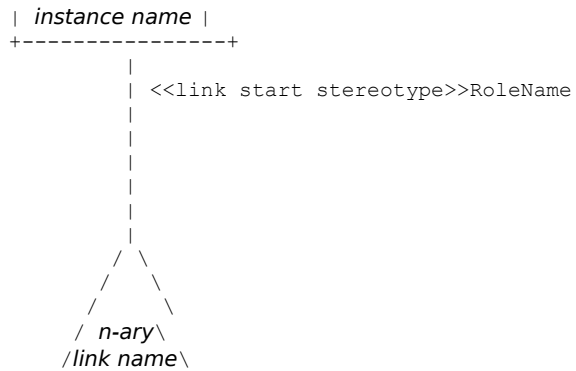
In: Collaboration Diagram Editor

An *N-ary Link Connector* is the connector part of an n-ary link.

One side of this connector must be connected to an instance symbol, the other side to an n-ary link symbol.

The symbol on the control panel is a small N-ary link symbol with lines at three of its corners. Labels

The connector representing the n-ary link connector has the following labels:



Link Stereotype	A <u>link stereotype</u> indicates a kind of implementation of the role.
Role Name	The name of the link roles. This label contains the name of a qualified data element. This name has data syntax: RoleName :ClassifierName
RoleName	You can specify a name for the role
ClassifierName	For binary links the RoleName is qualified by the type of the Instance at the opposite side of the link. For n-ary links the RoleName is qualified by the type of Instance it is linked to.

Qualified Aggregation Link

In: [Collaboration Diagram Editor](#)

Subsections :

[[Labels](#)]

[[Properties](#)]

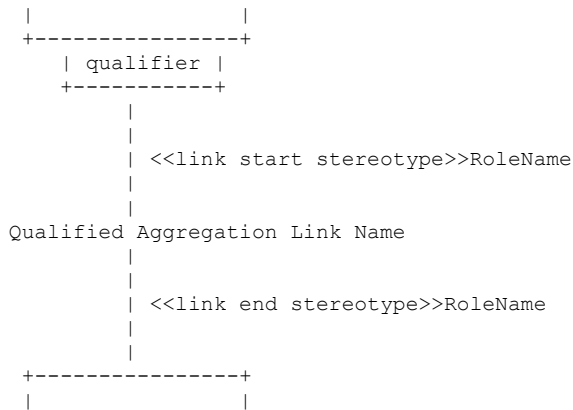
The *Qualified AggregationLink* consists of a aggregation link and a qualifier that represent the value of the qualifier in the aggregation from which this link is an instantiation.

The qualifier is stored as the *component property* **Qualifier Value** for a *Qualified Aggregation Link*. No items are generated from this value

You can specify the stereotypes of a qualified aggregation link through the [link stereotypes](#) in the control panel.

Labels

The connector representing a qualified aggregation link has the following labels:



Explanation:

Qualifier The qualifier is compulsory in a qualified link.

Qualified Aggregation Link Name

The name of the qualified aggregation from which this qualified aggregationlink is instantiated. This name has class syntax.

A qualified aggregation link can hold [messages](#) showing the information that flows along the link.

Link Stereotype A [link stereotype](#) indicates a kind of implementation of the role.

Role Name The name of the link roles. This label contains the name of a qualified data element. This name has data syntax:

RoleName :ClassifierName

RoleName You can specify a name for the role

ClassifierName For binary links the RoleName is qualified by the type of the Instance at the opposite side of the link.

For n-ary links the RoleName is qualified by the type of Instance it is linked to.

The name and role labels of the symbol refer to [items](#).

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol.

To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

Qualified Link

In: [Collaboration Diagram Editor](#)

Subsections :

[[Labels](#)]

[[Properties](#)]

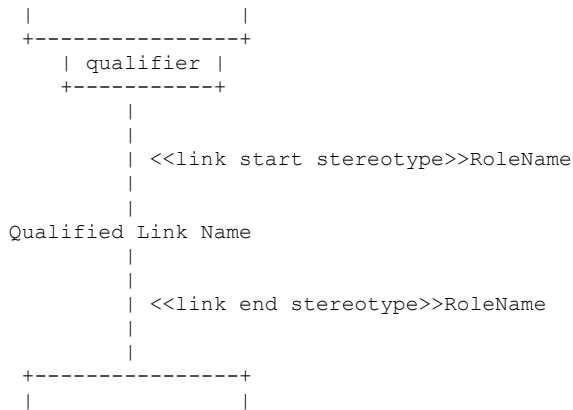
The *Qualified Link* consists of a link and a qualifier that represent the value of the qualifier in the association from which this link is an instantiation.

The qualifier is stored as the *component property* **Qualifier Value** for a *Qualified Link*. No items are generated from this value

You can specify the stereotypes of a qualified link through the [link stereotypes](#) in the control panel.

Labels

The connector representing a qualified link has the following labels:



Explanation:

Qualifier The qualifier is compulsory in a qualified link.

Qualified Link Name

The name of the qualified association from which this qualified link is instantiated. This name has class syntax.

A qualified link can hold [messages](#) showing the information that flows along the link.

Link Stereotype

A [link stereotype](#) indicates a kind of implementation of the role.

Role Name

The name of the link roles. This label contains the name of a qualified data element. This name has data syntax:

RoleName :ClassifierName

RoleName You can specify a name for the role

ClassifierName For binary links the RoleName is qualified by the type of the Instance at the opposite side of the link.

For n-ary links the RoleName is qualified by the type of Instance it is linked to.

The name and role labels of the symbol refer to [items](#).

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol.

To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

Link Stereotypes

In: Collaboration Diagram Editor

You can attach a **Stereotype** attached to a Link Role Name to show the type of implementation of the role.

You can set a Link Stereotype to both sides of a Link:

- The Link Start Stereotypes are situated at the bottom on the left side of the control panel.
- The Link End Stereotypes are situated at the bottom on the right side of the control panel.

You can choose from the following kinds of Link Stereotypes

- (none): omit the link stereotype
- association
- parameter
- global
- local
- self

Message Flow

In: Diagram Editors

A message flow is the sending of a message from one object to another. A message flow can be implemented in different ways in different diagrams.

Sequence diagram Message flows:

- Flat Message
- Nested Message
- Asynchronous Message
- Return Message

Collaboration Diagram:

- Backward /Forward Flat Message
- Backward /Forward Message
- Backward /Forward Asynchronous Message

Note - symbol

In: [Diagram Editors](#)

In addition to the Free Text property, you can add one or more *Notes* to components in a diagram. The note is used to add visible comments to your diagram.

The contents of the notes are not generated in a report.

You can create and edit notes like any component in a diagram. You can change the note font with *Options | [Font](#) | Annotation...* By default, the Annotation font is a smaller version of the Normal font

Notes can be connected to diagram objects and connectors with the *Note connector*. A single note can be connected to more than one object or connector in your diagram. However, a note cannot be connected to another note.

It is not possible to set properties for a note or note connector.

Note connector - symbol

In: [Diagram Editors](#)

The Note connector allows you to connect [Notes](#) to any object or connector in a diagram.

A single note can be connected to more than one object or connector in your diagram. However, a note cannot be connected to another note.

Vertex - symbol

In: [Diagram Editors](#)

The *vertex* can be used to make corners in connectors. In most editors, it is just a drawing aid and as such very useful as it offers you more flexibility in the placement and shape of connectors.

When you are drawing a connector, a vertex is automatically inserted each time you press the left mouse button with the cursor in the drawing area and not on the destination object.



Asynchronous Message (Sequence Diagram)

In: Sequence Diagram Editor

Subsection :

labels

The **Asynchronous Message** symbol is a message flow symbol represented by a half stick arrowhead.

It shows an asynchronous message between two objects.

Labels

-----Asynchronous Message Name----- \

The default syntax for this name is:

```
<message_name>[(argument_list)]
```

Explanation:

message_name Assign an asynchronous message name. This name has scope File.

(*argument_list*) The arguments in the argument list are separated by commas.

Nested Message (Sequence Diagram)

In: [Sequence Diagram Editor](#)

Subsection :

[labels](#)

The **Nested Message** symbol is a [message flow](#) symbol represented by a filled solid arrowhead.

It represents a procedure call or other nested flow of control. The nested sequence is completed before the outer level sequence resumes.

Labels

-----Nested Message Name-----|>

The default syntax for this name is:

```
<message_name>[(argument_list)]
```

Explanation:

message_name Assign an nested message name. This name has scope File.

(*argument_list*) The arguments in the argument list are separated by commas.

Initiator - symbol

In Sequence Diagram Editor

Subsection :

Labels

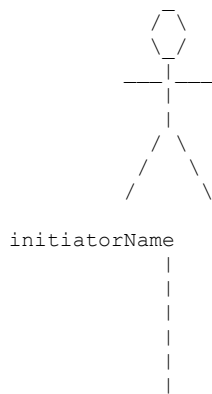
The initiator starts the sequence. It represents an external agent, such as a user, that interacts with the system. The initiator can be a class, but does not have to be. The following illustration shows an initiator:

Because a sequence describes a particular scenario occurring after one particular external event, each sequence can only have one initiator. However, you can draw more than one sequence in a diagram.

Labels

The Initiator has one label:

The Initiator is represented by a vertical line with a stick man figure on top and has the following label:



Initiator Name Assigning an *SD Initiator* name is *optional*.

The default syntax is:

< object_name>[:<object_class>],

For example: Smith:Patient or Patient

The name and type labels of the symbol refer to items.

In Scope Region - symbol

In Sequence Diagram Editor

Subsection :

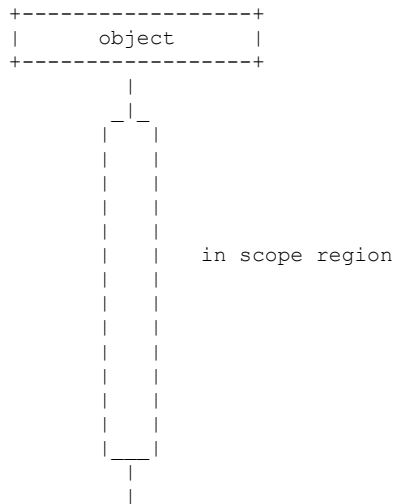
Labels

The In Scope region is a white rectangle that can be placed on top of an initiator or an object to show that the object is in scope.

When a message is nearby the symbol will be linked to that event. The In Scope region can be moved up and down the axis of the object, but not away from it. When it is moved up or down, the message(s) attached to it will be moved with it. However, when the message is moved, the In Scope region is not moved and you can connect the message to another object.

Labels

The In Scope region has no labels.



Object - symbol

In: [Sequence Diagram Editor](#)

Subsections :

[[Labels](#)]

[[Navigation](#)]

An *SD Object* symbol represents an object participating in a sequence of events between two or more objects. An SD Object refers to an object of a Class.

Labels

```
+-----+
| object_name:class_name |
+-----+
|
|
|
|
```

The default syntax is: <object_name>[:<object_class>]

Explanation :

objectName Assign an *Object Name*. This name has scope File.

className Assign *Class Name*. This name has scope Phase.

The *className* must correspond to the name of an existing [CD Class](#) or [Use Case](#).

The *object_name* is expected to be the *class_name* when no *class_name* is assigned.

Examples : Smith:Patient or :Patient.

The name and type labels of the symbol refer to [items](#).

Navigation

You can open or create an item-related diagram by selecting an SD Object symbol from the [drawing area](#) and select [File | Open](#). The [Select Operation](#) dialog box appears.

Object Termination - symbol

In Sequence Diagram Editor

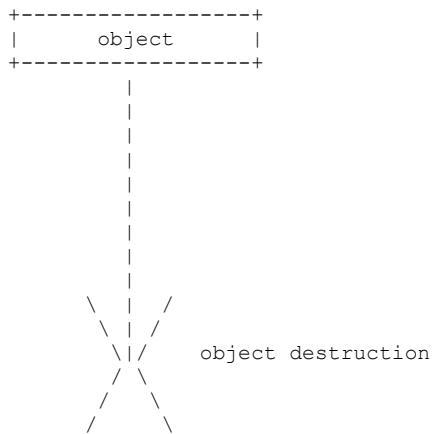
Subsection :

Labels

The *object termination* symbol shows the destruction of the object.

Labels

The object termination symbol has no labels.



Return Message

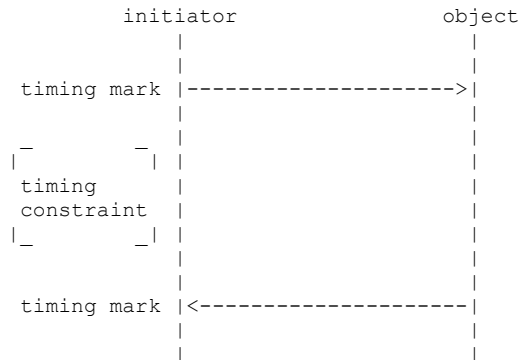
Sequence Diagram Editor A return message is a message that an object returns as a result of a process initiated by a message flow.

Timing Constraint and Timing Mark

In: Sequence Diagram Editor

A timing constraint is a text block tool that can be placed between two timing marks in a Sequence Diagram. The timing mark can be set at the beginning and end of each message. When you move your cursor close to the message begin or end position and to the opposite the side of the object where the message arrives or leaves, the text cursor appears, and you can type your text.

The timing constraint can be placed anywhere in the drawing area of the diagram, but usually it is positioned between two timing marks.



The use of timing constraints and timing marks is not compulsory. Timing constraints and marks can be used independently of each other.

Event Message - symbol

In: [State Transition Diagram Editor](#)

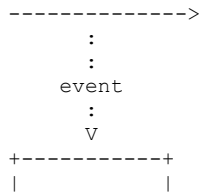
Subsections :

[[Labels](#)]

An *event message* symbol is a connector indicating an event sent to a [class](#). An Event Message connects an STD Class with a [transition](#).

Labels

An event message symbol has one label:



Assigning an *event* name is *optional*.

It can only be used to specify events of type *event* or *event (attribute)*.

The name of the symbol refers to a number of [items](#).

Navigation

You can open or create an item-related diagram by selecting an event message symbol from the [drawing area](#) and select [File > Open](#).

STD Final State - symbol

History State - symbol

In: State Transition Diagram Editor

Subsections :

[Labels]

A *STD Final State* symbol symbolizes the final state of a object in a State Transition Diagram Editor. It is represented by a bull's eye.

Labels

The Final State symbol has one label, which can be entered next to the symbol. Assigning this name is *optional*.

The name of the symbol refers to a number of items.

Start State - symbol

In: State Transition Diagram Editor

Subsections :

[Labels]

A *STD Start State* symbol symbolizes the initial state of a object in a State Transition Diagram. It is represented by a black dot.

Labels

The Start State symbol has one label, which can be entered next to the symbol. Assigning such a Start State Name is *optional*.

The name of the symbol refers to an item.

State - symbol

In: [State Transition Diagram Editor](#)

Subsections :

[[Labels](#)]

[[Navigation](#)]

A *state* represents an externally observable mode of behavior, i.e. an interval of time over which some behavior persists. During its life, an object passes through various states of behavior. As a state occupies an interval of time, it has duration.

The name of the state is the name of the behavior exhibited by the object.

Important : When a state has internal actions or transitions, a [State with internal actions](#) is used.

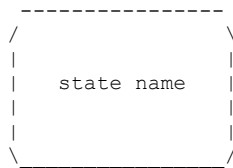
You can use [Edit | Replace](#) to replace a *state* with a *state with internal actions*.

Special kinds of states are:

- [Superstate](#)
- [Start State](#)
- [Final State](#)
- [History State](#)

Labels

The state symbol has one section:



Explanation:

State Name Assigning a State name is *compulsory*

The name of the symbol refers to an [item](#).

Navigation

You can open or create an item-related diagram by selecting a state symbol from the [drawing area](#) and select [File | Open](#).

State Class - symbol

In: [State Transition Diagram Editor](#)

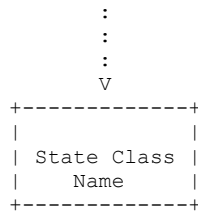
Subsections :

[[Labels](#)]

A *STD State Class* symbol represents a class receiving an event. It is connected to a [transition](#) by an [event message symbol](#).

Labels

The State Class has the following label:



Assigning a *State Class Name* is *compulsory*. This name must correspond to an existing [CD class](#).

The name of the symbol refers to an [item](#).

Navigation

You can open or create an item-related diagram by selecting a state class symbol from the [drawing area](#) and select [File | Open](#).

Transition - symbol

In: [State Transition Diagram Editor](#)

[[Labels](#)] [[Navigation](#)]

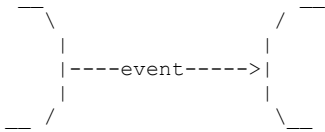
A *transition* symbol is a connector indicating an event. The object changes from one state to the other. Transitions connect:

- [Start States](#) (only as source)
- [States](#)
- [States with internal actions](#)
- [Super States](#)
- [Final States](#) (only as target)
- [History States](#) (only as target)

With the [complex transition node](#) you can diverge and converge transitions.

Labels

The transition symbol has one label:



event

Assigning an event to a Transition symbol is *optional*.
The following can be specified:

- *event*
- *event* (attribute)
- *event* / action
- *event* [guard]
- *event* / event2 (only as target)

The labels of the symbol refer to a number of [items](#).

Navigation

You can open or create an item-related diagram by selecting a transition symbol from the [drawing area](#) and select [File > Open](#).

Super State / Composite State- symbol

In: [State Transition Diagram Editor](#)

Subsections :

[[Labels](#)]

Super state / Composite state

A super state is also called a **composite state**. It is a way of generalizing parts of a diagram. The generalized part, the super state, can be treated from the outside as a single state.

By nesting a relatively autonomous part of a diagram in a super state, the number of transitions can be reduced, thereby clearing up the diagram.

State regions

A super state can be divided into [state regions](#) with the [concurrent state separators](#) .

Sub states

In the diagram, the super state is a rectangle with rounded corners. States and transitions inside the super state are called *sub states* and

Connecting super and sub states

Sub states can be connected to the outside via the super state or directly to another state.

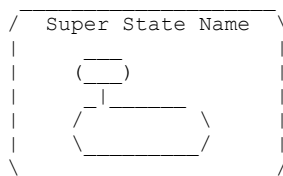
Transitions can be attached from the inside, i.e. from sub states to super state and from the outside, i.e. from a state external to the super state to a sub state in the super state

Super States do not support:

- Entry -action or activities
- Concurrent sub diagrams
- Control splitting or synchronization

Labels

The super state symbol represents the following:



Assigning a *Super State Name* is *optional*.

The name of the symbol refers to an [item](#).

Actor - symbol

In: [Use Case Diagram Editor](#)

Subsections :

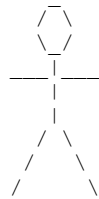
[Labels](#)

[Properties](#)

A *Use Case Actor* represents the person, software, hardware, or other agent external to the system that is interacting with the system.

Labels

The Use Case Actor is represented by a stick man figure and has the following label:



actorName

Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

By default the UCD actor has the following properties:

Type .

A field that contains one of the following values:

- user* (default) indicates that the actor represents an end user.
- system* indicates that the actor represents another system.

Initiator .

A field that is either true or false:

- true* indicates that this actor initiates the use case.
- false* (default) indicates that this actor participates in the use case, but does not initiate it.

Use Case - symbol

In: [Use Case Diagram Editor](#)

Subsections :

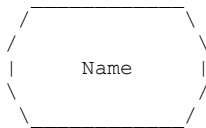
[Labels](#)

[Properties](#)

A *Use Case* represents the (information) system or part of the system with which an [actor](#) communicates.

Labels

The use case has the following labels:



Properties

Depending on the phase you are in, you can specify different [properties](#) for this symbol. To see the currently available properties for a symbol, select the symbol in the [drawing area](#) of the diagram and then select [Item | Edit Properties](#) or [Item | Show Properties](#).

By Default the use case has the following properties: **Basic Course of Action.**

A free text field in which you can describe the most common or important sequence of transactions for the use case.

Alternative Course of Action.

A free text field in which you can describe the variants on the basic course; for example, the sequence of transactions that occurs in the case of an error.

Precondition . A free text field in which you can describe the conditions that must be met before the use case can be performed.

Postcondition . A free text field in which you can describe the state of the system after the use case is performed.

Classification .

A field that contains one of the following values:

primary (default)

indicates that this use case represents the basic course of action.

secondary

indicates that this use case represents the alternative course of action.

Generalization - symbol

In: Use Case Diagram Editor

A *Use Case Generalization* represents communication between two use cases. The source use case includes the behavior of the destination use case.

- - - - - >

Changing Multiplicity

In: Class Diagram Editor

To change the multiplicity of an association:

1. In the drawing area, select the relation symbol you want to change the multiplicity of
 - Association
 - Aggregation
 - Qualified Association
 - Qualified Aggregation
 - N -ary Association Connector
2. Select a relation symbol from the control panel.
3. Select the appropriate Begin <multiplicity> and End <multiplicity> symbols from the control Panel (*where <multiplicity> stands for Mandatory, Optional or Many*).
4. Select Edit | Replace.
The multiplicity of the association is now changed.

Defining a Component as Item

Diagram Editors

A CD class becomes a Defined Item automatically when you create either an attribute or an operation, or when you navigate from that class. However, not all components become Defined Items automatically. When you cannot find the name of a component you created in the Browser object <defined items>, you can define it as a defined item.

To Define an Item:

1. Open a diagram.
2. Add a component to the diagram
3. Select the component.
4. Select **Item | Edit Properties**.
The Edit Properties dialog box appears.
5. In the left pane of the dialog box, select the item.
The item properties appear in the right pane.
6. Select the **Define Item** button to define the item.
7. If the item already exists, the Define Item button is disabled.

Alternative for classes

When you create a class in a CD, you often specify its attributes and operations before specifying its properties. When you add an attribute or operation to a class symbol, ObjectTeam defines the class item. Therefore, you rarely use the previous procedure to define class items.

Setting Multiplicity

In: [Class Diagram Editor](#)

See also: [Multiplicity](#)

To set the multiplicity of an association between classes:

1. Select the association symbol on the control panel:
 - [Association](#)
 - [Aggregation](#)
 - [Qualified Association](#)
 - [Qualified Aggregation](#)
 - [N -ary Association Connector](#)
2. Select the Begin <multiplicity> and the End <multiplicity> from the Multiplicity Buttons (*where <multiplicity> stands for :*)
 - 1 = Mandatory,
 - 0 ..1 = Optional
 - * = Many

Note . For the N-ary Association Connector only the Begin <multiplicity> is used.

3. Draw the association from one class to another or to the N-ary Association symbol.

Note . Use [Edit | Replace](#) if you want to change the multiplicity.

Class Diagram Editor

In: Diagram Editors

Use this diagram editor to draw Class Diagrams.

Window Sections

The default diagram editor window contains the following sections:

- Menu Bar
- Tool Bar
- Context Area
- Drawing Area
- Control Panel
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Collaboration Diagram Editor

In: Diagram Editors

Use this diagram editor to draw Collaboration Diagrams.

Window Sections

The default diagram editor window contains the following sections:

- Menu Bar
- Tool Bar
- Context Area
- Drawing Area
- Control Panel
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Navigation - Diagram Editors

Diagram Editors

Subsections :

Class diagram

Collaboration Diagram

Sequence diagram

State Transition diagram

Use Case Diagram

Use **File | Open** to open or create diagrams that refer to the same item. The anchor point is the label of the diagram object that refers to one or more items.

If you select such a symbol in the drawing area and select **File | Open** next, the **Select Operation dialog box** appears in which you can open an item-related diagram.

Depending on the circumstances, you may have to select which item and what type of operation (load diagram, make decomposition, create diagram, start editor) you are interested in.

You can open and create item-related diagrams from most diagram editors.

The navigation strategy between diagrams is defined in the following customization files:

- **opendefs .opendefs**
Use the Open Strategy Definition Editor to customize this file.
- **openlocs .openlocs**
Use the Open Strategy Availability Editor to customize this file.

Semantic Checking - Diagram Editors

In: [Diagram Editors](#)

The following table shows the semantic equivalence of diagram objects. The [Check Utilities](#) ensures that your model respects these semantics.

CD	class	class	operation	-	attribute
COD	-	-	-	-	-
SD	object	initiator	flow of control	qualified	-
			SD		
STD	class	class	event or event message	-	-
UCD	-	actor	communication association	use case	-
CCD	class	actor	message	-	-
MGD	-	-	message	-	-
DFD	actor or data store	-	data process, control flow, or update flow	-	data flow

Check Utilities

The OT Customization Guide has a complete list of the checks for each diagram.

You can use the following menus to check your diagrams:

In Browser:

- [Check | Contents](#)
- [Check | Local Model](#)
- [Check | Global Model](#)
- [Check | Use Case Model](#)

In Diagram Editors:

- [Check | Contents](#)
- [Check | Local Model](#)

Sequence Diagram Editor

In: Diagram Editors

Use the Sequence Diagram Editor (SDE) to draw Sequence Diagrams.
The objectives of this diagram technique are:

- To define messages between objects
- To define the sequence of the messages

Window Sections

The default diagram editor window contains the following sections:

- Menu Bar
- Tool Bar
- Context Area
- Drawing Area
- Control Panel
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Diagram Editors - Window

In: Diagram editors

The default diagram editor window contains the following sections:

- Menu Bar
- Tool Bar
- Context Area
- Drawing Area
- **Control Panel** - See each diagram editor for an overview of the Control Panel objects.
- **Message Area** - Displays system messages. You can browse through the message history with the arrow-up and arrow-down symbols at the right.

Communication Association - symbol

In: Use Case Diagram Editor

Properties

Navigation from Class Diagram

In: Diagram Editors

In a Class Diagram the following components allow opening or creation of other diagrams:

CD Class (Select Class Name in Item dialog box)

1. To **Class Diagram**
 - *Make decomposition* to create CD <class_name>.
or
 - *Load diagram* to edit CD <class_name>.
2. To **Collaboration Diagram**
 - *Create diagram* to create COD with qualifier <class_name> and assign <cod_name>.
and
 - *Start editor* to edit existing COD <class_name>:<cod_name>
3. To **Sequence Diagram**
 - *Create diagram* to create SD with qualifier <class_name> and assign <sd_name>.
and
 - *Start editor* to edit existing SD <class_name>:<sd_name>
4. To **State Transition Diagram**
 - *Create diagram* to create STD with qualifier <class_name> and assign <std_name>.
or
 - *Start editor* to edit existing STD <class_name>:top.

Navigation from Use Case Diagram

In: Diagram Editors

In a Collaboration Diagram the following components allow opening or creation of other diagrams:

COD Instance (Select Instance Type in Item dialog box)

1. To **Class Diagram**
 - *Create diagram* to create CD CD <instanceType_name>.
and
 - *Start editor* to edit CD <instanceType_name>.

Navigation from Sequence Diagram

In: Diagram Editors

In an Sequence Diagram the following components allow opening or creation of other diagrams:

SD Object (Select Object Type in Item dialog box)

1. To **Sequence Diagram**
 - *Make decomposition* to create SD with qualifier <objectType_name> and assign <sd_name>.
and
 - *Load diagram* to edit SD <objectType_name>:<sd_name>.
2. To **Class Diagram**
 - *Create diagram* to create CD <objectType_name> and assign <sd_name>.
or
 - *Start editor* to edit CD <objectType_name>:<sd_name>.
3. To **State Transition Diagram**
 - *Create diagram* to create STD <objectType_name>:top.
or
 - *Start Editor* to edit STD <objectType_name>:top.

Navigation from State Transition Diagram

In: Diagram Editors

In a State Transition Diagram the following components allow opening or creation of other diagrams:

STD State With Internal Actions (Select Event/Action in Item dialog box)

1. To **State Transition Diagram**
 - *Make decomposition* to create STD with qualifier <action_name> and assign <std_name>.
or
 - *Load diagram* to edit STD <action_name>:<std_name>.

Navigation from Use Case Diagram

In: Diagram Editors

In a Use Case Diagram the following components allow opening or creation of other diagrams:

UCD Use Case

1. To **Class Diagram**
 - *Make decomposition* to create UCD <useCase_name>.
or
 - *Load diagram* to edit UCD <useCase_name>.
2. To **Collaboration Diagram**
 - *Create diagram* to create COD with qualifier <useCase_name> and assign <cod_name>.
and
 - *Start editor* to edit existing COD <useCase_name>:<cod_name>
3. To **Sequence Diagram**
 - *Create diagram* to create SD with qualifier <useCase_name> and assign <sd_name>.
and
 - *Start editor* to edit existing SD <useCase_name>:<sd_name>

State with internal actions- symbol

In: [State Transition Diagram Editor](#)

Subsections : [[Labels](#)] [[Navigation](#)]

A state represents an externally observable mode of behavior, i.e. an interval of time over which some behavior persists. During its life, an object passes through various states of behavior. As a state occupies an interval of time, it has duration.

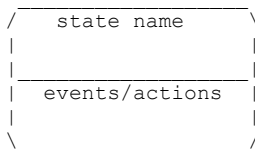
The name of the state is the name of the behavior exhibited by the object.

A *State with internal actions* symbol represents a state with activities and/or internal transitions.

You can use [Edit | Replace](#) to replace a *state* with a *state with internal actions*.

An *State with internal actions* symbol represents

Labels



Explanation:

State Name Assigning a State name is *optional* if events/actions are defined. If they are not defined, a State Name is *compulsory*.

events /actions Assigning events/actions is *optional* if a State Name is defined. If no State Name is defined, events/actions are *compulsory*.
Specified events/actions have one of the following forms:

Syntax	Example
entry/entry-action	entry/motor off
do/activity-A	do: reset item
event-1/action-1	coins in(amount)/add to balance
exit/exit-action	

The name of the symbol refers to an [item](#).

Navigation

You can open or create an item-related diagram by selecting a state symbol from the [drawing area](#) and select [File | Open](#).

STD State Region

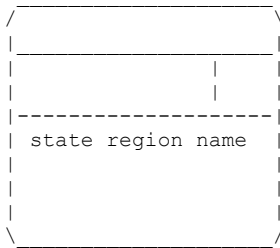
In: State Transition Diagram Editor

Subsections :

[Labels] A **state region** is created when you add concurrent state separators to a superstate. The superstate is then also called a composite state. The enclosed compartment is called a state region.

Labels

The State region has the following label:



state region name

Assigning an event to a state region is *optional*.

The labels of the symbol refer to a number of items.

Complex Transition Node - symbol

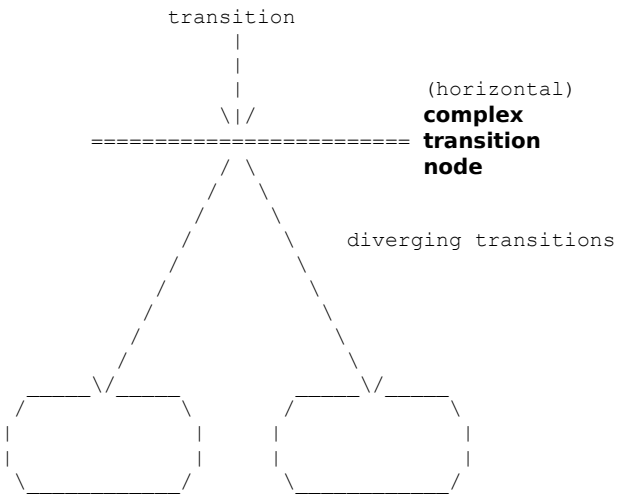
In: [State Transition Diagram Editor](#)

Subsections :

[[Labels](#)]

With the horizontal and vertical *Complex Transition Nodes* you can converge and diverge [transitions](#). Use the horizontal Complex Transition Node to converge/diverge more or less vertical transitions, and the vertical Complex Transition Node to converge/diverge more or less horizontal transitions.

Labels



Labels

name Assigning an event to a Complex Transition Node is *optional*.

The labels of the symbol refer to a number of [items](#).

Concurrent State Separator - symbol

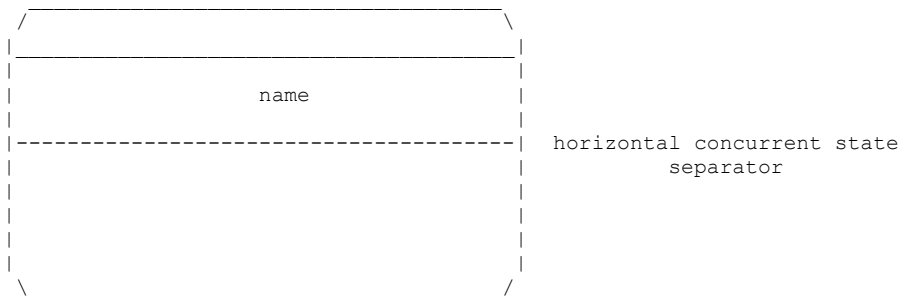
In: [State Transition Diagram Editor](#)

Subsections :

[[Labels](#)]

Use the horizontal and vertical *Concurrent State Separators* symbols to create concurrent state compartments in a [superstate](#).

Labels



Labels

You can assign a name to a *concurrent state* after you have divided a superstate into compartments using a concurrent state separator.

name Assigning an event to a Concurrent State compartment is *optional*.

The labels of the symbol refer to a number of [items](#).

History State - symbol

In: State Transition Diagram Editor

Subsections :

[Labels]

A *history state* is shown as a small circle with capital H and can only be inserted in a super state. Only one history state can exist in a one superstate.

The history state symbolizes the status quo of the super state and its sub states at the moment that an event is activated from the super state to a state external to the super state. When the process resumes at the superstate, you can return a transition to the history state in the superstate to resume at the level that the superstate had before it was left.

A history state can have multiple incoming events/actions. It usually does not have outgoing events/actions.

Labels

The history state has no label.

Class Diagram Item Properties dialog box

In : [Diagram Editors](#)

(See also: [Item Properties dialog box](#))

The Free Text property appears for all objects in all phases. This property allows you to enter unrestricted text about the object, like comments, motivations, and revision data. Other properties available in the Class Association Diagrams depend on the current phase.

Analysis Phase

- Data Type (qualifier). Specify the data type of the qualifier for a qualified association.
- Nullable (attribute, qualifier). Persistent code generators only. Specify whether the attribute or the qualifier for the qualified association can contain a null value. For more information, see the Properties appendix in the *ObjectTeam Code Generation Guide* for the code generator that you are using (C++, Informix NewEra, and so on).

System Design Phase

- Data Type (qualifier). Same as in the Analysis phase.
- Nullable (attribute, qualifier). Persistent code generators only. Same as in the Analysis phase.
- Persistent (class). Persistent code generators only. Specify whether the class represents persistent data. For more information, see the Properties appendix in the *ObjectTeam Code Generation Guide* for the code generator that you are using (C++, Informix NewEra, and so on).

Object Design Phase

- In the Object Design phase, the properties that you can specify in Class Diagrams in the Object Design phase affect code generation. Therefore, which properties appear depend on which code generator you are using.

For information about these properties: Refer to the Properties appendix in the *ObjectTeam Code Generation Guide* for the code generator that you are using (C++, Smalltalk, Informix NewEra, and so on).

Collaboration Diagram Item Properties dialog box

In: [Diagram Editors](#)

(See also: [Item Properties dialog box](#))

- Free Text (all objects). Enter unrestricted text about the object, like comments, motivations, and revision data.

Sequence Diagram Item Properties dialog box

In : [Diagram Editors](#)

(See also: [Item Properties dialog box](#))

- Free Text (all objects). Enter unrestricted text about the object, like comments, motivations, and revision data.
- Interaction Type (initiator, object). Specify the type of object: entity (a persistent object that survives the program execution), interface (an object with external I/O, that is an object that is activated from outside the system), or control (any other type of object).

State Transition Diagram Item Properties dialog box

In: [Diagram Editors](#)

(See also: [Item Properties dialog box](#))

- Free Text (all objects). Enter unrestricted text about the object, like comments, motivations, and revision data.

Use Case Diagram Item Properties dialog box

In: [Diagram Editors](#)

(See also: [Item Properties dialog box](#))

In Use Case Diagrams you can specify the following properties:

- Actor Type* Specify **system** as the Actor Type if the corresponding use case actor is a *system* actor. The default Actor Type of a use case actor is *user*.
- Alternative Course of Action* The Alternative Course of Action property allows you to enter unrestricted text that describes the corresponding use case. Use this property to describe variants on the basic use case scenario; for example, the sequence of transactions that occurs in the case of an error.
- Basic Course of Action* The Basic Course of Action property allows you to enter unrestricted text that describes the corresponding use case. Use this property to describe the most common or important sequence of transactions for the use case.
- Classification* Specify the value **secondary** if the corresponding use case describes the system's behavior under exceptional conditions. The default value **primary** indicates main line functions of the system.
- Initiator* Turn this property on to define the corresponding use case actor as *initiator* for a use case. By default, this property is turned off.
- Postcondition* The Postcondition property allows you to enter unrestricted text that describes the corresponding use case. Use this property to describe the state of the system after the use case is performed.
- Precondition* The Precondition property allows you to enter unrestricted text that describes the corresponding use case. Use this property to describe the conditions that must be met before the use case can be performed.

Tool Bar

In: Diagram editors

The *tool bar* is a window area that contains icons for quick activation of frequently used menu entries. Instead of executing such a menu entry from the menu bar, you can just click on the corresponding icon in the tool bar.

The tool bar is located between the menu bar and the context area. You can find out which menu entry a toolbar button represents by dragging your mouse pointer over the button in the toolbar and keeping it there for a few seconds. A ToolTip with the name of the menu entry appears.

The following tools have tool bars

The tool bar in the Diagram Editor contains the following buttons by default:

- File > Save
- Edit > Copy
- Edit > Delete
- Edit > ZoomOut
- Edit > ZoomIn

Component-Item relations

In : [Browser](#)

Subsections :

- Explanation of tables
- [CD Symbols](#)
- [CDM Symbols](#)
- [COD symbols](#)
- [SD Symbols](#)
- [STD Symbols](#)
- [UCD Symbols](#)

In the tables below the relation between components and items is defined.
The columns have the following meaning:

Symbol Type This groups the different types of symbols:

- Nodes - The symbols that represent conceptual elements of the diagram technique. These are the end points in the diagrams.
- Connectors - The connecting lines between two nodes.
- Connected Nodes - these are connected to a connector. They function as a connector attribute or a connection point of another node.

Component Name This is the internal name of the component.

Label This is the internal name of the label associated with a component. Most labels have a name, many have additional labels as well.

Item Type This is the type of the item associated with the symbol or the label. If there is no item type listed, the element is not an item in the repository. The types are:

- cl - class
- de - data element
- et - event trace
- pe - process element
- st - state

Item Scope This lists the scopes that are valid for the item. The letters mean:

- P - Phase
- S - System
- F - File
- Q - the item is qualified. Its scope is the same as its owner item.

The default value is printed in *italics*.

CD (item type cl)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	cad_class	name	cl	<i>P, S</i>
		attributes	-	-
		methods	-	-
	cad_container	name	cl	<i>P, S</i>
		attributes	-	-
		methods	-	-
	nary-association	name	cl	<i>P, S, F</i>

	link_attr_box	attributes	-	-
	more_classes	-	-	-
	generalization	name	de	S
	overlap_gen	name	de	S
Connectors	association	name	cl	P, S, F
		role_start	de	Q
		constr_start	-	-
		role_end	de	Q
		constr_end	-	-
	aggregation	name	cl	P, S, F
		role_start	de	Q
		constr_start	-	-
		role_end	de	Q
		constr_end	-	-
	qualif_assoc	name	cl	P, S, F
		role_start	de	Q
		constr_start	-	-
		role_end	de	Q
		constr_end	-	-
		qualifier	de	F
	qualif_aggr	name	cl	P, S, F
		role_start	de	Q
		constr_start	-	-
		role_end	de	Q
		constr_end	-	-
		qualifier	de	F
	nary_assoc_conn	role	de	Q
		constraint	-	-
	generalization_conn	-	-	-
	overlap_gen_conn	-	-	-
	loop	-	-	-
	constraint	constraint	-	-
Connected Nodes	propagation	name	pe	F
	link_attrib	name	de	Q
		type	cl	P, S
		modifiers	-	-
		colon	-	-
		init_value	-	-

CDM (item type cl)

Symbol Type	Component Name	Label	Item Type	Item Scope
Rows	attribute	name	de	Q
		type	cl	P, S
		modifiers	-	-
		colon	-	-
		init_value	-	-
	method	name	pe	Q
		type	cl	P, S

		modifiers	-	-
		left_parenth	-	-
		right_parenth	-	-
		colon	-	-
		constraint	-	-
=====				
Cells	parameter	name	de	S
		type	cl	P, S
		colon	-	-
		comma	-	-

COD (item type cl)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	cod_instance	name	cl	xxxxxxxxxxxx
	cod_actor	name	cl	xxxxxxxxxxxx
Connectors	cod_message	name	pe	xxxxxxx

SD (item type et)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	etd_object	editor_only	-	-
		name	de	S
		type	cl	P, S
		colon	-	-
	etd_initiator	editor_only	-	-
		name	de	S
		type	cl	P, S
		colon	-	-
Connectors	etd_event	event	pe	S
			de	S

STD (item type cl:pe)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	state	name	st	Q
		editor_only	-	Q
	super_state	name	st	Q
	std_class	name	cl	P, S
	start_state	name	st	Q
	final_state	name	pe	Q
			de	S
Connectors	transition	editor_only	-	-
		event	pe	Q
			de	S
		condition	-	-
		action	pe	Q
			de	S
	event_msg	event	pe	Q
			de	S

Connected	state_action	event	pe	Q
Nodes			de	S
		condition	-	-
		action	pe	Q
			de	S
	activity	name	pe	Q
		do	-	-

UCD (item type pe)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	use_case	name	cl	P
	ucd_actor	name	cl	P
Connectors	use_case_gen	-	-	-
	und_com_assoc	name	pe	S
	dir_com_assoc	name	pe	S

Changing Link Stereotypes

In: Collaboration Diagram Editor

To change the link stereotypes of a role name:

1. In the drawing area, select the link symbol for which you want to change the stereotypes of the role names.
 - Link
 - Aggregation Link
 - Qualified Link
 - Qualified Aggregation Link
 - N -ary Link Connector
2. Select the wanted <Link Start Stereotype> and/or <Link End Stereotype> from the Link Stereotype Buttons on the control panel. Note. For the N-ary Link Connector only the <Link Start Stereotype> is used.
3. Select Edit | Replace.
The link stereotype of the association is now changed.

Setting Link Stereotypes

In: [Collaboration Diagram Editor](#)

See also: [Link Stereotypes](#)

To set the link stereotypes of a role name:

1. Select the link symbol on the control panel:
 - [Link](#)
 - [Aggregation Link](#)
 - [Qualified Link](#)
 - [Qualified Aggregation Link](#)
 - [N -ary Link Connector](#)
2. Select the wanted <Link Start Stereotype> and <Link End Stereotype> from the Link Stereotype Buttons on the control panel. Note. For the N-ary Link Connector only the <Link Start Stereotype> is used.
3. Draw the link from one class to another or to the N-ary Link symbol.
Note . Use [Edit | Replace](#) if you want to change the link stereotypes.

Flat Message

In: Sequence Diagram Editor

Subsection :

labels

The **Flat Message** symbol is a message flow symbol represented by a stick arrowhead.

It shows the progression to the next step in a sequence.

Labels

-----Flat Message Name----->

The default syntax for this name is:

<message_name>[(argument_list)]

Explanation:

message_name Assign an flat message name. This name has scope File.

(*argument_list*)
The arguments in the argument list are separated by commas.

Message Flow

In: Sequence Diagram Editor

In the Sequence Diagram, you can draw a message from the initiator to an object or from one object to another. You can draw the message in any direction as long as it begins and ends on the initiator or an object.

When a message is drawn to the object name box or to the object destroy symbol, the message becomes a create or destroy message.

The Sequence Diagram has the following message flows:

- Flat Message
- Nested Message
- Asynchronous Message

For returning information about events that happened upon the sending of a message, you can also draw a:

- Return Message

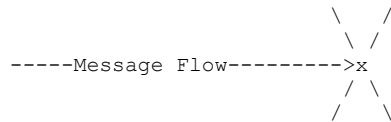
Create and Destroy Message

A message flow in a Sequence diagram can create and or destroy an object.

A **Create Message** is a message that connects to the *object name box of a new object*:



A **Destroy Message** is a message that ends in the object termination symbol of an object:



Asynchronous Message (Collaboration Diagram)

In: [Collaboration diagram](#), and [Message Flows in Collaboration Diagram](#)

The **Asynchronous Message** symbol is a *message flow* symbol represented by a half stick arrowhead. It shows an asynchronous message between two objects.

See [Message Flows in Collaboration Diagram](#) for the label syntax.

Flat Message (Collaboration Diagram)

In: [Collaboration diagram](#), and [Message Flows in Collaboration Diagram](#) The **Flat Message** symbol is a *UML message flow* symbol represented by a stick arrowhead. It shows the progression to the next step in a sequence.

See [Message Flows in Collaboration Diagram](#) for the label syntax.

Nested Message (Collaboration Diagram)

In: [Collaboration diagram](#), and [Message Flows in Collaboration Diagram](#) The **Nested Message** symbol is a *UML message flow* symbol represented by a filled solid arrowhead. It represents a procedure call or other nested flow of control.

The nested sequence is completed before the outer level sequence resumes.

See [Message Flows in Collaboration Diagram](#) for the label syntax.

Archive Cmd - dialog box

In: Corporate Management Tool

Archive Command:

Command	<p>Insert an archive command. You can add the following codes to the archive script</p> <ul style="list-style-type: none">• % P. Parent directory of the repository directory. This is an absolute pathname.• % N. Name of the Repository.• % R. Name of the Repository subdirectory in the parent directory. This is a single path component.• % O. Name of the project or model This code is only set to backup a project or model.• % S. Name of the project or model subdirectory in the repository. This code is only set to backup a project or model. This is a single path component.• % F. Full path of the archive file This is an absolute pathname.• % D. Directory part of the archive file This is an absolute pathname.• % T. File part the archive file• % W. Code to switch between archive and unarchive if your script contains both archive and unarchive options.
Browse Context	<p>You can use the Browse button to select an archive script. Select the context for the archive command:</p> <ul style="list-style-type: none">• None• Corporate .• Project

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Backup Repository dialog box

(In: [Corporate Management Tool](#))

The Backup utility of the Corporate Management tool archives a repository database, a single project or a model (groundworks).

In this dialog box you can select the backup method

Buttons

Dump ... Dumps the corporate, project, model tables into its DBDUMP directory in the destination directory.

- [The Dump Entire Corporate dialog box](#) appears.
- [The Dump Single Project dialog box](#) appears.
- [The Dump Model dialog box](#) appears.

Archive ... Creates an archive file from the corporate/project/model directory.

- [The Archive Entire Corporate dialog box](#) appears.
- [The Archive Single Project dialog box](#) appears.
- [The Archive Model dialog box](#) appears.

Note . For every backup/restore action it is a good idea to shutdown all servers during the restoration to maintain consistency between the database and file system parts of the repository. Click on the Shutdown button to stop all servers.

Change Repository dialog box

(In: [Corporate Management Tool](#))

Select **File > Change > Name** to change the name and the directory in which the file system part of the repository is stored. The dialog box that appears contains a Shutdown button and the fields described below.

You cannot change the repository name or directory if [dbservers](#) of the repository are running. Use the **Shutdown** button to shutdown current dbservers. If no error dialog appears, the servers are gone. (A **dbserver** cannot be shutdown if it has connected clients. You can use the [Client/Server Configuration tool](#) to determine which host servers still have clients running.)

Repository Name The name of the repository, which you can change in this dialog box.

Note: Each client has an M4_levelpath variable that is set to the name of the repository. When you change the name of the repository, all clients must change their M4_levelpath variables.

Parent Directory

The parent directory of the repository directory. You can change the name of the parent directory in this dialog box; the name of the repository directory always matches the name of the repository. (The repository directory is the file system part of the repository.)

Move repository directory to correspond with name

This option should always be selected. Deselecting it can cause inconsistencies in the repository. Selecting this option indicates that the repository directory should be renamed to match the new repository and parent directory names. If the repository name is changed, the repository directory name is also changed. If the parent directory is changed, the repository directory is moved to that parent directory.

Database information

The following fields appear in this group box:

- Database Name
(All repositories, except Oracle)
The name of the repository database.
- Database Server
(Informix repositories only)
The name of the computer on which the database software is running.
- Database Connect String
(Oracle repositories only)
The name of the connected database.
- Database User (Oracle and SSA repositories only)
For Oracle, the schema id of the Oracle user.
For SSA, the name of the SSA user.
- Database Password (Oracle and SSA repositories only)
Select the Enter button to open a dialog that allows you to enter the database password.
- Database Directory (SSA repositories only)
The name of the directory that holds the database.

Change Server Definition dialog box

(In: [Corporate Management Tool](#))

Select **File > Change > Server Definition** to edit the components of the entry of the selected object in the objservers file. A dialog box appears.

Note : You should never edit the objservers file manually; always use this option. Before making a change to the object servers file, the nameserver makes a backup in the same file but with a `.bak` suffix appended.

For changes to have any effect, running servers should be restarted. Use the **Shutdown** button in the dialog box to shutdown current servers. If no error dialog appears, all servers are gone and newly started servers will use the new server definition.

Use the dialog box to change the following components:

Executable Path The location of the **dbserver** executable. If it is not an absolute path, `M4_home/bin` is assumed as the location of the executable.

Host The computer on which dbserver runs.

Executable Name The name of the server as it occurs in the argument list for starting the server.

Database attributes The database name and other, DBMS-dependent attributes.

M4 Options This editable list can be used to add and remove M4 options from the server's command line. For example, adding an option `M4_orb_linger=300` will add the option `-M4_orb_linger=300` to the command line.

Delete Repository dialog box

(In: Corporate Management Tool)

Select **File > Delete** to delete all, or part of, a repository. A dialog box appears. Click the following check buttons to specify what you want to delete:

<i>Database</i>	The name of the database used in the selected repository.
<i>Directory</i>	The directory in which the file system part of the repository is stored.
<i>Server Entry</i>	The entry specifying the repository in the objservers file.

The Database Name and Server fields help you to identify the repository. They cannot be edited in this dialog box.

New Repository dialog box

In: Corporate Management Tool)

Select **File | New** to create a new repository. The dialog box that appears contains the following fields:

Corporate Name The name of the new browser object Corporate, the new top level in your browser hierarchy. The naming conventions for a new repository are the same as those for creating directories in the file system of the operating system you are using.

Make this repository the default repository.

Check this button if you want the new repository to be the default repository.

Directory ... This field specifies the directory in which the file system part of the repository is stored. You can enter a directory name in the Directory field by hand, or (in UNIX) you can click on the Browse button to locate the directory.

Database Name (All repositories, except Oracle)

A unique name for the database that will be created in the underlying RDBMS. Refer to your RDBMS documentation for naming conventions.

Database Server (Informix repositories only)

The name of the machine on which Informix is running. Make sure that the current permissions allow you to create new databases on this server.

Database Connect String

(Oracle repositories only)

If SQL*Net is installed, specify in this field the Oracle database specification string (ORACLE_STRING), e.g., oracle7.

If SQL*Net is not installed, enter `default` in this field and set the ORACLE_SID environment variable to the system identifier of the database that you want to use.

Database User (Oracle and SSA repositories only)

For Oracle, the schema id of the Oracle user.

For SSA, the name of the SSA user.

Database Password

(Oracle and SSA repositories only)

Select the Enter button to open a dialog that allows you to enter the database password.

Database Directory

(SSA repositories only)

The name of the directory to hold the database.

After you confirm the data in this dialog box with OK, **dbserver** will create the new repository. The output is displayed in a Monitoring Window.

After the repository is created successfully, the name server writes a new server entry in the **objservers** file using the corporate name you entered. This specifies how **dbserver** is started when a user tries to access the repository from an ObjectTeam tool.

Optimize Repository dialog box

(In: [Corporate Management Tool](#))

The Corporate Management tool lets you optimize your repository database.

You can optimize:

- Corporate tables by selecting the Corporate check box.
- Tables in all projects by selecting All Projects check box.
- Tables in selected projects by selecting projects from the Projects list box.

The following optimizations are possible:

- Drop indices
- Create indices
- Perform template optimizations

Note : **dboptimize** starts only if all observers can be shutdown. Therefore, all clients must be exited before you can optimize the repository.

Restore ... dialog box

In: Corporate Management Tool

The Restore utility of the Corporate Management tool unarchives an archive file produced by the Backup utility.

In this dialog box you can select the restore method

Buttons

Unarchive ... Unarchives a corporate, project or model archive file into its corresponding subdirectory.

- The Unarchive Entire Corporate dialog box appears.
- The Unarchive Single Project dialog box appears.
- The Unarchive Model dialog box appears.

Restore ... Restores the corporate/project/model tables from its corresponding DBDUMP directory in the destination directory.

- The Restore Entire Corporate dialog box appears.
- The Restore Single Project dialog box appears.
- The Restore Single Model dialog box appears.

Note . For every backup/restore action it is a good idea to shutdown all servers during the restoration to maintain consistency between the database and file system parts of the repository. Click on the Shutdown button to stop all servers.

Unarchive Command - dialog box

In: Corporate Management Tool

Unarchive Command:

Command	<p>Insert an unarchive command. You can add the following codes to the unarchive command</p> <ul style="list-style-type: none">• % P. Parent directory of the repository directory. This is an absolute pathname.• % N. Name of the Repository.• % R. Name of the Repository subdirectory in the parent directory. This is a single path component.• % O. Name of the project or model This code is only set to backup a project or model.• % S. Name of the project or model subdirectory in the repository. This code is only set to backup a project or model. This is a single path component.• % F. Full path of the archive file This is an absolute pathname.• % D. Directory part of the archive file This is an absolute pathname.• % T. File part of the archive file• % W. Code to switch between archive and unarchive if your script contains both archive and unarchive options.
Browse Context	<p>You can use the Browse button to select an unarchive command. Select the context for the unarchive command:</p> <ul style="list-style-type: none">• None• Corporate• Project

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.
Cancel Press this button to leave the dialog box without changes.

Change ORB Parameters dialog box

(In: [Client/Server Configuration Tool](#))

Select **File > Change > Parameters** to change parameters of the selected object. A dialog box appears displaying one or more ORB parameters, each with a slider with which the parameter can be changed. Press Apply to apply the changes to the dialog's object; press OK to apply the changes and close the dialog.

Which parameters can be changed depends on the type of node on which this operation is invoked:

Broker Node The only changeable parameter is orb_timeout.

Implementation Node

The parameters that can be changed are: orb_timeout, orb_linger, orb_report, orb_maxclients and orb_maxinstances. When these parameters are changed, they have effect on all servers of the implementation currently running, and on any new servers that are started.

Server Node The parameters that can be changed are orb_timeout, orb_linger and orb_report.

Parameter Display Mode dialog box

(In: [Client/Server Configuration Tool](#))

Select **Options > Parameter Display Mode** to change the way in which ORB parameters of implementations and servers are displayed in the information area.

A dialog box appears. Select one of the following values:

- Normal
Displays the parameters as name and value separated with a colon,
 - Meta4
Displays the parameters fit for inclusion in a Meta4UserEnv file,
 - Sh and Csh
Display the parameters fit for inclusion in a sh or csh script, respectively.
- The value of this option is stored in the M4 variable M4_csconfig_parammode.

Change Server Definition dialog box

(In: [Client/Server Configuration Tool](#))

Select **File > Change > Server Definition** to change the entry in the objservers file that corresponds to the object selected in the **Client/Server Configuration Tool**. A dialog box appears.

Note : Do not edit the objservers file manually; always use this option. Before making a change to the object servers file, the nameserver makes a backup in the same file but with a `.bak` suffix appended.

For changes to have any effect, running servers should be restarted. Use the **Shutdown** button in the dialog box to shutdown the current servers. If no error dialog appears, all servers are gone and newly started servers will use the new server definition.

Use the dialog box to change the following components:

Executable Path The location of the **dbserver** executable. If it is not an absolute path, `M4_home/bin` is assumed as the location of the executable.

Host The computer on which dbserver runs.

Executable Name The name of the server as it occurs in the argument list for starting the server.

Database attributes The database name and other, DBMS-dependent attributes.

M4 Options This editable list can be used to add and remove M4 options from the server's command line. For example, adding an option `M4_orb_linger=300` will add the option `-M4_orb_linger=300` to the command line.

Delete Server Definition dialog box

(In: Client/Server Configuration Tool)

Select **File > Delete** to delete the selected server definition from the object servers file.

A confirmation dialog box appears. Select Yes to delete the definition. A backup of the file is made by the nameserver in the same file but with a `.bak` suffix appended.

Shutdown Broker dialog box

(In: Client/Server Configuration Tool)

This option tries to shutdown the specified broker.

Shutdown proceeds immediately only if no servers, started by the broker, are still running. If there are still servers, shutdown is delayed until all servers have exited. While shutdown is delayed, the still running broker will not start any new servers.

Shutdown All Brokers dialog box

(In: Client/Server Configuration Tool)

This option tries to shutdown all brokers.

Shutting down will proceed immediately only if no servers, started by any broker, are still running. If there are still servers, shutdown is delayed until all servers have exited. While shutting down is delayed, the still running brokers will not start any new servers.

Shutdown Dbserver dialog box

(In: Client/Server Configuration Tool)

Shutdown of a **dbserver** will proceed immediately only if no clients exist. While shutdown is delayed, no new clients will be accepted.

Shutdown dialog box

(In: [Client/Server Configuration Tool](#))

Select **File > Shutdown** to shutdown the selected object. A confirmation dialog box appears. Select Yes to shutdown the object.

What is shutdown depends on the selected object:

ObjectTeam Node Shutting down this node will shutdown the entire ObjectTeam environment, meaning all running servers, including nameserver, brokers, **dbservers** and **lockserver**. All servers will be shutdown only if no clients are running. If clients are still running, the shutdown will proceed after all clients have exited.

Server Definition Node

Invoking shutdown will try to shutdown all servers that have been instantiated for the selected server definition, after a confirmation dialog has been satisfied.

Brokers Node Invoking shutdown will try to shutdown all brokers after a confirmation dialog has been satisfied.

Broker Node Shutting down will proceed immediately only if no servers, started by that broker, are still running. If there are still servers, shutdown is delayed until all servers have exited. While shutting down is delayed, the still running broker will not start any new servers.

Server Node Shutting down a server node will be preceded by a confirmation dialog, after which the server is shutdown. Shutting down servers has the following restrictions:

- For nameserver nodes, shutdown will proceed immediately only if no brokers are running and no object servers (e.g. a dbserver) are registered (else it is delayed until this is so). While shutdown is delayed, the nameserver will not allow new brokers to register themselves, nor new servers.
- For broker nodes, shutdown will proceed immediately only if no servers of that broker are running. While shutdown is delayed, no new servers will be started by that broker.
- For the lockserver node (there should be one lockserver at the most), shutdown will proceed immediately only if no locks exist. While lockserver shutdown is delayed, locks can still be acquired, since exiting clients or servers may need this to save their data. New clients, however, will not be accepted by the lockserver.
- For dbservers, shutdown will proceed immediately only if no clients exist. While shutdown is delayed, no new clients will be accepted.

Shutdown Implementation dialog box

Implementations are instances of brokers, such as Nameservers, **Lockservers** or **dbservers**. Shutting down an Implementation shuts down all instances of that implementation; for example, all dbservers of a corporate implementation.

See :

- [Shutting down a lockserver](#)
- [Shutting down a nameserver](#)
- [Shutting down a dbserver](#)

Shutdown Lockserver dialog box

(In: [Client/Server Configuration Tool](#))

Shutdown of a lockserver node (there should be one **lockserver** at the most), will proceed immediately only if no locks exist. While lockserver shutdown is delayed, locks can still be acquired, since exiting clients or servers may need this to save their data. New clients, however, will not be accepted by the lockserver.

Shutdown Nameserver dialog box

(In: [Client/Server Configuration Tool](#))

Shutdown of a nameserver will proceed immediately only if no brokers are running and no object servers (e.g. a **dbserver**) are registered (else it is delayed until this is so). While shutdown is delayed, the nameserver will not allow new brokers to register themselves, nor new servers.

Shutdown ObjectTeam dialog box

(In: Client/Server Configuration Tool)

This can be used to shutdown the entire ObjectTeam environment, i.e. all running servers, including nameserver, brokers, **dbservers** and **lockserver**. This option attempts to shutdown all servers. This will only succeed if no clients are running. If clients are still running, the shutdown will proceed after all clients have exited.

Shutdown Service dialog box

(In: [Client/Server Configuration Tool](#))

Invoking shutdown will try to shutdown all servers that have been instantiated for the selected server definition.

Change Lock Filter dialog box

(In: [Lock Management Tool](#))

To filter the locks displayed in the [Lock Management Tool](#): select **Options > Lock Filter**, fill in the following fields of the **Change Lock Filter** dialog box, then press OK or Apply. (OK dismisses the **Change Lock Filter** dialog box; Apply leaves it open.)

<i>Lock Type</i>	Selects only locks of the specified types. Possible types are read or write.
<i>Hanging Locks</i>	Selects only <u>hanging locks</u> . These are the only locks that can be removed.
<i>Lock Attributes</i>	Allows you to enter the following attributes: <ul style="list-style-type: none">• <i>Object Id</i> Selects only locks on objects with this identity.• <i>Host</i> Selects only locks that have been set by clients on this host.• <i>User</i> Selects only locks that have been set by clients started by this user.• <i>Process Id</i> Selects only locks that have been set by clients with this process id.• <i>Reason</i> Selects only locks that have been set for this reason. The specified string may contain a glob-style pattern.

To fill in the attributes easily, you can drag and drop diagram objects from an ObjectTeam browser onto the rectangular area (drop zone) to the right of the lock attributes. This fills in the Object Id field with the identity of the dropped object. Client nodes from the Client/Server Configuration tool can also be drag and dropped onto the drop zone. This fills in the Host, User and Process Id fields.

The lock filter is saved in the M4 variable `M4_lock_filter`. This variable contains a comma-separated list. The first element is a combination of the letters r, w, or a that indicates whether read, write or all locks should be selected. The second element is a 0 or 1 indicating whether only hanging locks should be selected. The remaining elements are strings for each of the lock attributes (object id, host, user, process id, and reason).

Remove Locks dialog box

(In: [Lock Management Tool](#))

File > Delete and the **Remove Locks** dialog box allow you to remove the selected lock(s). Only hanging locks can be removed. They can only be removed by the user (or broker) that started the lock server.

Hanging Lock

Normally , a client requests a lock from a **dbserver**, which requests the lock from the **lockserver**. When the client exits, the dbserver frees the locks requested by the client. When the dbserver exits, the lockserver frees the locks requested by the dbserver.

A *hanging lock* is a lock held by a client that the lockserver does not know about. Hanging locks are created as follows:

1. A client acquires a lock from a dbserver. The dbserver acquires the lock from the lockserver.
2. The dbserver exits abnormally so the lockserver frees the locks acquired by the dbserver. However, the client still holds the lock that it acquired from the dbserver.
3. The client exits abnormally before a new dbserver can be started.

The client lock is now a *hanging lock*: the new dbserver cannot free the lock and the lockserver does not know about the lock.

New Lock dialog box

(In: [Lock Management Tool](#))

File > New and the **New Lock** dialog box allow you to set a lock. Locks can only be set by the user (or broker) that started the **lockserver**.

To make it easier to fill in the required attributes, diagram objects from an ObjectTeam browser can be drag and dropped onto the rectangular area (drop zone) to the right of the lock attributes in the **New Lock** dialog. This fills in the Object Id field with the identity of the dropped object.

Client nodes from the Client/Server Configuration tool can also be drag and dropped onto the drop zone of the **New Lock** dialog. This fills in the following dialog fields: Host, User and Process Id.

Note : Client nodes can also be dropped into the lock list of the [Lock Management Tool](#), causing the **New Lock** dialog to appear with some of its fields filled in. However, this feature should be used after a lockserver crash as a final attempt to avoid losing data that has been entered in an editor.

Shutdown Lockserver dialog box

(In: [Lock Management Tool](#))

Select **File > Shutdown Lockserver** to shutdown the **lockserver** process.

A confirmation dialog box appears. Select Yes to shutdown the lockserver process. This succeeds immediately only if no locks exist. If locks still exist, the lockserver is marked as shutting down and exits after all locks have been released.

Remove External Files dialog box

(In: User Environment Tool)

Select **File > Delete** to delete an external file. A dialog box appears.

Selecting Yes deletes the selected files from the file system (but not from the repository). If the file is working, you can select Upload to copy the file from the file system into the repository.

Upload External File Versions dialog box

(In: User Environment Tool)

Select **File > Upload** to upload the selected file(s) from the file system to the repository.

A confirmation dialog box appears. Select Yes to upload the selected external file version(s).

Broker Node

(In: [Client/Server Configuration Tool](#))

This node represents a broker running on some host. The host can be retrieved from the node's name (after the @ sign). It is a child node of the Brokers node.

Available operations are [File > Change > Parameters](#) and [File > Shutdown](#). The only changeable parameter is orb_timeout.

Selecting shutdown on a broker displays a confirmation dialog. Shutting down will proceed immediately only if no servers, started by that broker, are still running. If there are still servers, shutdown is delayed until all servers have exited. While shutting down is delayed, the still running broker will not start any new servers.

The information area displays process information, broker information and ORB parameters (see the [Server Node](#) for details).

Brokers Node

(In: Client/Server Configuration Tool)

This is the parent of all broker nodes. It is a child node of the ObjectTeam node.

The only operation available on this node is File > Shutdown, which will try to shutdown all brokers after a confirmation dialog has been satisfied.

When this node is selected, the information area displays how many brokers are currently running, and if there are any brokers that are not responding.

Client Node

(In: Client/Server Configuration Tool)

This node represents a client of a server. (Only the lockserver and dbservers have clients.) The client node is a child node of a Server node. If the node has an icon that appears brighter than the icons of other client nodes, it represents the Client/Server Configuration tool itself. Note that one client can occur as more than one client node under the various servers.

There are no operations possible on a client node. However, you can drag and drop a client node to the lock list of a Lock Management tool or the drop zone of the New Lock dialog box. This (opens and) fills in the user name, host and process id fields of the **New Lock dialog box**.

When you select a client node, the information area of the Client/Server Configuration Tool displays the client's process info, and the name of server to which the client is connected.

Database Server Node

(In: Client/Server Configuration Tool)

A **dbserver** node is an instance of a server node.

Implementation Node

(In: [Client/Server Configuration Tool](#))

This represents a server definition for which the parent broker has started one or more servers. It is a child node of a [Broker node](#).

The only operation available is [File > Change > Parameters](#). The parameters that can be changed are: orb_timeout, orb_linger, orb_report, orb_maxclients and orb_maxinstances. When these parameters are changed, they have effect on all servers of the implementation currently running, and on any new servers that are started.

Information displayed when an implementation node is selected: implementation info (same information as shown when a [Server Definition node](#) is selected) and ORB parameters.

Lock Server Node

(In: Client/Server Configuration Tool)

A lock server node is an instance of a server node.

Name Server Node

(In: Client/Server Configuration Tool)

A nameserver node is an instance of a server node.

Object Servers Node

(In: [Client/Server Configuration Tool](#))

This node represents the object servers file and is a child node of the ObjectTeam node. The only operation available on this node is [File > Reload](#). The information area displays the location of this file and the owner of the file.

ObjectTeam Node

(In: Client/Server Configuration Tool)

This is the root node of the Client/Server Configuration tree. It can be used to shutdown the entire ObjectTeam environment, meaning all running servers, including nameserver, brokers, **dbservers** and **lockserver**.

If you select File > Shutdown on this node, a confirmation dialog is opened, then the tool attempts to shut down all servers. This will only succeed if no clients are running. If clients are still running, the shutdown will proceed after all clients have exited.

When this node is selected, the information area displays various information about the ObjectTeam environment. This includes the product package, version and date, the repository DBMS used, M4 home location, and the Meta4UserEnv file that is currently used. All client/server related M4 variables currently set are shown as well.

Server Definition Node

(In: Client/Server Configuration Tool)

This represents one server definition from the object servers file. It is a child node of the Object Servers node. Available operations are: File > Delete, File > Change > Server Definition and File > Shutdown.

Invoking shutdown will try to shutdown all servers that have been instantiated for the selected server definition, after a confirmation dialog has been satisfied.

The information area displays the server definition: implementation name, id (in the form *server id.server version*), implementation policy, protocol used when communicating with servers for this definition, location of executable, command line used to start a server, and the name of the host on which the server runs.

Server Node

(In: [Client/Server Configuration Tool](#))

This node represents a server started by some broker. It is a child node of an Implementation node. The nameserver, brokers, **lockserver** and **dbservers** are all nodes like this, differing only in their icon in the tree.

Note that brokers are not displayed as children of the **ot_broker** implementation, since the broker nodes are already accessible as children of the [brokers node](#). Possible operations on server nodes are [File > Change > Parameters](#) and [File > Shutdown](#).

The only parameters that can be changed are orb_timeout, orb_linger and orb_report.

Selecting shutdown on a server node displays a confirmation dialog, after which the server is shutdown. Shutting down servers has the following restrictions:

- For nameserver nodes, shutdown will proceed immediately only if no brokers are running and no object servers (e.g. a dbserver) are registered (else it is delayed until this is so). While shutdown is delayed, the nameserver will not allow new brokers to register themselves, nor new servers.
- For broker nodes, shutdown will proceed immediately only if no servers of that broker are running. While shutdown is delayed, no new servers will be started by that broker.
- For the lockserver node (there should be one lockserver at the most), shutdown will proceed immediately only if no locks exist. While lockserver shutdown is delayed, locks can still be acquired, since exiting clients or servers may need this to save their data. New clients, however, will not be accepted by the lockserver.
- For dbservers, shutdown will proceed immediately only if no clients exist. While shutdown is delayed, no new clients will be accepted.

When a server node is selected, the information area displays process information (host name, process id and name of the user that started the process), server information (name and id of implementation, port number used in communication with the server, time the server has been running, number of requests the server has handled, and the busyness of the server, which is a number that becomes larger when the server becomes busier) and ORB parameter settings.

Lock

(In: [Lock Management Tool](#))

Locks are used to:

- prevent concurrent update of objects; for example, diagrams
- guarantee that an object is not changed while processing
- synchronize caches

So , there are three kinds of locks: write locks, read locks and cache locks.

Read locks are used by clients. Read and write locks for a client are requested by the server automatically; for example, when a client does an edit operation, a write lock is set. Cache locks are used by the server and are not accessible from the Lock Management Tool.

Only one process can have a write lock. While one process has a write lock on an object, no other processes are allowed to have either a read lock or write lock on the same object.

Multiple processes can have a read lock on the same object at the same time. While one or more processes have a read lock on an object, all requests for a write lock on the same object are refused.

External File

(In: [User Environment Tool](#))

An external file is a file version for which a file exist in the client's file system. This is the file system that is accessible from the host on which the User Environment tool is started. Examples are generated code files and generated document files.

Working external file versions are not included when a repository is backed up. However, you can upload all working files, so that the repository contains the right versions.

Client/Server Configuration Tool

(In: [Repository Tool](#))

This tool can be used to examine and change the current client/server configuration of an ObjectTeam environment. Changing the configuration is restricted.

The tool is divided in two parts, on the left a tree containing an overview of the ObjectTeam environment, and on the right an information area that shows information on the object currently selected in the tree. The tree has at its root an ObjectTeam node, with always two child nodes, namely a node called [object servers](#), representing the object servers file, and a node called [brokers](#), representing all brokers that are currently running. Nodes in this tree can be classified by looking at their icon: a rectangular icon indicates a definition or implementation, while a diamond-shaped icon represents a process.

Menu Bar

Below are the default menu items. You can add new menus and redefine existing ones by using the [Customization Editor](#).

[[File](#)] [[View](#)] [[Options](#)] [[Help](#)]

- **File** menu
 - [Delete ...](#)
 - Reload
 - Reads the object servers file. This item is only available if the [object servers node](#) is selected. It is necessary only if the object servers file has been edited manually, and you should not edit the object servers file manually (use [File > Change > Server Definition](#) instead).
 - Change
 - [Server Definition...](#)
 - [Parameters ...](#)
 - [Shutdown ...](#)
 - Exit
- **View** menu
 - Refresh
 - Refresh Selected
 - ToolBar
 - Context Area
 - Message Area
- **Options** menu
 - [Parameter Display Mode...](#)
 - Font
- **Help** menu
 - [What 's This?...](#)
 - [On Help](#)
 - [Help Topics](#)
 - About ...

Corporate Management Tool

(In: [Repository Tool](#))

This tool can be used to work with an entire repository. For example, use this tool to create, delete, backup, or restore a repository.

The *Repository* drop-down list above the display area of the tool allows you to select the repository that you want to view.

Menu Bar

Below are the default menu items. You can add new menus and redefine existing ones by using the [Customization Editor](#).

[[File](#)] [[View](#)] [[Options](#)] [[Help](#)]

- **File** menu
 - [New ...](#)
 - [Delete ...](#)
 - Open
 - Change
 - [Name](#)
 - [Server Definition](#)
 - [Optimize ...](#)
 - [Backup ...](#)
 - [Restore ...](#)
 - Exit
- **View** menu
 - Refresh
 - ToolBar
 - Context Area
 - Message Area
- **Options** menu
 - Font
 - [Archive Command](#)
 - [Unarchive Command](#)
- **Help** menu
 - [What 's This?...](#)
 - [On Help](#)
 - [Help Topics](#)
 - About ...

Lock Management Tool

(In: [Repository Tool](#))

This tool can be used to examine, set or remove locks. (Locks can only be set or removed by the user (or broker) that started the lock server.)

The tool displays a list with all locks selected using the current lock filter. The following information is displayed for each lock: object on which lock is placed (if the object is a Version its name, type and version is shown, if it is a Versionable its name and type is shown), object type (read or write lock; this is also indicated using the lock's icon), user id of the client that has set the lock, host on which the client runs, process id of the client, date at which lock was created, and reason for the lock.

Menu Bar

Below are the default menu items. You can add new menus and redefine existing ones by using the [Customization Editor](#).

[[File](#)] [[Edit](#)] [[View](#)] [[Options](#)] [[Help](#)]

- **File** menu
 - [New ...](#)
 - [Delete ...](#)
 - Make Lockserver Operable
When a **lockserver** is started, if it detects that the previous lockserver exited abnormally (for example, due to a hardware failure), it will not create new locks for any client. This is done in case existing clients with a lock on a diagram are still running. (In this case, the lockserver could mistakenly grant a lock for that same diagram.)
Use the [Client/Server Configuration Management Tool](#) to ensure that no clients are running, then use this menu item to make the lockserver operable.
 - [Shutdown Lockserver](#)
 - Exit
- **Edit** menu
 - Select All
 - Deselect All
 - **View** menu
 - Refresh
 - ToolBar
 - Context Area
 - Message Area
 - **Options** menu
 - Font
 - [Lock Filter...](#)
- **Help** menu
 - [What 's This?...](#)
 - [On Help](#)
 - [Help Topics](#)
 - About ...

ObjectTeam Repository Tool

The repository tool is a collection of tools that allow the administrator (and in some cases, the user) to configure his or her ObjectTeam environment.

It is assumed that one and the same user owns all of these parts:

- repository directory (file system part of the repository)
- repository database (relational database part of the repository)
- object servers file
- server processes (nameserver, brokers, lockserver)

If this is not the case, the repository tool should be run by the user who owns the part that is intended to be modified. This implies that the options that modify several parts can only be used if these parts are owned by the same user. To prevent inconsistencies between these parts, it is not allowed to modify these parts individually.

A subtool can be started by double-clicking its icon, or by selecting its icon and calling File > Open.

You can start the following tools from the ObjectTeam Repository Tool:

- Corporate Management Tool
- User Environment Tool
- Lock Management Tool
- Client /Server Configuration Tool

User Environment Tool

(In: [Repository Tool](#))

The User Environment tool can be used to get an overview of all external file versions in a repository. The tool displays all those external file versions for which a file exists in the client's file system (this is the file system that is accessible from the host on which the User Environment tool is started). This information is of use when a repository needs to be backed up, since working external file versions will not be included in this backup. You can use this tool to upload all working files so that the repository contains the right versions.

Above the display portion of the window, are three drop-down lists:

- *Repository*
Allows you to select a repository.
- *Project*
Allows you to filter the display so that only the external files of one project are listed.
- *Config*
Allows you to filter the display so that only the external files of one configuration (in one project) are listed.

After selecting a different Repository, Project, or Configuration, select View > Refresh to update the display area.

The following information is displayed for each file: full path name, object name, version name, file version status, owner of file, size of the file project name, and configuration version name. Double-click on a file in the list to start a viewer for it.

Menu Bar

Below are the default menu items. You can add new menus and redefine existing ones by using the [Customization Editor](#).

[[File](#)] [[Edit](#)] [[View](#)] [[Options](#)] [[Help](#)]

- **File** menu
 - [Delete ...](#)
 - Show
 - [Upload ...](#)
 - Exit
- **Edit** menu
 - Select Working
 - Select All
 - Deselect All
- **View** menu
 - Refresh
 - ToolBar
 - Context Area
 - Message Area
- **Options** menu
 - Font
- **Help** menu
 - [What 's This?...](#)
 - [On Help](#)

- Help Topics
- About ...

Set Default Repository dialog box

In: Corporate Management Tool)

The Set Default Repository box asks you to confirm if the selected repository should be made default. It also shows the changes its makes to the M4_levelpath variable.

Dump Single Project dialog box

In: Corporate Management Tool

Project name Fill in the Project name.
Shutdown observers before dumping
 Activate this checkbox if desired.

Database

Name (All repositories, except Oracle)
 The name of the project database.
Server (Informix repositories only)
 The name of the computer on which the database software is running..
Database Connect String
 (Oracle repositories only)
 The name of the connected database.
Database User (Oracle and SSA repositories only)
 For Oracle, the schema id of the Oracle user.
 For SSA, the name of the SSA user.
Database Password
 (Oracle and SSA repositories only)
 Select the Enter button to open a dialog that allows you to enter the database password.
Database Directory
 (SSA repositories only)
 The name of the directory that holds the database.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.
Cancel Press this button to leave the dialog box without changes.

Unarchive Entire Corporate dialog box

In: Corporate Management Tool

- Source Type in the source or use the Browse button to insert it.
- Command By default, the unarchiving command set in the Unarchive Command dialog box is displayed here. You can edit the command (see Help on the Unarchive Command dialog box).
- Shutdown observers before unarchiving
 Activate this checkbox if desired.
- Into Directory The directory where you want the file system part of the repository to be stored.
- Buttons :
- OK Press this button to accept the settings in the dialog box and to leave it.
- Cancel Press this button to leave the dialog box without changes.

Unarchive Model dialog box

In: Corporate Management Tool

Source The pathname of the archive file. Type in the source name or use the Browse button to insert it. This must be the same name as when you backed up

Command By default, the unarchiving command set in the Unarchive Command dialog box is displayed here. You can edit the command (see Help on the Unarchive Command dialog box).

Shutdown observers before unarchiving
 Activate this checkbox if desired.

Into Directory The directory where you want the file system part of the model to be stored.
Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Restore Entire Corporate (select restore command) dialog box

In: Corporate Management Tool

From Corporate directory (created by unarchive command)

Type in the Corporate name or use the Browse button to insert it

New Corporate name (defaults to directory name)

Fill in the Corporate name

Shutdown observers before restoring

Activate this checkbox if desired.

Database

Database to create/overwrite

You can choose to create a new database or overwrite an existing database. If you want to overwrite an existing database, the Overwrite Existing Database option must be selected.

Database Name (All repositories, except Oracle)

The name of the repository database.

Database Server (Informix repositories only)

The name of the computer on which the database software is running.

Database Connect String

(Oracle repositories only)

The name of the connected database.

Database User (Oracle and SSA repositories only)

For Oracle, the schema id of the Oracle user.

For SSA, the name of the SSA user.

Database Password

(Oracle and SSA repositories only)

Select the Enter button to open a dialog that allows you to enter the database password.

Database Directory

(SSA repositories only)

The name of the directory that holds the database.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Restore Single Project dialog box (select restore command)

In: Project Management Tool

From Project directory (created by unarchive command)

Type in the Project name or use the Browse button to insert it

New Project name (taken from dump if not specified)

Fill in the Project name

Overwrite Project (not overwritten if not specified)

Fill in the Project name

Shutdown observers before restoring

Activate this checkbox if desired.

Database

Database to create/overwrite

You can choose to create a new database or overwrite an existing database. If you want to overwrite an existing database, the Overwrite Existing Database option must be selected.

Database Name (All repositories, except Oracle)

The name of the repository database.

Database Server (Informix repositories only)

The name of the computer on which the database software is running.

Database Connect String

(Oracle repositories only)

The name of the connected database.

Database User (Oracle and SSA repositories only)

For Oracle, the schema id of the Oracle user.

For SSA, the name of the SSA user.

Database Password

(Oracle and SSA repositories only)

Select the Enter button to open a dialog that allows you to enter the database password.

Database Directory

(SSA repositories only)

The name of the directory that holds the database.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Restore Single Model dialog box (select restore command)

In: Project Management Tool

From Model directory (created by unarchive command)

Type in the Project name or use the Browse button to insert it

New Model name (taken from dump if not specified)

Fill in the Model name

Overwrite Model (not overwritten if not specified)

Fill in the Model name

Shutdown observers before restoring

Activate this checkbox if desired.

Database

Database to create/overwrite

You can choose to create a new database or overwrite an existing database. If you want to overwrite an existing database, the Overwrite Existing Database option must be selected.

Database Name (All repositories, except Oracle)

The name of the repository database.

Database Server (Informix repositories only)

The name of the computer on which the database software is running.

Database Connect String

(Oracle repositories only)

The name of the connected database.

Database User (Oracle and SSA repositories only)

For Oracle, the schema id of the Oracle user.

For SSA, the name of the SSA user.

Database Password

(Oracle and SSA repositories only)

Select the Enter button to open a dialog that allows you to enter the database password.

Database Directory

(SSA repositories only)

The name of the directory that holds the database.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Project Browser dialog box

In: Corporate Management Tool)

Projects in corporate <corporate_name>

Shows the available projects in the selected repository and their Operable status.

Subdirectory Shows the corresponding name of the subdirectory in the repository.

Operable With this button you can set the operability of the project.

Model Browser dialog box

(In: Corporate Management Tool)

Models in corporate <corporate_name>

Shows the available projects in the selected repository and their Operable status.

Subdirectory Shows the corresponding name of the subdirectory in the repository.

Operable With this button you can set the operability of the project.

Delete Non-hanging Lock dialog box

In: Lock Management Tool) This is a warning dialog box in which you have to confirm whether you want to delete the lock you selected.

Delete Client Node dialog box

In: Client/Server Configuration Tool This box warns you whether you are sure to delete the registration of the selected client.

Archive Entire Corporate dialog box

In: [Corporate Management Tool](#)

Corporate name Type in the Corporate name or use the Browse button to insert it.

Shutdown observers before archiving
Activate this checkbox if desired.

Destination Type in the Destination directory where you want to save your repository or use the Browse button to select one.

Command By default, the archiving command set in the [Archive Command dialog box](#) is displayed here. You can edit the command (see Help on the [Archive Command](#)).
Note .

- When you want to backup a corporate directory, you must run the command from the parent directory of the corporate directory.
- When you want to backup a project or model directory, you must run the command from the corporate directory.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Archive Single Model dialog box

In: Corporate Management Tool

Single Model directory name

Type in the Model directory name or use the Browse button to insert it.

Shutdown observers before archiving

Activate this checkbox if desired.

Destination

Type in the Destination directory or use the Browse button to insert it.

Command

By default, the archiving command set in the Archive Command dialog box is displayed here. You can edit the command (see Help on the Archive Command).
Note .

- When you want to backup a corporate directory, you must run the command from the parent directory of the corporate directory.
- When you want to backup a project or model directory, you must run the command from the corporate directory.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Archive Single Project dialog box

In: Corporate Management Tool

Project directory name

Type in the Project directory name or use the Browse button to insert it.

Shutdown observers before archiving

Activate this checkbox if desired.

Destination

Type in the Destination directory or use the Browse button to insert it.

Command

By default, the archiving command set in the Archive Command dialog box is displayed here. You can edit the command (see Help on the Archive Command).

Note .

- When you want to backup a corporate directory, you must run the command from the parent directory of the corporate directory.
- When you want to backup a project or model directory, you must run the command from the corporate directory.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Dump Entire Corporate dialog box

In: Corporate Management Tool

Corporate name Fill in the name of the repository to be dumped.

Shutdown observers before dumping

Activate this checkbox if desired.

Database

Name (All repositories, except Oracle)

The name of the repository database.

Server (Informix repositories only)

The name of the computer on which the database software is running..

Database Connect String

(Oracle repositories only)

The name of the connected database.

Database User (Oracle and SSA repositories only)

For Oracle, the schema id of the Oracle user.

For SSA, the name of the SSA user.

Database Password

(Oracle and SSA repositories only)

Select the Enter button to open a dialog that allows you to enter the database password.

Database Directory

(SSA repositories only)

The name of the directory that holds the database.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Dump Model dialog box

In: Corporate Management Tool

Model name Fill in the Model name.

Shutdown observers before dumping
Activate this checkbox if desired.

Database

Name (All repositories, except Oracle)
The name of the model database.

Server (Informix repositories only)
The name of the computer on which the database software is running..

Database Connect String
(Oracle repositories only)
The name of the connected database.

Database User (Oracle and SSA repositories only)
For Oracle, the schema id of the Oracle user.
For SSA, the name of the SSA user.

Database Password
(Oracle and SSA repositories only)
Select the Enter button to open a dialog that allows you to enter the database password.

Database Directory
(SSA repositories only)
The name of the directory that holds the database.

Buttons :

OK Press this button to accept the settings in the dialog box and to leave it.

Cancel Press this button to leave the dialog box without changes.

Unarchive Single Project dialog box

In: Corporate Management Tool

- Source Type in the source or use the Browse button to insert it.
- Command By default, the unarchiving command set in the Unarchive Command dialog box is displayed here. You can edit the command (see Help on the Unarchive Command dialog box).
- Shutdown observers before unarchiving
 Activate this checkbox if desired.
- Into Directory The directory where you want the file system part of the project to be stored.
- Buttons :
- OK Press this button to accept the settings in the dialog box and to leave it.
- Cancel Press this button to leave the dialog box without changes.

Archive Corporate

In: [Corporate Management Tool](#)

Use **Options | Archive...** in the Corporate Management Repository tool to specify the Archive Command. The [Archive Command dialog box](#) appears.

This command allows changing of the M4 variable `M4_archive_cmd` which is used by Backup (see [File | Backup](#)) to archive a repository, project or model directory.

Backup

In: Corporate Management Tool

Use **File | Backup...** to select a backup method for an entire repository or a project file. Select

- Entire Repository..., or
 - Single Project...
- from the cascading menus.

Restore

In: Corporate Management Tool

Use **File | Restore...** to select a restore method for an entire repository or a project file. Select

- Entire Repository..., or
 - Single Project...
- from the cascading menus.

Unarchive Command

In: [Corporate Management Tool](#)

Use **Options | Unarchive...** in the Corporate Management Repository tool to specify the Unarchive Command. The [Unarchive Command dialog box](#) appears.

This command allows changing of the M4 variable `M4_unarchive_cmd` which is used by Restore (see [File | Restore](#)) to unarchive a repository, project or model directory.

Backup Entire Repository

In: Corporate Management Tool

Use **File | Backup... | Entire Repository...** to select a backup method for an entire repository. **The Backup Entire Corporate dialog box appears.**

Backup Single Project

In: Corporate Management Tool

Use **File | Backup... | Single Project...** to select a backup method for an entire repository. **The Backup Single Project dialog box appears.**

Restore Entire Repository

In: Corporate Management Tool

Use **File | Restore... | Entire Repository...** to select a Restore method for an entire repository. **The Restore Entire Corporate dialog box appears.**

Restore Single Project

In: Corporate Management Tool

Use **File | Restore... | Single Project...** to select a Restore method for an entire repository. **The Restore Single Project dialog box appears.**

Change Model Reservation dialog box

In: Corporate Management Tool

You can change the status of a model from reserved to unreserved.

If the currently selected model is not reserved, use this dialog box to set the Exclusive Access option for the model and enter reservation comments.

If the currently selected model is reserved, use this dialog box to release the Exclusive Access option for the model, making it available for other users.

