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]                  |
|-----------------------------------------------|
|                    [Tool Bar]                 |
|-----------------------------------------------|
|                 [Context Area]                |
|                                               |
|-----------------------+-----------------------|
|                       |                       |
:                       :                       :
:  [Navigation Area]    :  [Information Area]   :
:                       :                       :
|                       |                       |
|-----------------------+-----------------------|
|                [Message Area]          /\  \/ |
+===============================================+
```

- **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
Corporate	Corporate level
Project	Project level
ConfigVersion	Configuration level
PhaseVersion	Phase level
SystemVersion	System level
SystemVersion	System level (implementation)
DocumentVersion	Document level

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

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

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

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

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

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)

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.)

Edit Group Structure dialog box

Use this dialog box to specify the files and the items you want to add to the current group.

This dialog box contains the following sections:

[mode]
[New...] [Show...] [Delete...]
[Contents]

---

mode

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

New ...

explicit group version mode	In this mode, New allows you to select one or more group versions. The contents of the selected group versions are added to the current group version.
group version filter mode	In this mode, New allows you to specify a group version filter, which filters on group names or property values. The contents of the group versions matching the filter are added to the current group version. Group version filters are created with the New Subgroup Filter dialog box.
explicit file mode	In this mode, New allows you to select one or more specific file(s) to add to the group version.
file filter mode	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.
file selector mode	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.
item filter mode	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
item selector mode	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.

New File Selector dialog box

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 CAD 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 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.

New <object> Filter dialog box

Where <object> can be:

- Subgroup
- File
- Item

Use this dialog box to create a filter that selects group versions, 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 (not available for subgroups)
- 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 matching.

Move Definition dialog box

menu entry: **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 this dialog box you specify whether or not you want item property values to be overwritten during Import 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.

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
  - **Misc** : most of the properties can be found on this page

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

Depending on the selected object, you can set or view the values of the following default properties:

```
[File System Path Part]
[Free Text]
[Document properties] (DocumentVersions only)
[Structure Generator] (Local sections only)
[Contents Generator]  (Local sections only)
```

```
-------------------------------------------------
```

File System Path Part

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
==========================================
Corporate           <Corporate_name>
Project             <Project_name>
ConfigVersion       <ConfigVersion_name>_<ConfigVersion_number>
PhaseVersion        <PhaseVersion_name>
SystemVersion       <SystemVersion_name>
FileVersion         <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.

```
-------------------------------------------------
```

Free Text

Use this property to annotate a browser object. Free text is not bound to any syntax or constraints.

```
-------------------------------------------------
```

Document Properties

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

Structure Generator

(local sections)

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.

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

Contents Generator

(local sections)

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**, **Il** 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 Generators, refer to the *ObjectTeam Document Generation Guide*.

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 actions that can be allowed or prohibited for a specific object are also displayed in the Navigation Area of the Browser underneath that object. For example, the actions that can be carried out on a Project can be viewed by unfolding its Access Rights pseudo-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.

Dialog box

Objects	Select the object 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. These you can set to:

- **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 *Undefined*, no access rule is entered in the database. As a result, no ownRights are visible for that object in the browser.

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.

Changing the location of an external link

(In: Browser)

Use **File > Change > Location...** to change the file an **External Link** refers to. 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.

You select the new file in a File selection dialog box.

Changing the link status of a browser object

(In: Browser)

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.

Besides the browser objects mentioned above, there is another object with a link status:

- 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.

––

Dialog Box

(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.

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 Association Diagrams in the current system. It checks all classes and events defined in these diagrams. Events can be specified in:

- Event Trace Diagram: ETD event
- Class Communication Diagram: Communication Message
- 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:

- **CAD selected:** Only the events received by classes from the current Class Association Diagram are checked. The result is the same as when Check Local Model is invoked from the Class Association Diagram Editor.
- **CCD selected:** Only the classes corresponding to the Communication Messages of the selected Class Communication Diagram(s) are checked.
- **ETD selected:** Only the classes corresponding to the ETD events of the selected Event Trace 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.

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.

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 specify what you want to compare in the Compare With Previous Phase dialog box.

____

Dialog box

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.

Setting the 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 default compare command is **fcomp**, a compare utility that can be found in the `<M4_home>/bin` directory.

The compare command specified here is only used to compare *text* files. For *diagram* file comparison, ObjectTeam uses the tool **udm_cmp**.

You specify the **compare command** in the Compare Command dialog box.

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.

____

Dialog box

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.

Deactivate Role... - menu entry

(In: Browser)

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

Editing browser objects

(In: Browser)

[Project Level] [Configuration Level] [Phase Level] [System Level] [Implementation System Level] [Document Level]

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.

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

Edit - menu entry (system level)

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.

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

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 an 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.

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

Edit - menu entry (document level)

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.

- *< local section type>*
  The appropriate word processor is started in a separate window.
- *Customization file*
  Depending on the type of customization file currently selected, the customization editor or the default ASCII text editor is started.

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

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 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.

Determining access

An access rule determines a role's access to a particular action. 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

Filtering the Information Area

(In: Browser)

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

Dialog box

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

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).

Importing data from the previous phase

(In: Browser)

Use **Utilities > Import From Previous Phase** to copy systems and file versions from a previous phase into the current one.

Note : If you import data from the Object Design phase to the Implementation phase (on phase or on system level), you start the code generation process.

The current browser level and menu option determine what is imported:

- phase level:

New Systems	With this option, you import all systems that are not present yet in the current phase.
Specific Systems...	With this option, you can select the systems you want to (re-)import into the current phase.

- system level:

New	With this option, you import all file versions that are not present yet in the current phase.
Specific Systems...	With this option, you (re-)import the file versions that are currently selected in the information area.

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 dialog box.

Dialog Box

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 *frozen*, 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 *frozen*, 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:

- *working*
- *frozen*

The status of File Sections can be:

- *fixed*
- *selected*

Text Some explanatory text on the selected object.

Type Within the environment of the browser, objects are recognized by their *Type*.

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.

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, open the psuedo object <defined items>, which appears under the browser object system. All items in that system appear in the Information Area. After selecting the items that you are interested in, select **Utilities > Move Definition**. When the Move Definition dialog box appears, select the system that you want to move the definitions to.

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
Corporate level	[Project] [User] [Role] [UserRoleLink]
Project level	[Configuration] [UserRoleLink]
Configuration level	[Phase]
Phase level	[SystemVersion] [DocumentVersion]
System level	[FileVersion] [GroupVersion] [ExternalLink]
Document level	[DocumentStructureMatrix]
All levels	[CustomizationFile]

Below, the appropriate conditions for every menu entry from the **File > New** menu are listed:

File > New >Configuration Version...

Conditions:

- The browser must be on project level.

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

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.

File > New >Document Structure Matrix

Conditions:

- The browser must be on document level.

- Only one Document Structure Matrix per document level can exist.

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.

File > 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.

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.

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.

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.

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.

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.

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

File > New >Project...

Conditions:

- The browser must be on corporate level.

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

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.

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.

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.

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.

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**.

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.

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

Dialog box

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

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
: ...         ...    ...       ...
```

Reverse Engineering...

(In: Browser)

Use **Utilities > Reverse Engineer...** to create classes from header files. This allows you to do the following:

- Make class libraries available inside the ObjectTeam environment.
- Understand structure and functionality of class libraries.

Note : Reverse engineering is not designed to capture all the information in a header file. Therefore, do not use reverse engineering to import classes in order to maintain them in ObjectTeam.

The selected header files are converted to a Class Association Diagram with CDMs using the parameters specified in the Reverse Engineer Conversion Dialog Box. (Reverse engineering ignores all Preprocessor symbols, such as #ifdef and #define, in the header files. However, in the Reverse Engineer dialog box, you can specify a preprocessor to be run before reverse engineering.)

Reverse engineering is currently supported for the following target languages:

- C ++
- Informix NewEra
- SmallTalk

Delete Unreferenced Items - menu entry

(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.

Displaying Your Access Rights

(In: Browser)

Use **Security > Show Access Rights** to examine the effective access rights. These 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.

____

Dialog box

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.

Version Management

(In: Browser)

[Understanding Versions]

[Activate]	[Make Fixed]
[Compare...]	[Make Selected...]
[Copy...]	[Make Snapshot...]
[Deactivate]	[New]
[Delete...]	[Restore...]
[Deselect...]	[Select...]
[Freeze...]	[Select Fixed...]
[Make Corporate...]	[Snapshot...]
[Make Current]	

The Version menu contains a number of menu entries that you can use for version management on certain browser objects. Below, every menu entry of the Version menu is discussed briefly, including information about the browser level they can be used in and the browser *objects* they can be used for.

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.

Understanding Versions

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.

Version Numbering

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 > Activate

Available on:

- System level

You can use this menu entry to activate a Configuration version that has been deactivated, or to add corporate groups to the current system.

Activating Configuration versions: Activating a Configuration version changes its status from *backGround* to *frozen*, and moves it from the file system into the repository. Use Version > Deactivate to deactivate a Configuration version.

Add corporate groups: You can also use this menu entry to add objects from corporate groups to the current system. 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.

The added objects have status reused and the in corporate flag set to yes. Objects from corporate groups cannot be versioned. Also, they cannot be deleted individual, only the corporate group as a whole can be deactivated.

Version > Deactivate

Available on:

- System level

You can use this menu entry to deactivate a Configuration version, or to remove corporate groups from the current system.

Deactivating Configuration versions: Deactivating a Configuration version changes its status from *frozen* to *backGround*, and moves it out of the repository into the file system. Use Version > Activate to activate a deactivated Configuration version.

Remove corporate groups: You can also use this menu entry to remove corporate groups from the current system.

Version > Freeze

Available on:

- All browser levels (except Corporate)

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.

CAD : If you freeze a Class Association 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.

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

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 > Unfreeze

Available on:

- All browser levels (except Corporate)

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.

––

Version > New

Available on:

- All browser levels (except Corporate)

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 selected object.

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.

––

Version > Copy

Available on:

- All browser levels (except Corporate)

Use this menu entry 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.

––

Version > Delete

Available on:

- All browser levels (except Corporate)

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.)

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 > Select

Available on:

- Configuration level
- Phase level
- System level
- Implementation System level
- Document Level

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. If there is more than one version of the current Configuration, you can even select versions of objects from another Configuration version.

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

You select versions of objects in the Select Version dialog box.

––

Version > Deselect

Available on:

- Configuration level
- Phase level
- System level
- Implementation System level
- Document Level

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.

A deselected object is removed from the information area and therefore becomes invisible in the browser. However, it is not deleted, since you can put it back in the foreground again by using Version > Select.

Version > Compare

Available on:

- System level
- Implementation System level
- Document Level

You can compare a *selected* version of a diagram or text file with another version of the same object, using this menu entry. The result of the comparison is displayed in a Monitoring Window.

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.

You select the version you want to compare the selected version with, in the Compare Version dialog box.

Version > Make Fixed

Available on:

- Document Level

You can use this menu entry to change the status of a File Section or File Property Section from *selected* into *fixed*.

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*.

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

Version > Select Fixed...

Available on:

- Document Level

You can use this menu entry to change the status of a File Section or File Property Section from *selected* into *fixed*.

As opposed to Version > Make Fixed, you have a choice with Version > Select Fixed... which version you want to fix the reference to. You can make your choice in the Select Version dialog box.

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 > Make Current

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 Snapshot

Available on:

- Document Level

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

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 > Make Selected

Available on:

- Document Level

You can use this menu entry to change the status of a File Section or File Property Section from *fixed* into *selected*.

A File Section or File Property Section with status *selected* 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.

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

Version > Make Corporate

Available on:

- System level
- Implementation System level

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

When you use Version > Corporate for a SavedGroupVersion, 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.)

Dialog Box

The Make Corporate Dialog Box allows you to define a name for your corporate group. Default is the name of the Saved Group Version. A comment field is also available.

Version > Restore

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 > Snapshot...

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.

Dialog Box

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.

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: • CAD • CCD
Process	This view is only available on system level. In this view, only the objects types that are process-oriented are displayed. These are: • CCD • DFD • ETD • MGD • 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. You can specify different viewers for different types of files.

____

Dialog box

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

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 ...
  - Effective Roles...
  - 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
    - 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
  - Activate Role...
  - Deactivate Role...
  - Show Access Rights...
  - Role Rights...
    - Edit ...
    - Show ...
- **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 ...
  - Effective Roles...
  - 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
        - Text ...
        - Graphical ...
    - Viewer ...
    - Copy User Environment
        - To Corporate Environment...
- **Version** menu
    - Freeze ...
    - Unfreeze
    - New
    - Copy ...
    - Delete ...
    - Select ...
        - New ...
        - Selected ...
    - Deselect
    - Compare ...
- **Security** menu
    - Activate Role...
    - Deactivate Role...
    - Show Access Rights...
    - Role Rights
        - Edit ...
        - 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 ...
  - Update Document Directory
  - Change
    - Link Status...
  - Properties
    - Edit ...
    - Delete ...
    - Show ...
  - Effective Roles...
  - 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
  - Activate Role...
  - Deactivate Role...
  - Show Access Rights...
  - Role Rights...
    - Edit
    - Show
- **Utilities** menu
  - Clone
  - Class Browser
  - Monitoring Window...
  - Execution Window...
  - Move Definition...
  - Delete Unreferenced Items
  - Import 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 ...
  - Effective Roles...
  - 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
    - 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
    - Activate Role...
    - Deactivate Role...
    - Show Access Rights...
    - Role Rights...
        - Edit ...
        - Show ...
- **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 ...
  - Effective Roles...
  - 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
  - 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
    - Activate Role...
    - Deactivate Role...
    - Show Access Rights...
    - Role Rights...
        - Edit
        - Show
- **Utilities** menu
    - Clone
    - Class Browser
    - Monitoring Window...
    - Execution Window...
    - Move Definition...
    - Delete Unreferenced Items

    - Import From Previous Phase
        - New ...
        - Selected ...
    - 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

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

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 in one of three ways:

- 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. To view the childRight role rights, select *childRight* in the *Role Right* field.
- Open the appropriate ControlledList. The browser will show all defined access rules for the object: ownRight and childRight. (*Note:* If all access rules for a particular object are set to *Undefined* for all Roles, when you unfold the object in the navigation area, no access rules appear.)
- Select the appropriate object of type *childRight* in the browser and select File > Info. Read the information in the *Text* field.

Controlled Class - browser object

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

You cannot add or delete controlled classes. To view the controlled classes defined by ObjectTeam, open the pseudo object <controlled classes> on the appropriate level: either Corporate or Project. To specify the role rights for a controlled class, select the class and then select Security > Access Rules > Edit.

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

```
{Corporate| Project}
 |
 +- ...
 |
 +- ControlledClasses
     |
     +- ControlledClass
        |
        +- ownRight
```

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.

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

```
 ...
 |
 +- ControlledLists
    |
    +- ControlledList
       |
       +- ownRight
       +- childRight
```

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    |
----------------------------------------
```

File Version - browser object

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:

- cad
- cdm
- ccd
- dfd
- etd
- mgd
- 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
```

Group - browser object

[Group] [Saved Group] [Corporate Group]

A *Group* is a collection of diagram or text files and items. A *GroupVersion* defines a group, a *SavedGroup* is a snapshot of the contents of a group, and a *Corporate Group* 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 using File > New > Group Version. 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.

SavedGroupVersion

A Saved Group is a snapshot of a group. Saved Groups are listed in the pseudo object <saved groups>:

```
...
|
+- System
   |
   +- ...
   |
   +- <saved groups>
      |
      +- SavedGroupVersion
```

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>:

```
CORPORATE
|
+- ...
|
+- <corporate groups>
   |
   +- CorporateGroup
```

ownRight - browser object

A browser object *ownRight* specifies the role rights for browser objects of the following types:

- **ControlledClass**
- **ControlledList**
- **< controlled objects>**

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, there are three ways of doing it:

- 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.)
- Unfold the object in the navigation area of the Browser. The browser will show all defined access rules for the object. (*Note:* If all access rules for a particular object are set to *Undefined* for all Roles, when you unfold the object in the navigation area, no access rules appear.)
- Select the appropriate object of type *ownRight* in the browser and select File > Info. Read the information in the *Text* field.

PhaseVersion - browser object (PhaseVersion)

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

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:

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

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 File > New > User. 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 corporate level and on project level. A user creating a repository automatically becomes the 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 access rights 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

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 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.

Importing a System From the previous phase

1. Make sure the browser is on the appropriate phase level.
2. Select Utilities > Import From Previous Phase.
3. Select **New systems...** to import all systems in the previous phase that do not yet exist in the current phase. Or, select **Specific Systems** to explicitly specify the system(s) you want to import.

If your access rights allow it, the specified system(s) are imported into the current phase.

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 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

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):

Project	Phase Version
Configuration	System Version
View	Filter

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
ControlledClasses	ControlledClass
ControlledLists	ControlledList
ControlledList	ownRight
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
ControlledClasses	ControlledClass

ControlledLists	ControlledList
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	**PhaseVersion**
Customization files	Customization file
ControlledLists	ControlledList

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	**SystemVersion** **DocumentVersion**
Customization files	Customization file
ControlledLists	ControlledList

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	**FileVersion** **GroupVersion**
Customization files	Customization file
Saved Groups	SavedGroupVersion
Defined Items	Defined Item
SavedGroupVersion	FileVersion
Controlled Lists	ControlledList
ControlledLists	ControlledList

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	**dsm**
	local section
	file section
	property section
Customization files	Customization file
ControlledLists	ControlledList

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

Navigation Area

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 ETD 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*.

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 Association Diagram

(In: Class Browser)

Use **Utilities > Edit CAD** to start the Class Association Diagram Editor for the currently open class. If the opened class appears in several CADs, 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.

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**.
OK button	Use this button to open only the first class matching the specified search pattern. The dialog box then disappears.
Next button	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.

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.

____

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.
OK button	Use this button to open only the first class matching the specified search pattern. The dialog box then disappears.
Next button	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.

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.

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 specified in the list box.
List box	This list box shows the current settings for the filter elements belonging to the class feature selected in the field *Select Filter*. To change the settings of a filter element, select the element and then select the *Set Filter Element* button.
Set Filter Element	Press this button to change the settings of the filter element selected in the *List box*.

Enabled	Switch this button on to enable the filter setting for this 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. Features that match the pattern specified, will appear in the Class Browser.

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 closed.
Reset	Use this button to cancel the changes made to the Filter Feature settings.
Cancel	Use this button to cancel the operation.

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 likes 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 Association 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.

Dialog box

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.

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

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 CAD
- **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 CAD:

- 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 - 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):

Project	SystemVersion
Configuration Version	Class
PhaseVersion	

Display Area - 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 CAD.

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 ...
```

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:

Group	The name of the group
Check value	The current message type of the group. You can modify this with the Check Value dialog box.
Description	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.

____

Check Value dialog box

The Check Value 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

Editing Menu Object Properties - dialog box

(In: Customization Editor)

Use **Edit > Edit Properties** in the Menu Customization Editor to specify a menu button further.
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** contains the following property pages:

- Storage Page
- Interface Page
- Command Page
- Enable /disable Page
- Arbiters Page (MenuBarButton and CascadeButton only)

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.

Interface page

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 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.

Command page

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.

Enable /disable page

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 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

Arbiters page

Allows you to specify up to three arbiters for the button. The Arbiter page is available for:

- MenuBar buttons
- Cascading buttons

Editing Objecttype Properties - dialog box

(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** 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.

Editing Navigation Strategy Definition Properties - dialog box

(In: Customization Editor)

Use **Edit > Edit Properties** to specify properties for a navigation strategy.
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.

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
- ccd
- dfd
- etd
- mgd
- std
- ucd

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
- ccd
- dfd
- etd
- mgd
- std
- ucd

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.

Editing Navigation Strategy Location Properties - dialog box

(In: Customization Editor)

Use **Edit > Edit Properties** to specify properties for a navigation strategy location.
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** contains only one property page with the following fields:

Strategy	Uniquely identifies the open strategy
Item Type	Select an item here.
Context	Select a phase here.
Diagram Type	Select a diagram type here
Component Type	Indicates the type of components for which a property can be manipulated.
Label Type	Indicates the type of labels within a diagram for which the property can be manipulated. Examples of label types are: `cad_class`, `data_store`, `con_message`.
Condition	The values *KEY* and *NO_KEY* can be specified for properties of key and non-key attributes of a class. *TOP*, *MIDDLE* and *LEAF* 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.

Editing Property Definition Properties - dialog box

(In: Customization Editor)

Use **Edit > Edit Properties** to specify properties for a property definition.
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** 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.

Editing Property Location Properties - dialog box

(In: Customization Editor)

Use **Edit > Edit Properties** to specify properties for a property location.
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** contains only one property page with the following fields:

Container Kind	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 *Clear* 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.
Container Type	The type of object for which the property can be manipulated. Examples of container types for the Container Type *Item* are: `cl` and `de`. Examples for the Container Type *Component* are: `association`, `parameter` and `qualif_aggr`.
Phase Type	Indicates the Phase in which the property is available.
Diagram Type	Indicates the file version in which the property is available.
Component Type	Indicates the type of components for which a property can be manipulated. Specifying this is only useful if the current Container Kind is *Item*.
Label Type	Indicates the type of labels within a diagram for which the property can be manipulated. Examples of label types are: `cad_class`, `data_store`, `con_message`.
Condition	The values *KEY* and *NO_KEY* can be specified for properties of key and non-key attributes of a class. *TOP*, *MIDDLE* and *LEAF* 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.

Editing View Properties - dialog box

(In: Customization Editor)

Use Edit > Edit Properties in the View Customization Editor to edit the properties of (new) views.
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** contains the following property pages:

- Storage Page
- Interface Page
- Properties Page
- Sort Command Page

____

Storage page

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.

Interface page

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 checkbutton switched on in the Menu Customization Editor Interface Page will appear always in the toolbar.

Properties page

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

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.

New Customization File - dialog box

Customization File:

You can select the desired 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 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 User Customization File - dialog box

External File:	You can select the desired 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.

New Button/Separator - dialog box

(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.

Dialog Box

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: 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: Customization Editor)

This dialog box appears when you select **Edit > New...** from the menu bar.

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: 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: 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`, `DropDwnComboBox`

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: 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: 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 Association 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.

New ...	Use this button to add a symbol to the current selection, which is displayed under *Contents:*. You can only select symbols from a fixed list.
Test ...	Use this button to see how the symbols listed under *Contents:* will be arranged in the control panel. The result depends on the number of columns selected.
Delete	Use this button to delete a symbol from the current selection of symbols, listed under *Contents:*
Contents :	This box shows the current selection of symbols.
columns	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 *Test...* 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 File, browser object

Customization files can be defined for every browser level below corporate level. They can be edited with the Customization Editor, the Control Panel Dialog Box or a text editor.
You can define the following types of customization files:

```
[User Interface Files]      [Miscellaneous]
[Document Generation Files] [User Customization Files]
[Code Generation Files]
```

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

```
 {Corporate, Project, Configuration, Phase, System}
 |
 +- Customization files
    |
    +- Customization file
```

Customization Files: User Interface

Menu Customization Editor

- **desk.mnu** - Browser menus
- **class .mnu** - Class Browser menus
- **diagram .mnu** - Generic diagram menus
- **cad .mnu** - CAD menus
- **ccd .mnu** - CCD menus
- **dfd .mnu** - DFD menus
- **etd .mnu** - ETD menus
- **mgd .mnu** - MGD 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** - CAD control panel
- **ccd .pnl** - CCD control panel
- **dfd .pnl** - DFD control panel
- **etd .pnl** - ETD control panel
- **mgd .pnl** - MGD 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

----

Customization Files: Document Generation

(Only at Corporate level)

- **fm40_imp .str** -
  Document Structure file for generating a default ObjectTeam document in FrameMaker 4.0 (for Implementation phase).
- **fm40_mod .str** -
  Document Structure file for generating a default ObjectTeam document in FrameMaker 4.0 (for all phases except Implementation phase).
- **fm40book .con** -
  Initial contents file for a FrameMaker 4.0 book file
- **fm40doc .mif** -
  ASCII file containing Maker Interchange Format (MIF) tags and Tcl macros. This file is used to generate FrameMaker 4.0 document files: local sections of type Doc
- **fm40mif .con** -
  ASCII file containing Maker Interchange Format (MIF) tags. This file is used to generate MIF files: local sections of type Mif
- **fm40title .mif** -
  ASCII file containing Maker Interchange Format (MIF) tags and Tcl macros. This file is used to generate default FrameMaker document files containing a Title Page
- **fm40toc .con** -
  Empty FrameMaker Table of Contents (TOC) file, used to generate TOC files
- **il60_imp .str** -
  Document Structure file for generating a default ObjectTeam document in Interleaf 6.0 (for Implementation phase).
- **il60_mod .str** -
  Document Structure file for generating a default ObjectTeam document in Interleaf 6.0 (for all phases except Implementation phase).
- **il60catalog .asc** -
  ASCII file containing LISP tags and Tcl macros. This file is used to generate Interleaf catalogs.
- **il60class .asc** -
  ASCII file containing Interleaf structuring elements
- **il60doc .asc** -
  ASCII file containing LISP tags. This file is used to generate Interleaf documents
- **il60link .asc** -
  ASCII file containing LISP tags and Tcl macros. This file is used to generate Interleaf links to diagram files
- **il60title .asc** -
  ASCII file containing LISP tags and Tcl macros. This file is used to generate Interleaf title pages.
- **word_imp .str** -
  Document Structure file for generating a default ObjectTeam document in Microsoft Word for Windows 7.0 (for Implementation phase).
- **word_mod .str** -
  Document Structure file for generating a default ObjectTeam document in Microsoft Word for Windows 7.0 (for all phases except Implementation phase).
- **word_mod .str** -

- **Word.dot** - MS-Word Document Template (DOT) file, used to generate Microsoft Word for Windows (7.0) documents.

Customization Files: Code Generation

- **lang_types** - Maps standard types to OO language types
- **create .hdr** - Template used for generating SQL
- **create_procs .hdr** - Template used for generating SQL
- **create_tables .hdr** - Template used for generating SQL
- **db_types** - Maps standard types to RDBMS dependent database types
- **drop .hdr** - Template used for generating SQL
- **drop_procs .hdr** - Template used for generating SQL
- **drop_tables .hdr** - Template used for generating SQL
- **maketmpl** - Generic makefile template
- **sqlrules** - SQL Rules File (all levels)
- **stand_types** - Configuration file for standard datatypes

----

Customization Files: Miscellaneous

(only on Corporate level, unless otherwise indicated)

- **checkconfig** - checking message report
- **checkmap** - Check Mapping file
- **charset .map** - Character mapping file (8 bit -> 7 bit)
- **phases** - Phase definitions
- **prolog .ps** - PostScript prolog
- **ps** - directory containing PostScript fonts used by ObjectTeam
- **m4env** - M4 Environment File (all levels)
- **compare_rules** - Comparison performed by **udmcmp** and/or Compare Phase
- **Xdefaults** - Standard X Defaults (Unix only)
- **browserprocs** - Predefined Browser procedures (only at corporate level)
- **editorprocs** - Predefined diagram editor procedures (only at corporate level)

User 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

Various Customization Files

M4 Environment File

M4 Environment Files contain environment variables known as M4 variables that allow you to control the appearance and behavior of the ObjectTeam tools.

There are two M4 environment files: **m4env** and **Meta4UserEnv**. An m4env file always exists at Corporate level in the etc directory. This file can be redefined on any level in the repository by opening the <customization files> object on that level in the Browser, selecting File>New>Customization File Version, and selecting m4env from the list. The Meta4UserEnv file is stored in your home directory and applies only to you. This file overrides the settings in any m4env file. This file can only be made by opening the <user customization files> object in the Browser,selecting File>New>External File Version, and selecting Meta4UserEnv from the list.

SQL Rules File

The SQLRules File contains rules for insert-, delete- and update policies. Documentation is included.

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 Editor

The customization editor is a tool that makes it easier to edit the following customization files:

File name	Customization Editor
***.mnu**	Menu Customization Editor
objtype.objtype	Object Type Customization Editor
***.vie**	View Customization Editor
proplocs.proplocs	Property Availability Editor
propdefs.propdefs	Property Definition Editor
opendefs.opendefs	Open Strategy Definition Editor
openlocs.openlocs	Open Strategy Availability Editor

The name of the current customization file is displayed in the window section **Context Area**, behind the label *File (Version)*.

Customization Editors offer you dialog boxes in which you can enter information, select options and switch buttons on and off in order to customize parts of ObjectTeam.
If you want to edit a customization file of a different type than the one listed above, you can use the default text editor.

Window sections

The default customization editor window contains a number of sections.

```
+======================================+
|              [Menu Bar]              |
|--------------------------------------|
|              [Tool Bar]              |
|--------------------------------------|
|            [Context Area]            |
|                                      |
|--------------------------------------|
|                                      |
:                                      :
:            [Display Area]            :
:                                      :
|                                      |
|--------------------------------------|
|            [Message Area]            |
+======================================+
```

- **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 project, configuration, 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.

Menu Customization Editor

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

Object Type Customization Editor

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

View Customization Editor

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

Property Availability Editor

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

Property Definition Editor

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

Open Strategy Definition Editor

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

Open Strategy Availability Editor

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

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

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 version 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.

Dialog box

This dialog box appears after you select **File > Generate** n the browser on phase level

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.

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
  - Update Document Directory
  - Change
    - Link Status...
  - Properties
    - Edit ...
    - Delete ...
    - Show ...
  - Effective Roles...
  - 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
            - 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
        - Activate Role...
        - Deactivate Role...
        - Show Access Rights...
        - 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.

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.

Press the *OK* button to confirm the *Name* of the new object. Press the *Cancel* button to discard the operation. Press the *Edit* button to create and then edit the object.

Report Options dialog box

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.

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

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

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:

- Graphical printer settings (printer command, page width, page height)
- Text printer settings (printer command, line length, page length)

Dialog box

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
```

The default value is retrieved from the M4 variable **M4_ps_printer**.

Page Width	The default value in inches is retrieved from the M4 variable **M4_ps_page_w**.
Page Height	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
```

The default value is retrieved from the M4 variable **M4_a_printer**.

Line Length	The number of characters on a line. The default is retrieved from **M4_a_printer_llen**.
Page Length	The number of lines on a page. The default is retrieved from **M4_a_printer_plen**.

Creating 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:

- On Actors...
- On CDMs...
- On Classes...
- On Class Generalizations...
- On Communications...
- On Corporate Groups...
- On Documents...
- On Events...
- On Files...
- On Flows...
- On Groups...
- On Items And Properties...
- On Components And Properties...
- On Phases...
- On Projects...
- On Saved Groups...
- On Systems...
- On Unreferenced Cdms...
- On Use Cases...

Report output is sent to a Monitoring Window.

On Projects...

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

On Phases...

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

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

On Documents...

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

On Groups...

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

On Saved Groups...

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

On Corporate Groups...

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

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

On Communications...

Reports on communications is available on all browser levels. It reports communication messages between classes and other Class Communication Diagram components. Only the communication messages defined within the current level are reported. The communication messages are listed per selected project, configuration, phase version and system version.
The report includes:

- The name of the first component
- The type of the communication message
- The direction of the message `(<- or ->)`
- The name of the communication message
- The name of the second component
- The name of the Class Communication Diagram containing the communication message

----

On Unreferenced Cdms...

Reports on Class Definition Matrices whose corresponding class does not exist anymore.

The CDMs are listed per selected project, configuration, phase and system version.

On Actors...

Report on actors is available on two levels:

- **phase version**
- **system version**

The report includes, for each actor, the communication associations to or from the actor.

On Classes...

Report on classes is available on two levels:

- **phase version**
- **system version**

The report includes:

- Class names
- Class properties
- Attributes
- Operations

On CDMs...

Report on CDMs is available on two levels:

- **phase version**
- **system version**

The report includes:

- CDM name
- Diagrams in which the asssociated class appears
- System version that contains the CDM

On Class Generalizations...

Report on class generalizations is available on two levels:

- **phase version**
- **system version**

The report includes:

- Superclass name and type
- Generalization type
- Subclass name and type

On Events...

Report on events is available on two levels:

- **phase version**
- **system version**

The report includes:

- Event name
- Source and destination components of the event
- Diagram in which the event appears

On Flows...

Report on flows is available on two levels:

- **phase version**
- **system version**

The report includes:

- Flow name
- Depending on the type of flow, the source node, the destination node, or both

----

On Use Cases...

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.

On Components and Properties...

Report on components and properties is available only at the **system level**.

The report includes:

- Component name and type
- Property names and values

On Items and Properties...

Reports on items and properties is available on all browser levels.

The report includes:

- Item name and type
- Property names and values

(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.

____

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:

```
csh
```

Windows To start a DOS box for instance, enter the following command:

```
command.com
```

Menu Bar - 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)

You can invoke the Table of Contents of the Help Tool by:

- 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.

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 following tools have tool bars:

[Browser] [Diagram Editor] [Customization Editor] [Class Browser] [Help Tool]

You can change the buttons in the tool bar by using the Customization Editor.

Browser

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

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

Diagram Editor

The tool bar in the Diagram Editor contains the following buttons by default:

- File > Save
- Edit > Copy
- Edit > Delete
- Edit > ZoomOut
- Edit > ZoomIn

Customization Editor

The tool bar in the Customization Editor contains the following buttons by default:

- File > Save
- Edit > Redefine
- File > Info
- Edit > Delete

Class Browser

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

- File > Open
- File > Close

Help Tool

(UNIX-based platforms only)

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

- Navigate > Contents...
- Navigate > Index...
- Navigate > Previous Entry

Report Options dialog box

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:

- Event Trace Diagram
- Class Communication Diagram
- State Transition Diagram

Reverse Engineer - dialog box

Use this dialog box to modify the defaults for creating the diagrams. These are the options:

Diagram The name of the Class Association Diagram. If no name is entered no diagram is created.

Generate Select the generated types:

- CADs and CDMs
- Only CADs
- 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 CADs 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 header 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. 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.

Dialog Box

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.

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 generated during Import From Previous Phase (SQL). 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 a Database / (Schema)

(In: Browser on implementation system level)

Use **Database > Select Database** (for Oracle databases: **> Select Schema**) to select the database your application will work on. 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: **> Deselect Schema**) to deselect it.

Dialog Box

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.

Setting Password to access Database (Server)

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

Server	If it is the database server that requires a password, specify it in this field.
Schema	If it is the Oracle schema that requires a password, specify it in this field.
Database	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)

If you edit attributes or operations in the generated C++ header files, you can use **Utilities > Round Trip Selected** to move your changes back into the CDMs of the Object Design phase. This ensures that your changes are preserved when you regenerate the header files.

Note : Round-trip engineering is only available for C++ header files created for non-persistent classes.

Round -trip engineering can capture changes to the following:

- attributes
- operations

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 a Monitoring Window, 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.
4. Respond to each proposed action by entering y (yes) or n (no).
   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.

Changes You Can Make to Attributes

In a header file generated by ObjectTeam, the section that defines attributes (Private section only) is indicated by the comment: `User-defined attributes`. Within this section, you can make the following types of changes and round-trip engineering can move them back into the CDMs of the Object Design phase:

- Add attributes
- Delete attributes
- Reorder attributes
- Change the default value of an attribute
  *Note:*If you change a default value, be sure to change it in the comment as well as in the code; round-trip engineering reads the value from the comment.

*Note :*If you rename an attribute, round-trip engineering removes and adds the attribute; therefore, its properties are lost.

For more information, see the *ObjectTeam Code Generation Guide for C++*.

Changes You Can Make to Operations

In a header file generated by ObjectTeam, the sections that define operations (Public, Protected, and Private sections) are indicated by the following comment: `User-defined methods`. Within these sections, you can make the following types of changes and round-trip engineering can move them back into the CDMs of the Object Design phase:

- Add methods
- Delete methods
- Change method signatures by adding, removing, or reordering parameters, changing default values, or changing the return type.

Note : If you rename an operation or parameter, round-trip engineering removes and adds the object; therefore, its properties are lost.

Tip : If you have multiple methods with the same name, do not delete one and change the signature of another at the same time. Delete one, run round-trip engineering, change the signature of the other, and run round-trip engineering. This approach prevents potential errors in round-trip engineering.

Compile With Options Dialog Box

The Compile With Options Dialog Box allows you to enter options for the **make** utility. For details, refer to the man page for **make** (on Unix) or the compiler documentation (on Windows NT or Windows 95).

Dialog Box

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 association 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 assocition 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.

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.

____

Save Output dialog box

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.

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.

Control Panel - Class Association Diagram Editor

The following symbols are available in the control panel of the Class Association Diagram Editor (arranged according to their position in the control panel).

CAD Class	Link Attribute Box
CAD Container Class	N-ary Association
More Classes	Note
Vertex	Association
Aggregation	Qualified Association
Qualified Aggregation	N-ary Association Connector
Constraint	Link Attribute Loop
Generalization	Overlapping Generalization
Note Connector	Propagation
Select	
Begin Mult. One-Mandatory	End Mult. One-Mandatory
Begin Mult. One-Optional	End Mult. One-Optional
Begin Mult. Many	End Mult. Many

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

Control Panel - Class Communication Diagram Editor

The following symbols are available in the control panel of the Class Communication Diagram Editor (arranged according to their position in the control panel).

CCD Class	CCD Class Reference
CCD Container Class	CCD Actor
Subject	Note
Vertex	Communication Message
Note Connector	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

Control Panel - Data Flow Diagram Editor

The following symbols are available in the control panel of the Data Flow Diagram Editor (arranged according to their position in the control panel).

Data Process	Data Store
DFD Actor	DFD Anchor
Note	Vertex
Data Flow	Result Flow
Control Flow	Update Flow
Note connector	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

Control Panel - Event Trace Diagram Editor

The following symbols are available in the control panel of the Event Trace Diagram Editor. (Arranged according to their position in the control panel.)

ETD Initiator	ETD Object
Note	Vertex
ETD Event	Note connector
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

Control Panel - Message Generalization Diagram Editor

The following symbols are available in the control panel of the Message Generalization Diagram Editor. (Arranged according to their position in the control panel.)

Message Definition	Note
Vertex	Message Generalization Connector
Note connector	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

Control Panel - State Transition Diagram Editor

The following symbols are available in the control panel of the State Transition Diagram Editor. (Arranged according to their position in the control panel.)

State	Super State
Start State	Final State
STD Class	Note
Vertex	Transition
Event Message	Note connector
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

Name Conflict dialog box

(In: Diagram Editors)

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:

Overwrite	Overwrites the existing item name: the existing item is replaced by the current item.
Refer	Makes a reference to the existing item: the current item is replaced by the existing item.
Cancel	Discards the renaming operation and quits the dialog box.

New State Transition Diagram dialog box

Class Name	Enter the name of the class you are creating the new State Transition Diagram for.
Name	Enter the name of the new State Transition Diagram.

Press the *OK* button to create the new diagram, the *Edit* button to create the new diagram and open an STD editor, or the *Cancel* button to discard the operation.

The name of the new State Transition Diagram has the format:

```
<Class Name>:<Name>
```

Properties dialog box (items and components)

menu entry: **Item > Edit Properties**
menu entry: **Item > Show Properties**

[Dialog Box Layout]
[Storage of Properties]

[Properties in CADs]
[Properties in UCDs]

You can specify the property values of selected diagram objects in this dialog box. The list of available properties depends on:

- The current phase
- The selected diagram object(s)

For almost every object in every diagram in every phase you can specify the property *Free Text*. This property allows you to enter unrestricted text about the object, like comments, motivations, and revision data.

Like most properties Free Text is inherited by newer versions of the object.

Dialog Box Layout

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.
- A *book* at the right that allows you to edit or view the properties of the selected item. These properties are arranged under different tab pages. Examples of tab pages are:
  - **Text** : contains the property *Free Text*
  - **Misc** : contains most other properties
  - **RI** : contains properties concerning referential integrity

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.

Properties in CADs

The properties you can specify in Class Association Diagrams in the Object Design phase affect the code generation. These properties are therefore target language-specific.

For more information on these properties: Refer to the *ObjectTeam Code Generation Guide* of the applicable target language (C++, SmallTalk, Informix NewEra, and so on).

Properties in UCDs

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.

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.

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 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)
(In: 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 Association 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 events received by classes can be:

- Communication Message
- ETD Events
- 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
- navigate between classes and container classes

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 **Color** menu 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.

In the DFD, you can create a branching flow by connecting mulitple flows to a vertex; in this case, deleting the vertex also deletes all the flows connected to that vertex.

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*).

―――

Dialog box

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.

Folding and unfolding a class

(In: Diagram Editors)

Use **Edit > Fold** to reduce the size of a CAD class or a CAD Container class by hiding its attribute section and operation section:

```
 +---------------+       +----------------+
 |               |       |                |
 +---------------+       +----------------+
 |               |           (folded)
 +---------------+
 |               |
 +---------------+
    (unfolded)
```

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 CADs, actors and data stores in DFDs, initiators in ETDs, 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.

____

Dialog Box

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.

Manipulating Item Properties

(In: Diagram Editors)

Use the following **Item** menu entries to *edit*, *show* and *delete* property values of:

- Items referred to in a diagram objects
- Components

Item > Edit Properties...
Use this option to change the property values of the item and components referred to in the selected object(s) using the Properties dialog box.

Item > Delete Properties...
Use this option to delete all the property values of the items and components referred to in the selected object(s).

Item > Show Properties...
Use this option to view the property values of the items and components referred to in the selected object(s) using the Properties dialog box.

You can select more than one object in the drawing area by dragging the mouse cursor around the objects of your choice. The objects included in the rectangle are then selected.

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.

Opening a diagram from another diagram

(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.

Load a specified diagram
: by selecting or entering its name in the Open By Name dialog box.

Reload the current diagram
: by selecting or entering its name in the Open By Name dialog box.

Create a new diagram
: by entering its name in the Open By Name dialog box.

Some diagram names consist of two parts: a qualifying and a qualified part. In that case, both parts must be entered separated by a colon.

If the diagram you try to open is already being edited, it opens in read-only mode.

Opening Diagrams through Diagram Objects

(In: Diagram Editors)

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, 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.

Class Association Diagram

In a Class Association Diagram the following components allow opening or creation of other diagrams:

CAD Class /
CAD Container Class /

Data Type of (Link)Attribute /

Data Type ofMethod /

Data Type ofParameters
Opening these items lead up to one of the following actions:

- Creating or opening a decomposition diagram of the component
- Creating or opening a Class Communication Diagram with name of the component
- Creating or opening a Data Flow Diagram with name of the component
- Creating or opening a Event Trace Diagram with name of the component
- Creating or opening a State Transition Diagram with name:
  `<component_name>:top`

Class Communication Diagram

In a Class Communication Diagram the following components allow opening or creation of other diagrams:

CCD Class /
CCD Class Reference /

CCD Container Class
Opening this type of component leads up to one of the following actions:

- Creating or opening a Class Association Diagram with the name of the component
- Creating or opening a decomposition diagram of the component
- Creating or opening an Event Trace Diagram with the name of the component
- Creating or opening a State Transition Diagram with the name:

`< component_name>:top`

CCD Actor Opening a CCD Actor leads up to one of the following actions:

- Creating or opening a Class Communication Diagram with the name of the component
- Opening a Class Association Diagram which contains a class with the name of the component

Subject Opening a Subject leads up to one of the following actions:

- Creating or opening a Class Association Diagram with the name of the component
- Create or open a system with the name of the component

Communication Message
Opening a Communication Message leads up to:

- Creating or opening a Message Generalization Diagram with the name of the component

----

Data Flow Diagram

In a Data Flow Diagram the following components allow opening or creation of other diagrams:

Data Process Opening a Data Process leads up to:

- Creating or opening a decomposition diagram of the component

DFD Actor Opening a DFD Actor leads up to:

- Creating or opening a Class Association Diagram with the name of the component.

Data Store /
Data Flow /
Update Flow / Opening this type of component leads up to:

- Creating or opening of a Class Association Diagram with the name of the data type of the component.

Event Trace Diagram

In an Event Trace Diagram the following components allow opening or creation of other diagrams:

ETD Event Opening an ETD Event leads up to:

- Creating or opening a Message Generalization Diagram with the name of the component

ETD Object Opening an ETD Object leads up to one of the following actions:

- Creating or opening a Class Association Diagram with the name of the component
- Creating or opening a Class Communication Diagram with the name of the component
- Creating or opening a Event Trace Diagram with the name of the component
- Creating or opening a State Transition Diagram with the name:
  `<component_name>:top`

Message Generalization Diagram

In a Message Generalization Diagram the following components allow opening or creation of other diagrams:

Message Definition

Opening a Message Definition leads up to:

- Creating or opening a decomposition diagram of the component.

State Transition Diagram

In a State Transition Diagram the following components allow opening or creation of other diagrams:

STD Class Opening a STD Class leads up to:

- Creating or opening a Class Association Diagram with the name of the component

State Action /
Transition /
Event Message /
Activity Opening this type of component leads up to:

- Creating or opening a decomposition diagram of the component.

Use Case Diagram

In a Use Case Diagram the following components allow opening or creation of other diagrams:

Actor Opening an actor leads to:

- Creating or opening a Class Association Diagram
- Creating or opening a Class Communication Diagram
- Creating or opening a Event Trace Diagram

Use Case Generalization

Opening a use case generalization leads to:

- Creating or opening a Message Generalization Diagram

Use Case Opening a use case component leads to:

- Creating or opening a Use Case Diagram
- Creating or opening a Event Trace Diagram

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.

Dialog box

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.

Printing diagrams

(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.

____

Dialog Box

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.

Replacing a diagram object

(In: Diagram Editors)

Use **Edit > Replace** to replace one symbol with another symbol. To replace a symbol, do the following:

1. Select the new symbol from the control panel.
2. Select the symbol you want to replace in the drawing area.
3. Select **Edit > Replace**.

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.

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 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.

Syntax Dialog Box

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 Association 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*

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 > ZoomIn** and **ZoomOut** to increase or 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
    (CADE only)
  - Unfold
    (CADE 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
  (CADE only)

- **Check** menu
    - Contents
    - Local Model
      (not in MGDE)
    - 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 Association Diagram (CAD)

Class Association 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 Association Diagram is represented by the file version object of type *cad*. This object is a child object of the browser object SystemVersion. So the object *cad* appears in the information area on:

- System level

By opening the object *cad* from the browser on System level, you start the Class Association Diagram Editor.

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

```
Corporate
|
+- Project
   |
   +- ConfigVersion
      |
      +- PhaseVersion
         |
         +- SystemVersion
            |
            +- {file version}
```

Class Communication Diagram (CCD)

You use *Class Communication Diagrams* to model the interaction and communication between classes. The diagrams show which classes or group of classes communicate and what the contents of the communication are.

In the browser, a Class Communication Diagram is represented by the file version object of type *ccd*. This object is a child object of the browser object SystemVersion. So the object *ccd* appears in the information area on:

- System level

By opening the object *ccd* from the browser on System level, you start the Class Communication Diagram Editor.

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

```
Corporate
|
+- Project
   |
   +- ConfigVersion
      |
      +- PhaseVersion
         |
         +- SystemVersion
            |
            +- {file version}
```

Class Definition Matrix (CDM)

When you enter attributes or operations for a class in a Class Association 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 CAD in which the CDM's class appears. If the class appears in more than one CAD, a dialog box appears prompting you to select the CAD that you want to open.

Data Flow Diagram (DFD)

Operations in classes transform data values. Where this data comes from, where it goes to and what processes exist is modeled in a *Data Flow Diagram*.

In the browser, a Data Flow Diagram is represented by the file version object of type *dfd*. This object is a child object of the browser object SystemVersion. So the object *cad* appears in the information area on:

- System level

By opening the object *dfd* from the browser on System level, you start the Data Flow Diagram Editor.

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

```
Corporate
|
+- Project
   |
   +- ConfigVersion
      |
      +- PhaseVersion
         |
         +- SystemVersion
            |
            +- {file version}
```

Event Trace Diagram (ETD)

An event trace is an ordered list of events between objects. It describes a single scenario of events that can take place in the real world. The *Event Trace Diagram* usually models objects rather than classes.

In the browser, an Event Trace 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 *etd* from the browser on System level, you start the Event Trace Diagram Editor.

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

```
Corporate
|
+- Project
   |
   +- ConfigVersion
      |
      +- PhaseVersion
         |
         +- SystemVersion
            |
            +- {file version}
```

Item

Contents:

Items are not specified directly, but generated from symbol labels in diagrams. An unnamed component is just a component, but a named component creates an item in the repository. Items relate objects of certain types with their name. For example: when you open a class in a CAD, 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. The diagram and system refer to the same item as the class.

Items appear in the system of their definition under the pseudo object **<defined items>**.

Item -Component relations

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*.

CAD (item type cl)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	cad_class	name	cl	*P*, S
		attributes	-	-
		methods	-	-
	cad_container	name	cl	*P*, S
		attributes	-	-
		methods	-	-

	nary-association	name	cl	P, S, *F*
	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	-	-

CCD (item type cl)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	ccd_class	name	cl	*P*, S
	ccd_class_ref	name	cl	*P*, S
	ccd_container	name	cl	*P*, S
	subject	name	cl	*P*, S
	ccd_actor	name	cl	*P*, S
Connectors	com_message	name	pe	*S*

DFD (item type pe)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	data_proc	name	pe	P, *S*
	data_store	name	de	*S*
	dfd_actor	name	cl	*P*, S
	dfd_anchor	-	-	-
	data_vertex	-	-	-
	ctrl_vertex	-	-	-
	update_vertex	-	-	-
	result_vertex	-	-	-
Connectors	data_flow	name	de	*S*
	update_flow	name	de	*S*
	result_flow	-	-	-
	ctrl_flow	name	pe	*S*

ETD (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*

MGD (item type pe)

Symbol Type	Component Name	Label	Item Type	Item Scope
Nodes	message_def	name	pe	*S*
	message_gen	-	-	-
Connectors	message_gen_conn	-	-	-

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 Nodes	state_action	event	pe	Q
			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*

Message Generalization Diagram (MGD)

For better readability and standardization of interface, the *Message Generalization Diagram* allows you to separately model multiple messages between two classes in a Class Communication Diagram.

In the browser, a Message Generalization Diagram is represented by the file version object of type *mgd*. This object is a child object of the browser object SystemVersion. So the object *mgd* appears in the information area on:

- System level

By opening the object *mgd* from the browser on System level, you start the Message Generalization Diagram Editor.

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

```
Corporate
|
+- Project
   |
   +- ConfigVersion
      |
      +- PhaseVersion
         |
         +- SystemVersion
            |
            +- {file version}
```

State Transition Diagram (STD)

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}
```

Aggregation - symbol

[Labels] [Properties]

(In: Class Association Diagram Editor)

An *aggregation* is a special form of an association. You can specify the multiplicity of an aggregation through the multiplicity balls in the control panel.

Labels

The connector representing the aggregation has the following labels:

```
|                 |
+-----------------+
        / \
        \ /
   [(1)] | (2)
         |
         |
    Aggregation
        Name
         |
   [(1)] | (2)
```

Aggregation Name

The name of the aggregation.

(1) = *constraint* The multiplicity of an aggregation can be restricted numerically with this label. Depending 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>=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.

Association - symbol

[Labels] [Properties]

(In: Class Association Diagram Editor)

Class symbols can be related by the *association* symbol. You can specify the multiplicity of an association through the multiplicity balls 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>=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.

CAD Class - Attributes section

[Labels] [Properties] [Scope] [Navigation]

(In: Class Association Diagram Editor)

The middle section of a CAD 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 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.

CAD Container Class - symbol

[Labels]

(In: Class Association Diagram Editor)

A *CAD Container Class* symbol represents a container class object.

Labels

A container class symbol has three sections. Each of these has a label:

```
  +----------------------+
+----------------------+ |
| Container Class Name | |
|----------------------| |
|attribute             | |
|...                   | |
|----------------------| |
|operation             | |
|...                   |-+
+----------------------+
```

The CAD container class resembles the CAD class in many respects, so refer for details to the help topic CAD class.

The name of the symbol refers to an item.

CAD Class - symbol

[Labels] [Properties]

(In: Class Association Diagram Editor)

A *CAD Class* symbol represents a class object in a Class Association Diagram. It is represented unfolded or folded.

Labels

A class symbol has three sections. Each of these has a label:

```
+------------------+
|    Class Name    |
|------------------|
|attribute         |
|...               |
|------------------|
|operation         |
|...               |
+------------------+
```

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.

CCD Actor - symbol

(In: Class Communication Diagram Editor)

A *CCD Actor* symbol represents an object from the external world.
CCD Actors refer to the same objects as actors in Data Flow Diagrams.

Labels

The CCD Actor has one label:

```
   ----------------
  |                |
  |    Actor Name  |
  |                |
   ----------------
```

Assigning an Actor 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 CCD actor symbol from the drawing area and select File > Open.

CCD Class - symbol

(In: Class Communication Diagram Editor)

A *CCD Class* symbol represents a class object in a Class Communication Diagram.

Labels

The CCD Class has one label:

```
+-------------------+
|                   |
|    Class Name     |
|                   |
+-------------------+
```

Assigning a name to a CCD Class symbol is *compulsory*. This name must refer to a CAD class defined in a Class Association Diagram.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a ccd class symbol from the drawing area and select File > Open.

CCD Container Class - symbol

(In: Class Communication Diagram Editor)

A *CCD Container Class* symbol is a class that implements general purpose data structures.

Labels

```
  +---------------------+
+---------------------+ |
|                     | |
|Container Class Name | |
|                     |-+
+---------------------+
```

Assigning a *Container Class Name* to a CCD Container Class symbol is *compulsory*.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a CCD container class symbol from the drawing area and select File > Open.

CCD Class Reference - symbol

(In: Class Communication Diagram Editor)

Labels

A *CCD Class Reference* symbol in a Class Communication Diagram Editor represents a class object. A Class Reference refers to a CAD class or a CAD container class.

A CCD Class Reference has one label:

```
 . . . . . . . . . . . . . . . . . . .
 :                                   :
 :       Class Name                  :
 :                                   :
 :. . . . . . . . . . . . . . . . . .:
```

Assigning a *Class Name* is *compulsory*. This name must refer to a CAD class or a CAD container class in a Class Communication Diagram.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a ccd class reference symbol from the drawing area and select File > Open.

Control Flow - symbol

[Labels] [Properties]

(In: Data Flow Diagram Editor)

A *Control Flow* symbol is a connector indicating an event.

Labels

The Control Flow symbol has one label:

```
        - - - Flow Name- - ->
```

Assigning a *Flow Name* is compulsory.

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.

CAD Class - Class Name

[Labels] [Properties] [Scope] [Navigation]

(In: Class Association Diagram Editor)

The top section of a CAD 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*.

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 a CAD class is *Phase*. You can change the scope with Item > Edit Scope... to *System*.

Navigation

You can open or create an item-related diagram by selecting a CAD class symbol from the drawing area and select File > Open. Select the class name in the dialog box.

Communication Message - symbol

(In: Class Communication Diagram Editor)

A *Communication Message* symbol represents messages or message sets. Sets of messages are defined in Message Generalization Diagrams.

Labels

The Communication Message has one label:

```
----Message Name---->
```

Assigning a Message 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 communication message symbol from the drawing area and select File > Open.

Constraint - symbol

(In: Class Association Diagram Editor)

A *constraint* symbol is a connector indicating a general constraint between:

- CAD class - CAD class
- association - association
- CAD 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:

```
      --+                 +---
        |    Constraint   |
        +......name......>|
        |                 |
      --+                 +---
```

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.

DFD Actor - symbol

(In: Data Flow Diagram Editor)

A *DFD Actor* or Terminator symbol represents an object in the real world that produces or accepts data.

Labels

```
+-------------------+
|                   |
|  DFD Actor Name   |
|                   |
+-------------------+
```

Assigning a *DFD Actor Name* is *compulsory*. This name must correspond to the name of an existing CAD Class.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a DFD actor symbol from the drawing area and select File > Open.

DFD Anchor - symbol

(In: Data Flow Diagram Editor)

A *DFD Anchor* symbol is a drawing utility used to start or end flows. It is displayed as a large square dot.

Data Flow - symbol

[Labels] [Properties]

(In: Data Flow Diagram Editor)

A *Data Flow* symbol is a connector indicating transport of data.

Labels

The Data Flow symbol has one label:

```
        ----Flow Name---->
```

Assigning a *Flow Name* is compulsory unless the data flow connects a data process with a data store.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a data flow symbol from the drawing area and select File > Open.

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.

Data Process - symbol

(In: Data Flow Diagram Editor)

A *Data Process* symbol represents an operation on data. It must have at least one input flow and one output flow.

Labels

A Data Process symbol has the following label:

```
   ______________
  /              \
 /                \
 | Process Name |
 \                /
  \______________/
```

Assigning a *Process Name* to a Data Process symbol is *compulsory*.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a data process symbol from the drawing area and select File > Open.

DFD Data Store - symbol

[Labels] [Properties]

(In: Data Flow Diagram Editor)

A *DFD Data Store* symbol represents a passive storage place for data.

Labels

```
--------------------

 DFD Data Store Name

--------------------
```

Assigning a *DFD Data Store name* is *compulsory*. This name must correspond to the name of a standard type or an existing CAD Class.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a data store symbol from the drawing area and select File > Open.

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.

ETD Event - symbol

(In: Event Trace Diagram Editor)

A *ETD Event* symbol represents an event occurring between two objects.

Labels

An ETD Event has the following label:

```
      ------ETD Event Name------>
```

Assigning an *ETD Event Name* is *compulsory*.
The default syntax for this name is:

```
  <event_name>[(<attr_list>)]
```

`<attr_list>` is defined as:

```
  <attr_list> : = <attribute_name> [,<attr_list> ]
```

The label of the symbol refers to a number of items.

Navigation

You can open or create an item-related diagram by selecting an ETD event symbol from the drawing area and select File > Open.

ETD Initiator - symbol

(In: Event Trace Diagram Editor)

An *ETD Initiator* symbol represents an external object that initiates an event trace. There can be only one ETD Initiator in an Event Trace Diagram.

Labels

The ETD Initiator has one label:

```
     ETD Initiator Name
             |
             |
             |
             |
             |
             |
```

Assigning an *ETD 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.

ETD Object - symbol

[Labels] [Navigation]

(In: Event Trace Diagram Editor)

An *ETD Object* symbol represents an object participating in a sequence of events between two or more objects. An ETD Object refers to an object of a Class.

Labels

```
     ETD Object Name
            :
            :
            :
            :
            :
            :
```

Assigning an *ETD Object Name* is *compulsory*. This name must correspond to the name of an existing CAD Class or CAD Container Class. 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.

Navigation

You can open or create an item-related diagram by selecting an ETD object symbol from the drawing area and select File > Open.

Event Message - symbol

(In: State Transition Diagram Editor)

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:

```
--------------->
      :
      :
    event
      :
      V
+-----------+
|           |
```

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

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.

Generalization - symbol

[Labels] [Properties]

(In: Class Association Diagram Editor)

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:

```
           |
           |
          /#\ discriminator
    +-------------+
    |             |
    |             |
```

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.

Link Attribute Box - symbol

[Labels] [Navigation] [Scope] [Properties]

(In: Class Association Diagram Editor)

A *Link Attribute Box* contains the attributes of an association or an aggregation. Through a loop, it can be attached to an:

- Association
- Qualified association
- Aggregation
- Qualified aggregation

Labels

A link attribute box contains one part in which one or more link attributes can be specified.

```
+----------------+
|----------------|
| link attribute |
| ...            |
+----------------+
```

Defining at least one link attribute in a link attribute box is *compulsory*. The syntax for link attributes is equivalent to the syntax for attributes in a CAD class symbol.

The name and type labels of the symbol refer to items.

Navigation

When you select a link attribute symbol and then select File > Open, the 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.

Scope

Link Attribute	The scope of a link attribute is *File*.
Data Type	The initial scope of data types is *Phase*. You can change the scope with Item > Edit Scope... to *System*.

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.

Loop - symbol

(In: Class Association Diagram Editor)

A *loop* is used to connect a CAD class symbol or a link attribute box to an association. The association can be of the following type:

- Association
- Qualified association
- Aggregation
- Qualified aggregation
- N -ary association symbol

```
      ----Association Name----
              |    |
              |    |
               \  /
          +-------------+
          |             |
```

More Classes - symbol

(In: Class Association Diagram Editor)

. ..

A *More Classes* symbol indicates that more subclasses exist in a super-subtype relation.

Message Definition - symbol

(In: Message Generalization Diagram Editor)

A *Message Definition* symbol represents a message or a set of messages.

The symbol can act as top and as leaf symbol. As top symbol it can occur only once in a Message Generalization Diagram.

Labels

The Message Definition symbol has the following label:

```
+---------------------+
|                     |
| Message Definition  |
|         Name        |
|                     |
+---------------------+
```

Assigning a *Message Definition Name* is *compulsory*. The name of a top symbol must be equal to the name of the diagram itself.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a message definition symbol from the drawing area and select File > Open.

Message Generalization Connector - symbol

(In: Message Generalization Diagram Editor)

A *Message Generalization Connector* symbol describes the relationship between a Message Definition symbol and its sub Message Definitions.

A generalization symbol consists of a node part and a connector part. The node part is an empty triangle:

```
       |
       |
      / \
+--------------+
|              |
|              |
```

Multiplicity Balls - symbol

(In: Class Association Diagram Editor)

At the bottom of the control panel (beneath the select button) there are two columns, containing three buttons each. The buttons in the left column can be used to set the multiplicity for the start of an association or aggregation; the buttons in the right column can be used to set the multiplicity for the end of the association:

```
 Begin Mandatory Association   End Mandatory Association
 Begin Optional Association    End Optional Association
 Begin Many Association        End Many Association
```

An optional relation is represented as an open ball; a mandatory relation as a solid ball. Given a Begin Mandatory Association, the multiplicity variations are as follows:

- *Begin Mandatory Association - End Mandatory Association*
  1 : 1 relation (mandatory); represented as:

  ```
  ---------
  ```

  (multiplicity balls on neither side)
- *Begin Mandatory Association - End Optional Association*
  1 : 0 or 1 (optional) relation; represented as:

  ```
  --------o
  ```

- *Begin Mandatory Association - End Many Association*
  1 : 0 or more (optional) relation, represented as:

  ```
  --------*
  ```

- *Begin Mandatory Association - End Many Association*
  1 : 1 or more (mandatory) relation; represented as:

  ```
        1+
  --------*
  ```

N-ary Association - symbol

[Labels] [Properties] [Scope]

(In: Class Association Diagram Editor)

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:

```
           / \
          /   \
         /     \
     --- name  --
         \     /
          \   /
           \ /
            |
```

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 - symbol

[Labels]

(In: Class Association Diagram Editor)

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 balls in the control panel.

Labels

The connector representing the n-ary association connector has the following labels:

```
               / \
              /   \
   [(1)]     /     \   [(1)]
  -------- name   --------
   (2)      \     /     (2)
             \   /
              \ /
               |
               |
         [(1)] | (2)
```

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>=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.

CAD Class - Operation section

[Labels] [Properties] [Scope] [Navigation]

(In: Class Association Diagram Editor)

The bottom section of a CAD 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 - symbol

[Labels]

(In: Class Association Diagram Editor)

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.

A propagation symbol can be attached to the following symbols:

- Association
- Qualified association
- Aggregation
- Qualified aggregation
- N -ary association connector

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 - symbol

[Labels] [Properties]

(In: Class Association Diagram Editor)

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 balls in the control panel.

Labels

The connector representing the qualified aggregation has the following labels:

```
|                  |
+---+---------+---+
    |qualifier|
    +---------+
        / \
        \ /
   [(1)] | (2)
         |
         |
    Aggregation
        Name
         |
   [(1)] | (2)
```

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>=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 - symbol

[Labels] [Properties]

(In: Class Association Diagram Editor)

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 Association Diagrams.

You can specify the multiplicity of a qualified association through the multiplicity balls in the control panel.

Labels

The connector representing a qualified association has the following labels:

```
--+
  +---------+ [(1)]                        [(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>=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.

Result Flow - symbol

(In: Data Flow Diagram Editor)

A *Result Flow* symbol is used to indicate the creation of a new object in a data store.
A Result Flow is represented by a connector with an open arrow-head. Its only possible destination is a data store symbol.

---------->

STD State - symbol

[Labels] [Navigation]

(In: State Transition Diagram Editor)

An *STD State* symbol represents a state or activity of a class in a State Transition Diagram.

Labels

The two sections of a state symbol represent the following:

```
 ----------------
/   State Name   \
|                |
|                |
|                |
| actions/events |
\________________/
```

Explanation:

State Name	Assigning a State name is *optional* if actions/events are defined. If they are not defined, a State Name is *compulsory*.
actions /events	Assigning actions/events is *optional* if a State Name is defined. If no State Name is defined, actions/events are *compulsory*. Specified actions/events 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 Class - symbol

(In: State Transition Diagram Editor)

A *STD State Class* symbol represents a class receiving an event. It is connected to a transition by an event message symbol.

Labels

The STD State Class has the following label:

```
              :
              :
              :
              V
       +-------------+
       |             |
       | State Class |
       |    Name     |
       +-------------+
```

Assigning a *State Class Name* is *compulsory*. This name must correspond to an existing CAD class or CAD Container 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.

STD Start State - symbol

(In: State Transition Diagram Editor)

A *STD Start State* symbol symbolizes the initial state of a object in a State Transition Diagram. It is represented by a black dot.

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.

Subject - symbol

(In: Class Communication Diagram Editor)

A *Subject* symbol represents a group of classes or a system.

Labels

The Subject has the following symbol:

```
+---------------------+
|+-------------------+|
||                   ||
||  Subject Name     ||
||                   ||
|+-------------------+|
+---------------------+
```

Assigning a *Subject Name* is *compulsory*. This name must correspond to a Class Association Diagram or a Class Communication Diagram.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting a subject symbol from the drawing area and select File > Open.

STD Super State - symbol

(In: State Transition Diagram Editor)

A *STD Super State* symbol represents a state of a class in a State Transition Diagram.

Super States do not support:

- Entry -action or activities
- Concurrent sub diagrams
- Control splitting or synchronization

Labels

The super state symbol represent the following:

```
 ____________________
/  Super State Name  \
|     ___            |
|    (___)           |
|     _|______       |
|    /         \     |
|    \_________/     |
\____________________/
```

Assigning a *Super State Name* is *optional*.

The name of the symbol refers to an item.

Transition - symbol

[Labels] [Navigation]

(In: State Transition Diagram Editor)

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
- Super States
- Final States (only as target)

Labels

The transition symbol has one label:

```
   __                   __
     \                 /
      |               |
      |----event----->|
      |               |
  __ /                 \__
```

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.

Update Flow - symbol

(In: Data Flow Diagram Editor)

An *Update Flow* symbol is a connector indicating transport of data.

Labels

The Update Flow symbol has one label:

```
<---Flow Name--->
```

Assigning a *Flow Name* is compulsory unless the data flow connects a data process with a data store.

The name of the symbol refers to an item.

Navigation

You can open or create an item-related diagram by selecting an update flow symbol from the drawing area and select File > Open.

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.

Vertex - symbol

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.)

In the Data Flow Diagram Editor, you can also use the vertex to split or join connectors.

```
 /
/
\
 \
```

Aggregating Classes

(In: Class Association Diagram Editor)

Instructions

1. Make sure that the correct type of multiplicity balls 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 Association Diagram Editor)

Instructions :

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...

Changing Multiplicity

(In: Class Association Diagram Editor)

Instructions :

1. Select the association symbol you want to change the multiplicity of.
2. Select an association symbol from the control panel.
3. Select the appropriate connector beginning and connector end from the control panel.
4. Select Edit > Replace.

The multiplicity of the association is now changed.

Connecting a Message Definition

(In: Message Generalization Diagram Editor)

Instructions :

1. Select a Message Generalization Connector from the control panel,
2. Click on the Super-Message Definition in the drawing area,
3. Click on the first Sub-Message Definition.

Connections to subsequent Sub-Message Definitions are started from the triangle of the Message Generalization Connector.

Creating a Communication object

(In: Class Communication Diagram Editor)

We consider the following symbols as Class Communication objects:

- CCD Classes
- CCD Class Reference
- CCD Container Class
- CCD Actor
- Subject

All these components have the same drawing rules.

Instructions :

1. Select a component from the control panel.
2. Insert the symbol at the intended spot in the drawing area.
3. Enter the name of the component in the symbol.

Creating a Class

(In: Class Association Diagram Editor)

Instructions :

1. Select a CAD class symbol or a CAD container 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...

Creating a Message Definition

(In: Message Generalization Diagram Editor)

Instructions :

1. Select a Message Definition from the control panel.
2. Insert the symbol at the intended spot in the drawing area.
3. Enter the name in the Message Definition.

Defining an Association as Class

(In: Class Association Diagram Editor)

Instructions :

1. Insert a class symbol
2. Select the loop symbol from the control panel.
3. Press the left mouse button on the association symbol you want to define as class.
4. Move the cursor away from the association.
5. Press the left mouse button in the appropriate class symbol.
   The association and the class are now connected with the loop. Note that the association and the class must have a *name*.

To complete a class definition:

- define its attributes
- define its operations
- define its properties

Defining Attributes

(In: Class Association Diagram Editor)

Instructions :

1. Move the cursor in the drawing area to the Attribute section of the intended CAD class symbol.
2. Enter the appropriate attribute information.

To complete an attribute definition:

- define its properties using Item > Edit Properties...

Defining Operations

(In: Class Association Diagram Editor)

Instructions :

1. Move the cursor in the drawing area to the Operations section of the intended CAD class symbol.
2. Enter the appropriate operation information.

To complete an operation definition, define its properties using Item > Edit Properties.

Defining an Ordered Association

(In: Class Association Diagram Editor)

Instructions :

1. Move the cursor in the drawing area to the association symbol you want to specify as "ordered".
2. Enter the following text as multiplicity constraint:

```
{ ordered}
```

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 Association Diagram Editor)

Instructions :

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.

Instructions :

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.

Class Association Diagram Editor

Use this diagram editor to draw Class Association 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 Communication Diagram Editor

Use the *Class Communication Diagram Editor* (CCDE) to draw Class Communication Diagrams. The objectives of this diagram technique are to:

- define messages or groups of messages between classes
- model communication with the outside world or other systems
- model communication with data structures

Message sets can be resolved in the Message Generalization 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.

Data Flow Diagram Editor

Use the Data Flow Diagram Editor (DFDE) to draw Data Flow Diagrams.
The objectives of this diagram technique are to:

- show the flow of data within and between objects
- show the processes that modify the data flows

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 Editor

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 Association Diagram Editor
- Class Communication Diagram Editor
- Data Flow Diagram Editor
- Event Trace Diagram Editor
- Message Generalization Diagram Editor
- 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.

```
+==================================================+
|                    [Menu Bar]                    |
|--------------------------------------------------|
|                     [Tool Bar]                   |
|--------------------------------------------------|
|                  [Context Area]                  |
|                                                  |
|-------+------------------------------------------|
|[C  P  |                                          |
: o  a  :                                          :
: n  n  :             [Drawing Area]               :
: t  e  :                                          :
| r  l] |                                          |
| o     |                                          |
| l     |                                          |
|-------+------------------------------------------|
|                 [Message Area]             /\ \/ |
+==================================================+
```

Event Trace Diagram Editor

Use the Event Trace Diagram Editor (ETDE) to draw Event Trace Diagrams.
The objectives of this diagram technique are:

- To define events or messages between objects
- To define the sequence of the events

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.

Message Generalization Diagram Editor

Use the Message Generalization Diagram Editor (MGDE) to draw Message Generalization Diagrams. These diagrams can be used to decompose a communication message in a Class Communication Diagram or a communication association in a Use Case Diagram.

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.

State Transition Diagram Editor

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 - diagram editor

The *Context Area* in a diagram editor is located between the tool bar and the drawing area. It contains the following fields:

Project	System
Phase	Diagram Version

Drawing Area

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

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.

Control Panel - Use Case Diagram Editor

The following symbols are available in the control panel of the Use Case Diagram Editor. (Arranged according to their position in the control panel.)

Use Case	Use Case Actor
Note	Vertex
Undirected Communication Association	Directed Communication Association
Use Case Generalization	Note connector
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

Use Case Diagram (UCD)

A *Use Case Diagram* models the way in which an actor can use the system. You can then use the Event Trace 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}
```

Communication Association - symbol

(In: Use Case Diagram Editor)

There are two types of Communication Associations:

- **Undirected Communication Associations**, represented by a line, represent two-way communication between the actor and the use case.

```
    ----Name----
```

- **Directed Communication Associations**, represented by a line and arrow, represent one-way communication. The arrowhead shows the direction of communication.

```
    ----Name---->
```

You can use a Message Generalization Diagram to specify the messages represented by a Communication Association.

Use Case 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.

– – – – – – – – – >

Use Case Actor - symbol

(In: Use Case Diagram Editor)

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 has the following label:

```
+-----------+
|           |
|   Name    |
|           |
+-----------+
```

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.

Use Case - symbol

(In: Use Case Diagram Editor)

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:

```
         ______________
        /              \
       /                \
      |       Name       |
       \                /
        \______________/
```

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.

Changing the editing focus

(In: Diagram Editors)

A text field must have the *editing focus* before you can edit the text in the field. 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.

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.

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 and a value of *pointer* indicates the UNIX-like behavior.

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 can be used to add comments to your diagram.

The contents of the notes will not be generated in a report.

Notes can be created and edited as any component in a diagram. The note font can be changed when Options | Font | Annotation... is selected. By default, it 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.

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.
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 string to be passed to the USING clause of the CONNECT command. 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.

Archive Command dialog box

(In: Corporate Management Tool)

Select **Options > Archive Command** to set the default archive command (which is stored in the M4_archive_cmd variable). For UNIX, the default command is:

```
tar cf %F %N
```

For Windows, the default command is:

```
xcopy %R %F /I /E
```

The command is run in the parent directory of the repository directory. The value can contain the following codes, which are expanded by the repository tool before the command is executed. Alternatively, you can use these variables to write a script and then use the archive command to execute the script.

- *% N* name of the repository
- *% P* parent directory of the repository directory
- *% R* repository directory
- *% F* full path of the archive file
- *% D* directory part of the archive file
- *% T* file part of the archive file
- *% W* what the command is supposed to do: *archive* or *unarchive*

Backup Repository dialog box

(In: Corporate Management Tool)

The Backup utility of the Corporate Management tool archives a repository database. To maintain consistency between the database and file system parts of the repository, it is a good idea to shutdown all servers during backup. Click on the Shutdown button to stop all servers.

Fill in the desired options (explained below) and click OK to proceed with the backup of the repository.

Archive Repository Directory

It is strongly recommended that you always select both of these options; otherwise, inconsistencies between your file system part of the repository and the database part may occur.

Destination File	Enter the pathname of the directory in which you want to save your repository.
Command to archive repository directory	The default archive command is filled in here. You can change this here if you wish, or set it through the Archive command option.
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 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:

Database	The name of the database used in the selected repository.
Directory	The directory in which the file system part of the repository is stored.
Server Entry	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.

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 dbservers can be shutdown. Therefore, all clients must be exited before you can optimize the repository.

Restore Repository dialog box

(In: Corporate Management Tool)

The Restore utility of the Corporate Management tool unarchives an archive file produced by the Backup utility. To maintain consistency between the database and file system parts of the repository, it is a good idea to shutdown all servers during the restoration. Click on the Shutdown button to stop all servers.

Fill in the desired options (explained below) and click OK to proceed with restoring the repository.

Corporate Name	The name of the repository. This must be the same name as when you backed up the repository, otherwise the repository will be unusable. By default, the name of the repository selected in the Corporate Management tool is entered here.
Directory in which to restore the repository	The directory where you want the file system part of the repository to be stored.
Register Implementation	Add an entry for this repository in the objservers file.
Overwrite existing database	This specifies that, if the name of the database is the same as an existing one, the tables in the existing database are overwritten.
Unarchive dump	This unarchives the file system part of the repository into the repository directory specified above. By default, this option is selected. To prevent inconsistencies in the repository, leave this option selected.
Source file	The pathname of the archive file. In UNIX, you can use the Browse button to locate it.
Command to unarchive repository archive	The unarchive command.

Database to create/overwrite

You can choose to create a new database or overwrite en existing database. If you want to overwrite an existing database, the Overwrite Existing Database option must be selected. 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.

Unarchive Command dialog box

(In: Corporate Management Tool)

Select **Options > Unarchive Command** to set the default unarchive command (which is stored in the M4_unarchive_cmd variable). For UNIX, the default command is:

```
tar xf %F %N
```

For Windows, the default command is:

```
xcopy %F %R /I /E
```

The command is run in the parent directory of the repository directory. The value can contain the following codes, which are expanded by the repository tool before the command is executed. Alternatively, you can use these variables to write a script and then use the unarchive command to execute the script.

% N name of the repository
% P parent directory of the repository directory
% R repository directory
% F full path of the archive file
% D directory part of the archive file
% T file part of the archive file
% W what the command is supposed to do: *archive* or *unarchive*

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 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 preceeded 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.

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.)

Lock Type	Selects only locks of the specified types. Possible types are read or write.
Hanging Locks	Selects only hanging locks. These are the only locks that can be removed.
Lock Attributes	Allows you to enter the following attributes:

- *Object Id*
  Selects only locks on objects with this identity.
- *Host*
  Selects only locks that have been set by clients on this host.
- *User*
  Selects only locks that have been set by clients started by this user.
- *Process Id*
  Selects only locks that have been set by clients with this process id.
- *Reason*
  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 implemention, 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.

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

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 ...

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 ...

Lock Management Tool

(In: Repository Tool)

This tool can be used to examine, set or remove locks. (Locks can only to 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 ...

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 ...

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.

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 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 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 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.