Telelogic Logiscope

Studio Reference Manual

Version 6.5

Before using this information, be sure to read the general information under Appendix C "Notices", on page **C-1**.

Table of Contents

Chapter 4   **Quality Model Editor**

Chapter 1

Introducing Logiscope Studio

1.1 Purpose of Logiscope *Studio*

Studio is an environment tool to build and visualize the results produced by the Telelogic® Logiscope™ toolset including *CodeReducer, QualityChecker, RuleChecker* or *TestChecker* modules for application written in Ada, C, C++, or Java.

1.2 Files Used by *Studio*

Studio uses and produces several files during a session.

1.2.1 Workspace and Project

A logiscope project (**.ttp**) corresponds to a set of source (and header) files and configuration files: e.g. Rule Set(s), Quality Model, Dynamic Execution files according to the verification module applicable to the projet.

One or more projects can be included in a Logiscope workspace (**.ttw**).

1.2.2 Source Files / Header Files

These files are the source (and header) files constituting the application to analyze.

1.2.3 Quality Model

It is a file (**logiscope.ref** by default) describing the quality model used by *Logiscope QualityChecker*. It specifies the metrics, criteria and factors to qualify the application component.

1.2.4 Rule Set(s)

This file(**.rst**) is used by *Logiscope RuleChecker* to specify the set of rules to be checked on the application source files.

It can also include another rule sets, thus enabling a rule set to be composed of several others.

1.2.5 Dynamic Files

These files (**.dyn** or **.trc**) are the dynamic execution files used by *Logiscope TestChecker* to compute the test coverage.

1.3 Advanced Ergonomics

Studio is a context-sensitive graphical tool. It is consistent and intuitive in its design with palettes, menus, standard "drag and drop" features, as well as undo/redo.

On line context-sensitive help is available by using the Help icon or the F1 key

- to obtain help on the toolbar buttons, click the **Help** icon then click on the desired button;
- to get help on the pop-up menu commands, position the cursor on the desired command then press the <F1> key;
- to get help on menus or areas, use one of the above.

1.4 *Studio* Views

Studio is a multi-view tool, it can show source files as well as metrics values, rule violations or test coverage.

- the **Workspace View** displays the **File View** including Logiscope project files,
- the **Result Pane** displays various windows (source files, html reports, ...),
- the **Output Window** displays project messages, error messages, warnings or results.

These views are presented in detail in the following chapter.

Contacting IBM Rational Software Support

Support and information for Telelogic products is currently being transitioned from the Telelogic Support site to the IBM Rational Software Support site. During this transition phase, your product support location depends on your customer history.

Product support

- If you are a heritage customer, meaning you were a Telelogic customer prior to November 1, 2008, please visit the [Logiscope Support Web site](#).

  Telelogic customers will be redirected automatically to the IBM Rational Software Support site after the product information has been migrated.

- If you are a new Rational customer, meaning you did not have Telelogic-licensed products prior to November 1, 2008, please visit the [IBM Rational Software Support site.](#)

Before you contact Support, gather the background information that you will need to describe your problem. When describing a problem to an IBM software support specialist, be as specific as possible and include all relevant background information so that the specialist can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, or messages that are related to the problem?
- Can you reproduce the problem? If so, what steps do you take to reproduce it?
- Is there a workaround for the problem? If so, be prepared to describe the workaround.

Other information

For Rational software product news, events, and other information, visit the [IBM Rational Software Web site](#).

Chapter 2

Studio Interface

2.1 Starting up Studio

To start up Studio do one of the following:

On Windows:

- open the **Start** menu and select **Programs - Telelogic - Telelogic Logiscope <version>**, or

- in Windows Explorer, double-click a **.ttp** or **.ttw** file.

On UNIX:

- launch **vcs** binary.

2.2 Overview of Studio Windows

The picture below represents Logiscope Studio main windows:
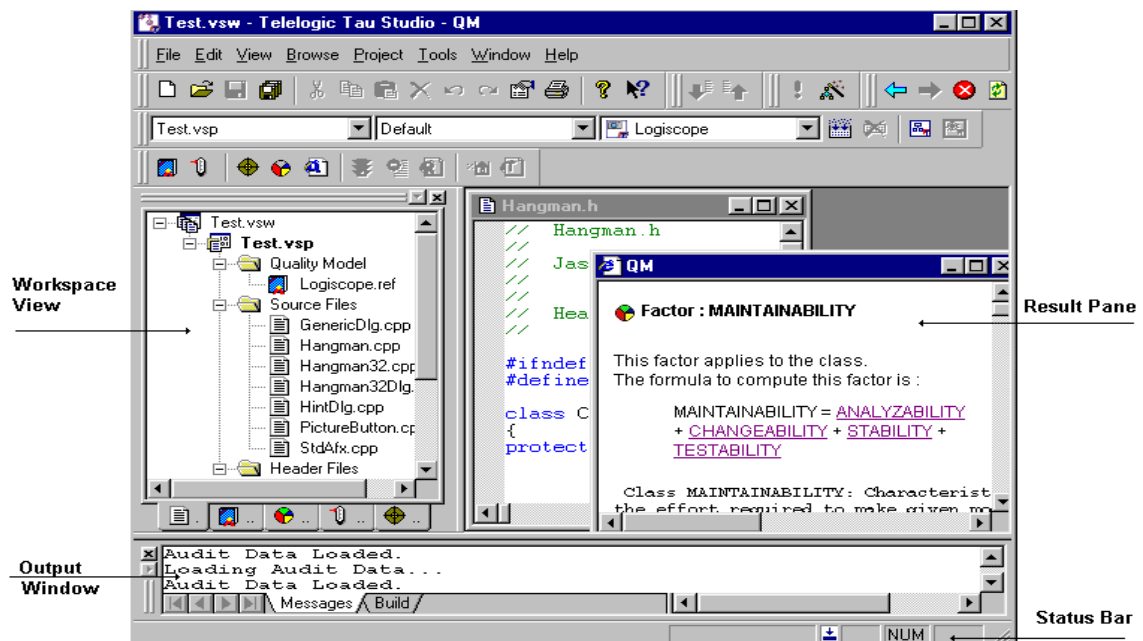
Description of Studio different windows:

Workspace View

The **Workspace View** displays the **File View** (main tab) with the project files: the quality model file, source and header files, rule files and dynamic files. A double click on any item displays it in the result pane.

This window is used to display any information related to the project including:

- the **File View** tab,
- the **Quality** tab,
- the **Dictionary** tab,
- the **Criteria** tab,
- the **Factors** tab,
- the **Metrics Values** tab,
- the **User Request** tab,
- the **Violations** tab,
- the **File Violations** tab,
- the **Components** tab,
- the **Tests** tab.

Result Pane

Displays various result windows: source files, html reports...

Output Window

Displays messages and information on the project. It contains the **Messages**, **Build, Browse** and **Script** tabs used to show warnings, error messages and results.

- the **Messages** tab is used to give information on the project loading.
- the **Build** tab is used to display the analysis results.
- the **Browse** tab is used to visualize metrics, rules or test information.
- the **Script** tab is used to view the Tcl execution results.

Status Bar

A display area for:

- help on toolbars buttons,

- location (Ln x, Col y) in the active window from result pane,

- the tool execution status: when running, the indicator is ⬚, otherwise it is ⬚ .

*Note: Status bar can be displayed or hidden by checking or clearing the **Display status bar** choice in the Editor tab of **Tools - Options** command.*

Menu bar

The **Menu bar** includes available pull-down menus. Click a name to open the menu and access the associated commands.

File | Edit | View | Browse | Project | Tools | Window | Help

Here is a short description of these menus:

File	contains project and file commands

*Note: **Recent Files** or **Recent Workspaces** last used can be opened by selecting the filename or the workspace from the File menu.*

Edit	contains editing commands
View	contains commands to customize Studio window
Browse	allows to display to different views of the project
Project	contains commands to update and customize the project
Tools	contains commands to customize tools, toolbars and options
Window	contains commands to arrange multiple windows
Help	contains online help commands

All commands are fully described in Chapter 3.

Note: Menu bar can be displayed or hidden by checking or clearing itself in other toolbar pop-up menu.

Toolbars

Toolbars provide shortcuts for some of the most common commands. Icons are grayed out when they are not active.

Each toolbar is displayed at a default location or at the last location.

Displaying or hiding a toolbar:

A toolbar can be displayed or hidden by checking or clearing in one of the following way:

- its name in the pop-up menu of any toolbar
- the **Hide** option in its own pop-up menu (except for Menu bar)
- its name in the **Toolbars** tab of **Tools - Customize** command (except for Menu bar)
- its name in **View** command (only for Standard toolbar)

Moving a toolbar:

Point to the toolbar title bar and drag it to the desired location.

Customizing a toolbar:

Customize a toolbar by removing unwanted buttons or adding new buttons, menus or commands with the **Tools - Customize** command.

Restoring a default toolbar:

Use the **Reset** button in the **Toolbars** tab of the **Tools - Customize** command.

Note: A custom toolbar cannot be reset.

Project Bar

The **Project Bar** contains three drop-down lists to set up the active project, its configuration and the tool. It also includes four buttons corresponding to commands of the active tool.

*Note: The active project can be set with a right click on a project in the **File View** and choose the **Set as Active Project** option.*

A configuration shows the tool active options. It is defined via the **Settings... - Project menu** command.

Currently, tools available in the Telelogic Logiscope environment are Logiscope *Viewer* and Logiscope *TestChecker*.

- **Build** to generate and build the project. If the project has already been built, this will only rebuild what has changed.

- **Stop** to stop the building process

- **Rebuild all** to generate and build the project. This will force a rebuild of the entire project.

- **Start Viewer** to launch **Viewer** (available if the *QualityChecker* and *TestChecker* modules have been activated).

- **Start TestChecker** 🖼 to launch **TestChecker** (available if the *TestChecker* modules have been activated).

*Note: To launch these commands, a project must be loaded in the **File View**.*

Logiscope Bar

Allows to show specific results for the current project. From left to right, symbols allow to:

- display the **Metrics Dictionary**
- display the **Criteria Level**
- display the **Factor Level**
- generate the HTML **Quality Report** and open it
- display the **Rule Violations**
- display the **Rule Violation List**
- generate the HTML **Rule Violation Report** and open it
- display the **Component Coverage**
- generate the HTML **Test Report** and open it

Note: Items are available depending on the module activated for the current project, i.e. QualityChecker, RuleChecker, CodeReducer or TestChecker.

To Shortcut Logiscope bar:

Click right on Logiscope toolbar and choose **To shortcut** command from the pop-up menu. The toolbar appears in the shortcut palette:

To reinstall the commands in the toolbar, click right in the shortcut palette (once the shortcut bar structure is developed) and choose the **As Toolbar** command from the pop-up menu. The toolbar will go back to its initial location.

Standard Bar

The **Standard bar** provides shortcuts for most commonly used commands of File and Edit menus.
From left to right respectively, symbols allow to:

- create a new project or workspace
- open an existing file
- save the active file
- save all files
- cut the selection
- copy the selection
- paste the selection
- delete the selection
- undo
- redo
- edit properties
- print the active file
- help on Studio
- help on specific topics

Note: Items are available depending on the context.

Browse Bar

Provides a navigation tool in the **Browse** tab of the output window. From left to right respectively, symbols allow to:

- select next item
- select previous item

TCL Script Bar

Allows navigation in Logiscope internal data. From left to right respectively symbols allow to:

- execute the **Tcl Script**
- open the **Tcl Script Wizard**

Mini Web Bar

Allows navigation in HTML documents and internal data.

From left to right respectively symbols allow to:

- go back to previous page
- go to next page
- stop the page loading
- refresh the current display
- go to the home page

Note:

> *For all the tabs displayed in the Workspace View and all the bars previsously described, a contextual pop-up menu can be activated to hide, show or customize tabs or bars.*
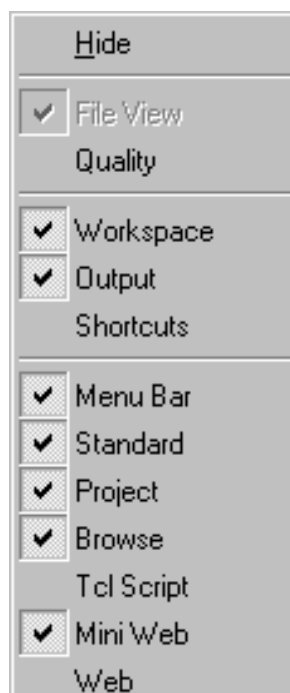
2.2.1 Managing Studio Windows

Several files can be opened at the same time.

The windows display mode can be modified. In the **Window** menu, three Studio operating modes are possible for window management. By default, the windows opened are displayed in cascade.

Modify the display mode by choosing either **Cascade** or **Tile** display:

• select **Cascade** or **Tile** option in the **Window** menu.
   The **Cascade** mode is as follows:
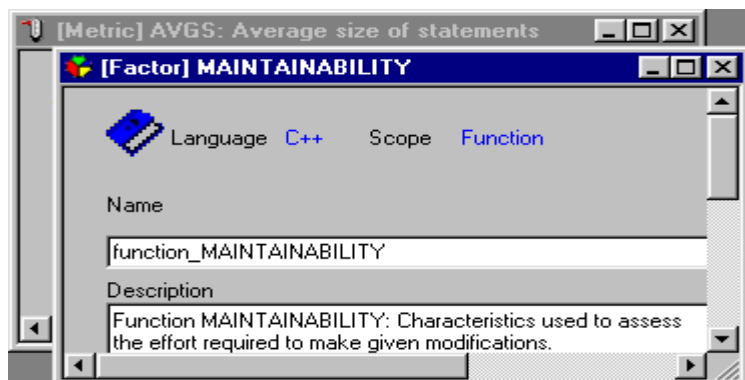
The **Tile Vertically** mode is as follows:
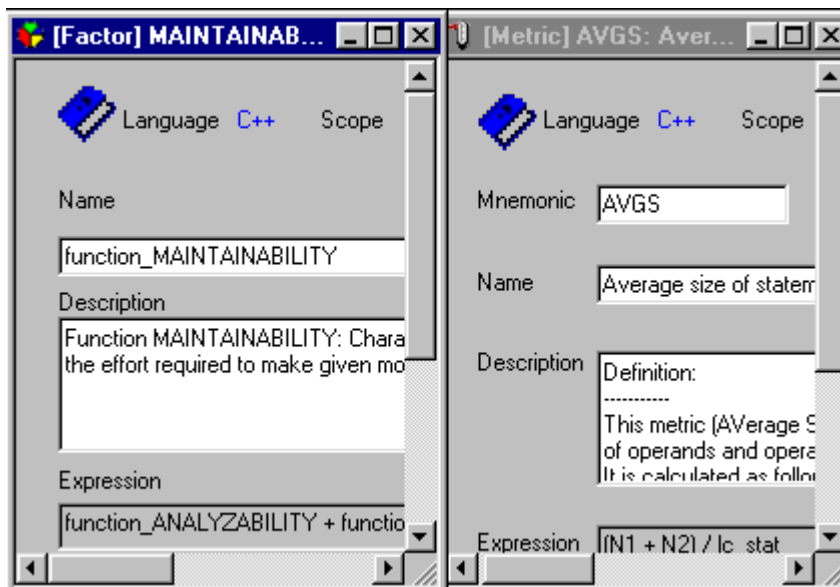
The **Tile Horizontally** mode is as follows:
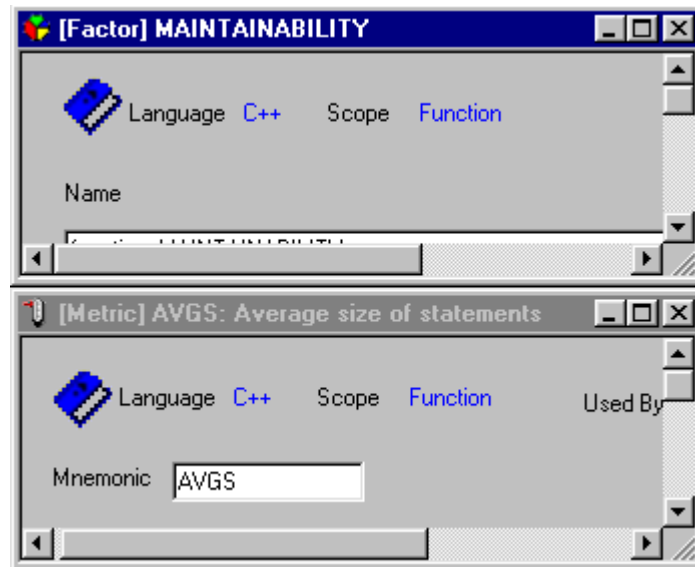
A window can also be duplicated:

• Select the **New Window** command in the **Window** menu.
  The screen now contains two windows, each containing the same file.

• The windows can be closed individually with the **Windows-Close** command or all at once, **Close All** or with the **File - Close** command.

2.3   File View (Tab)

The **File View** displays a workspace, one or more projects grouped in this workspace and all files referenced by each project depending on its type (*QualityChecker, RuleChecker, Codereducer* or *TestChecker*).

For easier visualization, the **File View** shows entities assembled by category in folders. In each category, entities are displayed in alphabetical order.

In the following example the Hangman project (**Hangman.ttp**) is included in the Hangman workspace (**Hangman.ttw)** with a quality model file (**Logiscope.ref)**. Source and header files of the project are also displayed as mentioned above.
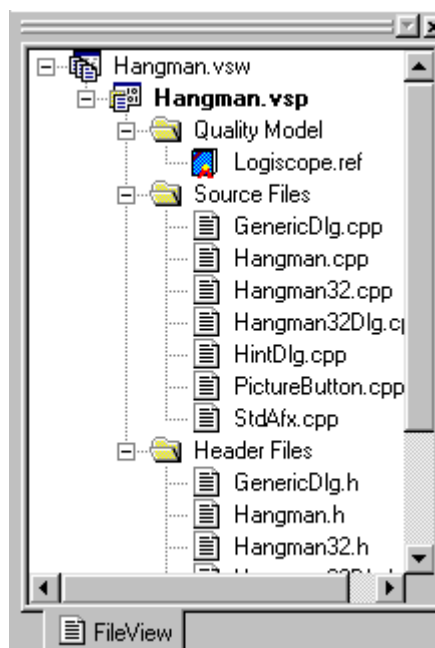
The **File View** content can be expanded (respectively collapsed) into parts:

*   to expand (collapse) an item, click its preceding + (-) symbol or
*   use the arrow keys or
*   double-click on a folder
*   to expand the whole **File View** content, use the star (*) key of the numeric pad.

To access an item, select it directly with the mouse or use the arrow keys to navigate in the view; or type the first letter of its name (the item will be found and automatically selected).

The **File View** main purpose is managing files in a Logiscope project. It allows to:

*   display the Logiscope project file list,
*   edit the workspace and project(s), i.e.:
    *   n display or edit (only for a project) their properties: select a workspace or project and click the **Properties** icon, or right-click and choose the **Properties** option from the pop-up menu, or use the **Properties - View menu** command,

- n add files (see the **Add To Project - Files - Project menu** command),
- n remove files by using **Cut - Edit menu** or **Delete - Edit menu** command,
- n organize the file presentation by using folders: add a new folder by using the **Add To Project - New Folder - Project menu** command; move a file from a folder to another by using the drag and drop mechanism (note that the appearance of the cursor changes depending on whether it points to an authorized or unauthorized location),

- manage the project settings (see the **Settings... - Project menu** command for more details),

- visualize Logiscope file references via the **Properties** dialog (for more details, see the **Properties - View menu** command).

2.4   Quality (Tab)

This function allows to read the quality model definition (metrics, criteria and factors used with their limits) according to Ada, C, C++ or Java languages. It is only available for *QualityChecker* projects.
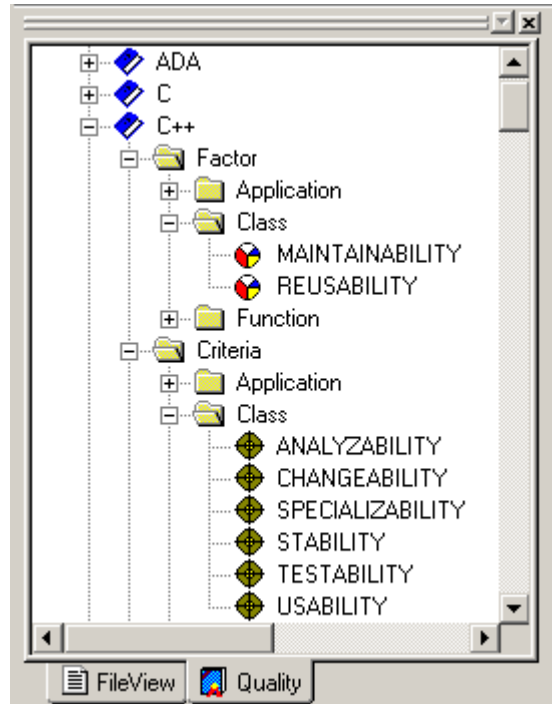
Double click an item (factor, criterion or metric), an html window displays its definition in the result pane.

2.5   Dictionary (Tab)

All available metrics definitions appear for each language (C, C++, Ada and Java). Double click a metric to get its corresponding description in the result pane.
It is always available.

All these metrics can be used to determine calculated metrics or criteria in the Quality Model.
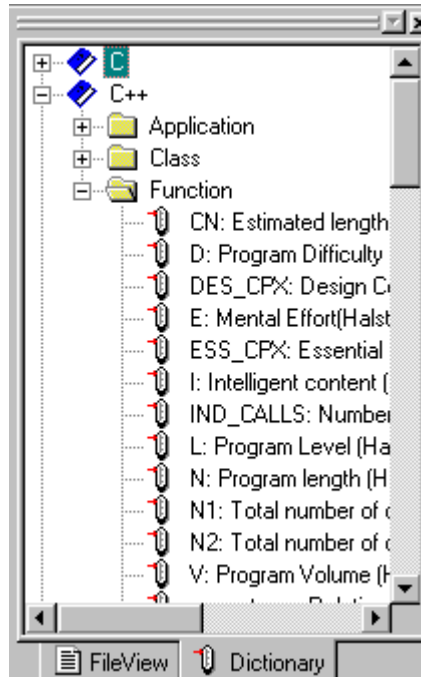
2.6   Criteria (Tab)

The **Criteria** tab displays components sorted by evaluation degree for each criterion. It is only available for *QualityChecker* projects.
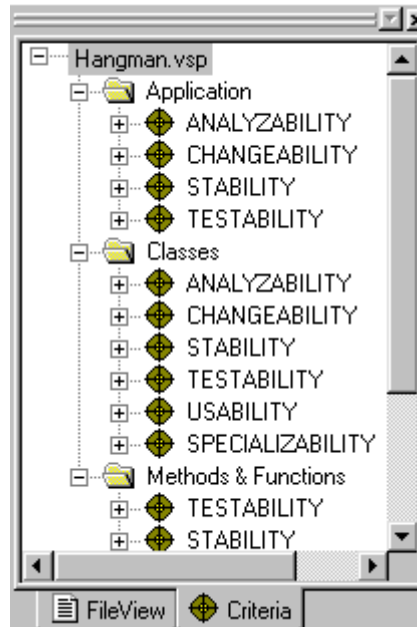
It allows to:

- Expand any criterion to get the analysis results with its evaluation degree ⚖ :
  EXCELLENT, GOOD, FAIR or POOR.

- Expand each evaluation degree to show components, classes or application belonging to that degree.

- Double click a criterion to get the corresponding html definition window with links to criterion composition and metrics definition.

- Double click a component or a class to display its code in the result pane and its metrics value in the output window (**Browse** tab).

- Double click a metric to get its definition in the result pane.

2.7 Factors (Tab)

The **Factors** tab offers the same possibilities as **Criteria**: see paragraph 2.6. It is only available for *QualityChecker* projects.
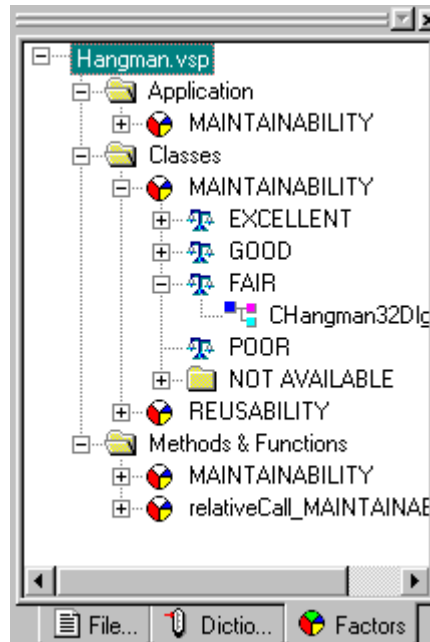
2.8 Metrics Values (Tab)

Displays the computed basic metrics values according to application, classes, methods & functions, depending on the type of the project. It is always available.

A double click on an application, a class or a method/function opens an html table in the result pane with the corresponding basic metric names and values.
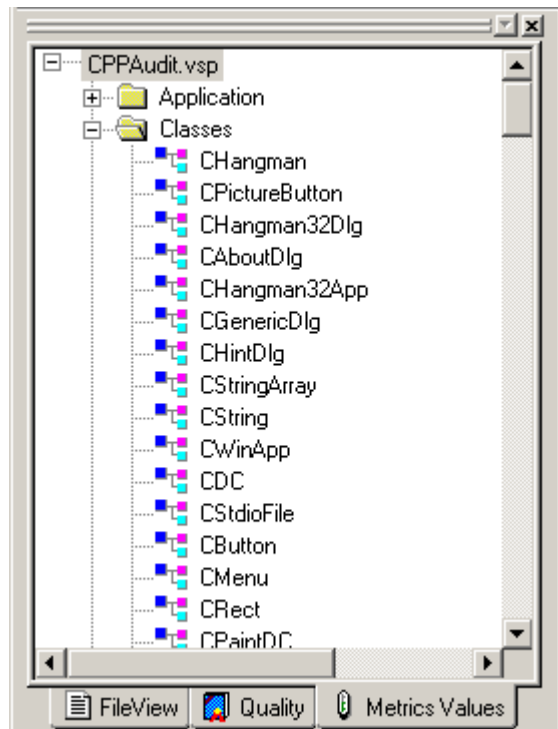
2.9   User Request (Tab)

The **User Request** tab displays for the current project the list of classes or methods/functions for the chosen criterion or factor evaluation degree. It is only available for *QualityChecker* projects.

A double click on any item will display the source file window and shows its metric values in the output window.
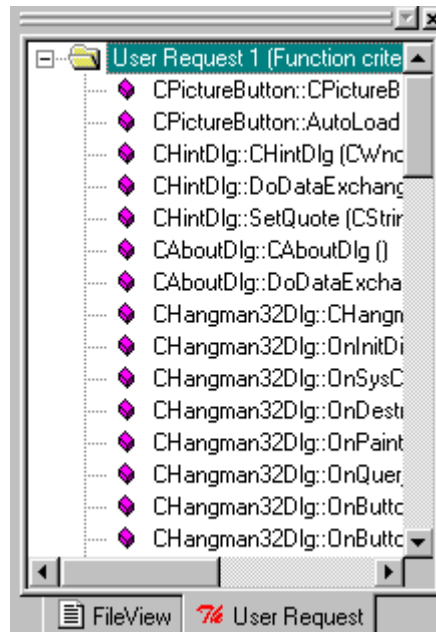
*Note: It is possible to make multiple requests by adding them in the **User Request** tab or by creating a new tab.*

2.10 Violations (Tab)

Displays the violated, clean (not violated) and ignored rules for the current project. A double click on a rule opens its description in html format in the result pane.
It is only available for *RuleChecker* projects.

A double click on a violated line opens the corresponding source file and highlights the violation.
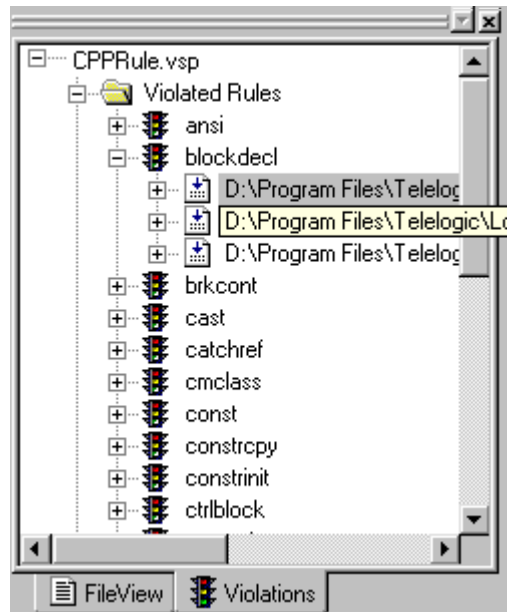
2.11 File Violations (Tab)

It displays the files in the current project where the violations occurred. It is only available for *RuleChecker* projects.

A double click on a rule opens its description in the result pane.

A double click on a violated line opens the corresponding source file and highlights the violation.
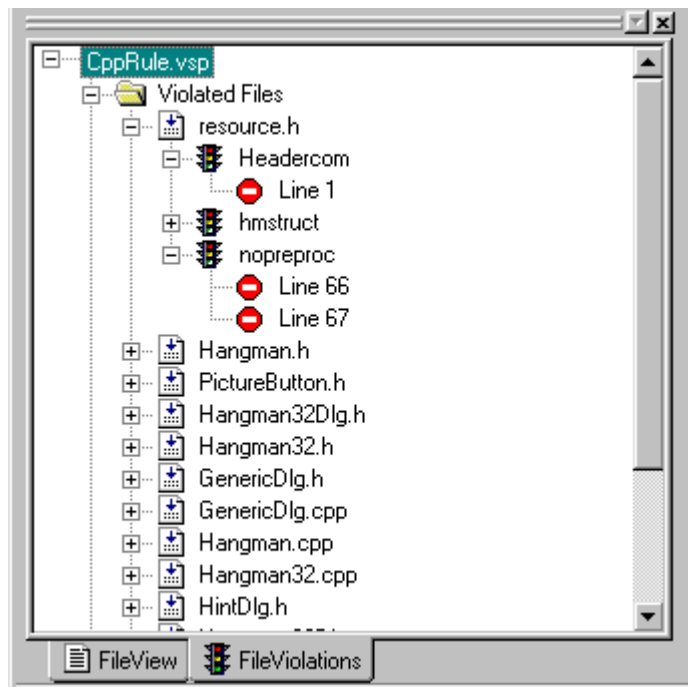
2.12 Components (Tab)

It displays components sorted by source files in the current project. It is only available for *TestChecker* projects.

A double click on a file opens a Component window in the result pane, with DDP and MCDC coverage rate for each component in this file.

A double click on a component opens a Test window in the result pane, with DDP coverage, DDP improvement, MCDC coverage, MCDC improvement, comment and status for each test.
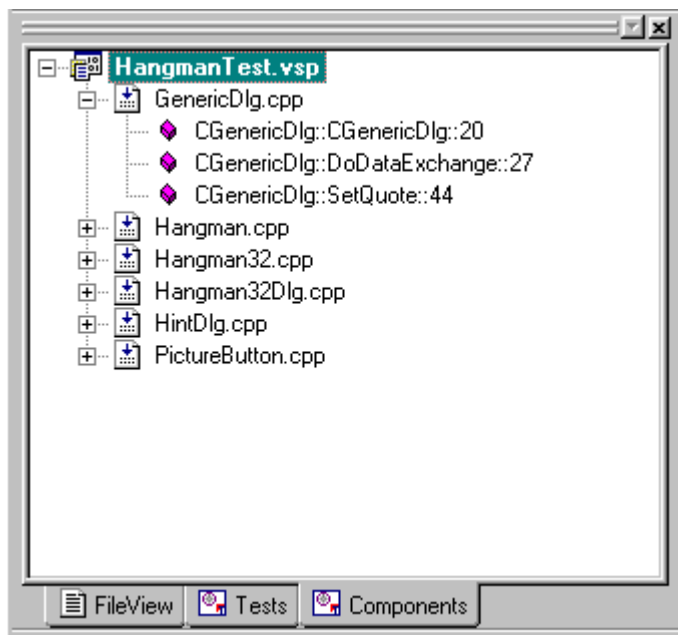
2.13 Tests (Tab)

It displays tests sorted by dynamic files in the current project. It is only available for *TestChecker* projects.

A double click on a dynamic file opens a Test window in the result pane, with DDP coverage, DDP improvement, MCDC coverage, MCDC improvement, comment and status for each test in this file.
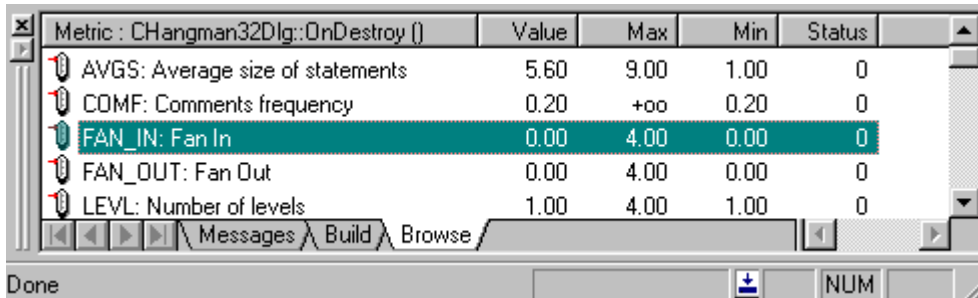
A double click on a test opens a Component window in the result pane, with DDP and MCDC coverage rate for each component.
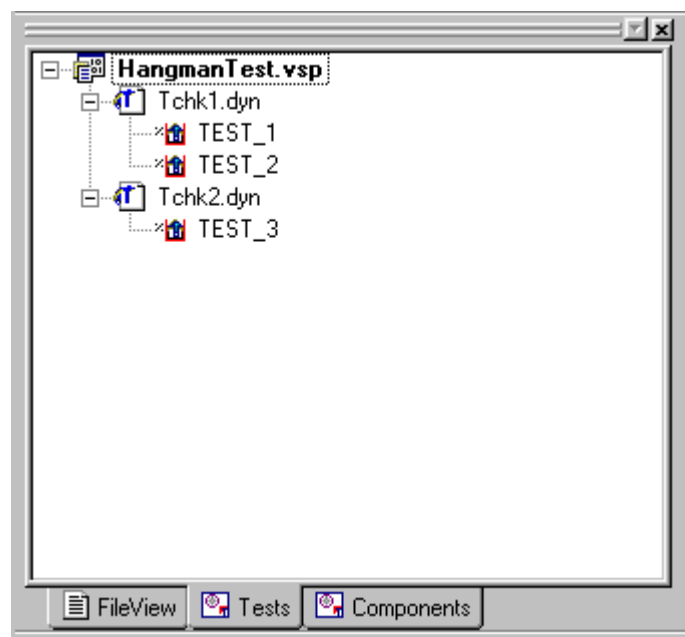
2.14 Browse (Tab)

This tab is an Excel table displaying textual results, that may be saved in an Excel file using the **Save As** pop-up menu. It is used for all types of projects (*QualityChecker*, *RuleChecker*, *CodeReducer* and *TestChecker*).

A double click on a metric or a selection and activating **Locate** pop-up menu opens its description in html format in the result pane.
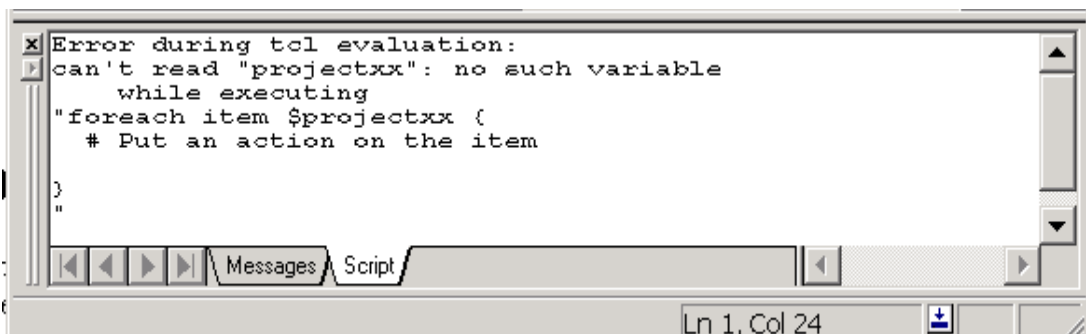
Metric : CHangman32Dlg::OnDestroy ()	Value	Max	Min	Status
AVGS: Average size of statements	5.60	9.00	1.00	0
COMF: Comments frequency	0.20	+oo	0.20	0
FAN_IN: Fan In	0.00	4.00	0.00	0
FAN_OUT: Fan Out	0.00	4.00	0.00	0
LEVL: Number of levels	1.00	4.00	1.00	0

Messages \ Build \ Browse /

Done   NUM

A click on a column header in the **Browse** tab sorts the list in ascending order.

2.15 Script (Tab)

It is used to print the a messages in the output window when the **Script** is executing a Tcl command.

```
Error during tcl evaluation:
can't read "projectxx": no such variable
    while executing
"foreach item $projectxx {
  # Put an action on the item

}
"
```

Messages \ Script /

Ln 1, Col 24

2.16 Kiviat Graph (only on Windows)

The **Kiviat Graph** is a representation of all metrics values. The outer circle symbolizes upper bounds and the inner circle symbolizes lower bounds. It is available for a component at criterion and factor level.

It is only available for *QualityChecker* projects.
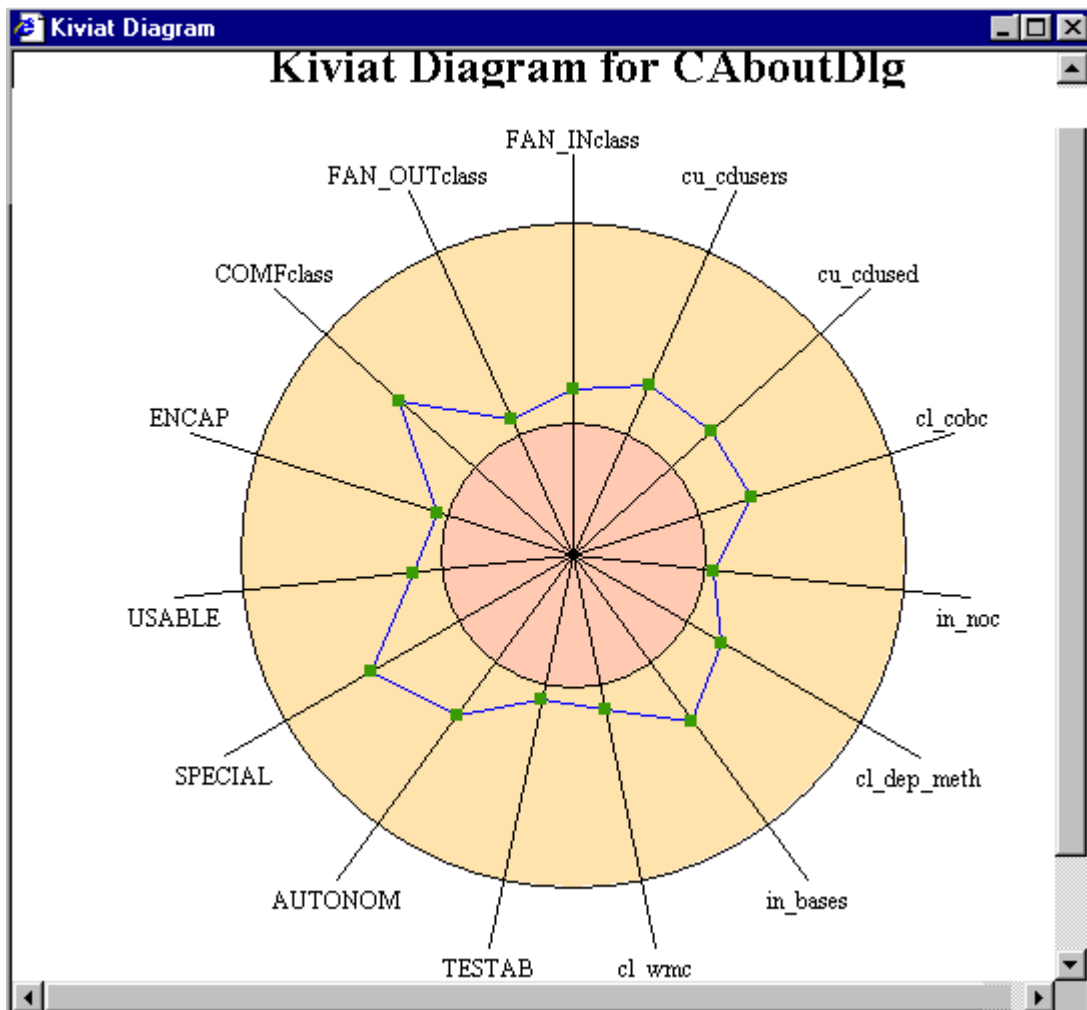
2.17 Script Wizard

The **Script Wizard** can be opened by clicking the script wizard button [icon] of the Tcl script toolbar. It allows to navigate in the UML representation of a Logiscope project and to create Tcl scripts to extract information from the project.
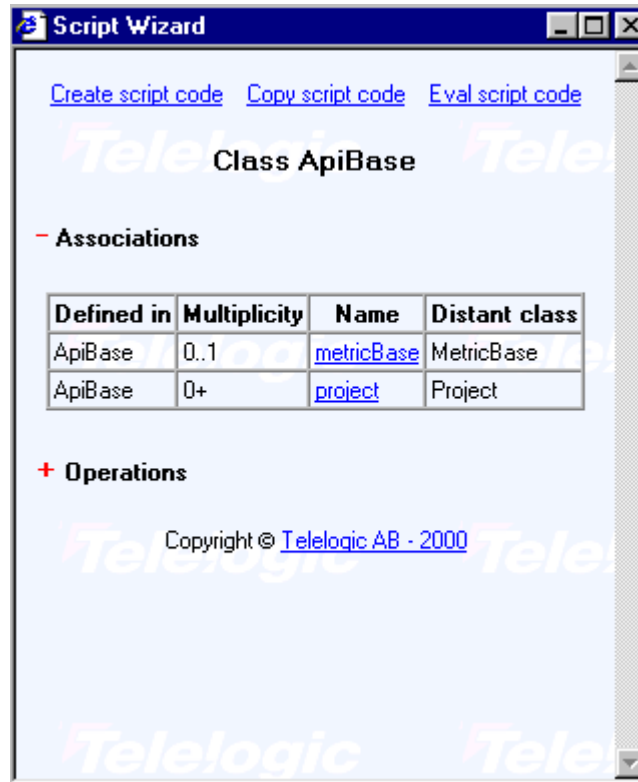
The UML metamodel describes the project structure and Logiscope entities (see the description in Appendix A). The **Mini Web** toolbar allows to navigate in the UML model.

To consult Logiscope data, simply navigate in the UML metamodel and click on **Eval script code** (always displayed at the top of the page): the data are displayed in the **Browse** tab of the output window or in the **Script** tab for textual information.

The **Browse** toolbar [icon] allows to navigate up and down in the **Browse** tab and to open the view associated to each item.

The navigation and the resulting model query can be stored in a Tcl script by using the **Create script code** option (always displayed at the top of the page): a Tcl view displays the script. Execute it by using the **Execute script** button [icon] of the Tcl script toolbar; save it by using the **Save - File menu** or **Save As - File menu** command. To simply copy it, use the **Copy script code** option (always displayed at the top of the page): then paste it in another script or document.

Use Tcl scripts to create another parameterizable view and display it in the **Framework View**. To create such a script, use the **Create script code** option as described previously, then use one of the following functions in it:

BrowserReportInit tabName [iconFile] [-keep] [-cb tclFile]: this command allows to activate a tree tab inside the framework. If needed the tab is created. If iconFile is present, then the icon will be used as the tab image. If keep is present then the contents of the tab (if any) are kept, otherwise the tab is reset. If the -cb tclFile is present, then the associated Tcl script will be evaluated when a double click on a tree object occurs and the OnDoubleClick proc of this Tcl script will be called with double clicked object as parameter.

BrowserReport [-expanded] identObj [parentObj]: this command is used to add a new object in the tree. If -expanded is present and if the object has children, then the node will be expanded, otherwise it will be collapsed. If parentObj is present the node is created as a child of the node corresponding to parentObj.

ReportInit [-check] [-cb tclfile] [column-label column-width column-alignment]+: This command is used to create a new report tab in the output window. If the -check is used, a check box is displayed before each result line. These check boxes allow a more powerful navigation: only the checked lines will be selected when navigating with the F4 key, others are ignored. If the -cb tclFile is present, then the associated Tcl script will be evaluated when a double click on a line object occurs and the OnDoubleClick proc of this Tcl script will be called with double clicked object as parameter. For each column-label column-width column-alignment a new column is created with the corresponding label, width and alignment. An alignment of 0 means left align and an alignment of 1 means right. For sorting, columns right aligned are considered as number and columns left aligned are considered as text.
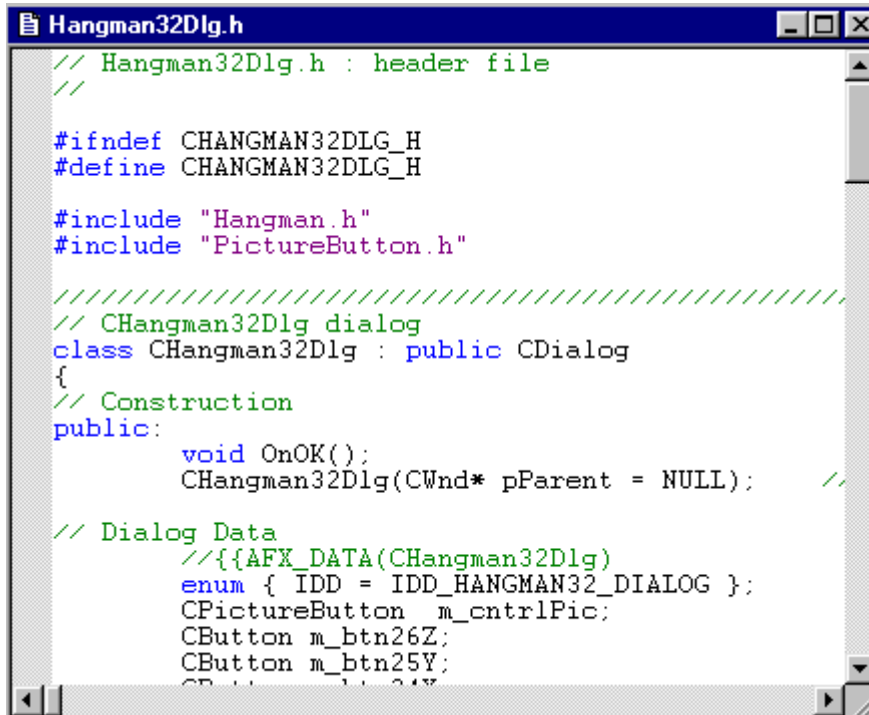
Report sourceObject [label]*: this command is used for adding new lines in a report tab. The sourceObject will be used to fill the first column (icon + text) automatically. The labels, are used to fill the remaining columns.

Output msg: this command allows Tcl script to print a message in the Tcl output tab.

Note: See Appendix A for more details.

2.18 Textual View

Displays the content of a textual file (source code, quality model, rule set, dynamic execution file).

The programming language keywords appear in a different color than the rest of the words in order to make reading easier.

Example: green for comments, blue for keywords, purple for strings and black as the default color.
The line and column for the cursor position are indicated to the right of the **Status bar**

A **Textual View** allows to:

- consult a textual file content,

- edit a textual file content; several functionalities facilitate its edition:
    - type new text or change an existing one,
    - find: to quickly locate character strings (see the **Find - Edit menu** command),
    - replace: to find and replace character strings (see the **Replace - Edit menu** command),
    - marks set: mark lines by using the <Ctrl+F2> key or the **Mark All** option of the **Find** dialog. Once the string occurrences have been marked, jump from one string to another by using the F2 key. To go down, press the F2 key. To go up, use the <shift+F2> key. Use the <Ctrl+F2> key again to delete a marker and use the <Ctrl+Shift+F2> key to suppress all marks.
    - undo/redo as for the others view,
    - cut/copy/paste character strings,

- propagate modifications to the Logiscope project by building it again.

2.19 HTML View

Used to display quality, rule violation and test coverage reports as well as metrics, criterion or factor definitions in the quality model.

Telelogic Logiscope
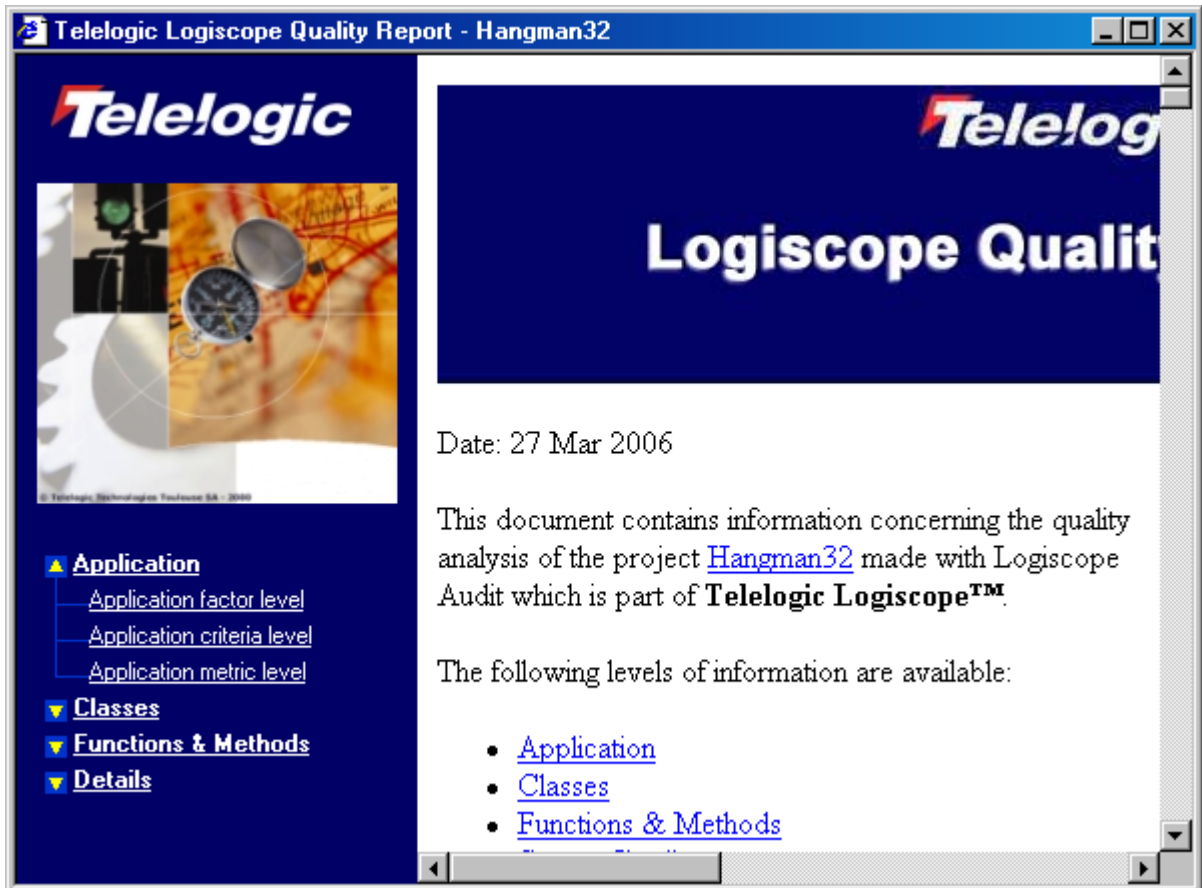
Chapter 3

Studio Commands

About Telelogic Logiscope - Help menu

Display *Studio* and Telelogic Logiscope version information

View

Can be launched from any view.

Context

Always available.

Seek command

- Button palette

Result

A message box containing *Studio* and Telelogic Logiscope version information pops up.

Note: Studio and Logiscope are two separate products, each having its own version number.

Add To Project - Components - Project menu

Not available

Add To Project - Files - Project menu

Add files to the current project

View

Can be launched from the **File View**.

Context

Either a project is active or a project or a project folder is selected.

Seek command

- Pop-up menu: Insert Files

Dialog

An **Open** dialog box pops-up (for more details, see **Open - File menu** command). Select or enter the name of the file(s) to be added.

Result

The new file is added in the project. The **File View** is updated, select the new file and drag and drop it into the required folder.

Save the workspace to take all changes into account.

Possible errors

The following blocking errors may occur because of incorrect entries:

- The added file does not exist,
- The user does not have read access to the file to be added,
- The added file already exists in this project (same name and path),
  In all these cases, the dialog box remains opened even after clicking **OK**.
  The user has to enter another filename or cancel the command.

Add To Project - New Folder - Project menu

Add a folder to the current project

View

Can be launched from the **File View**.

Context

Either a project is active, or a project or a project folder is selected.

Seek command

- Pop-up menu: **New Folder**

Dialog

A **New Folder** dialog box is opened: enter the name of the folder to be added.

Save the workspace to take all changes into account.

Result

A new folder is added to the project and displayed in the **File View**. If a project folder is selected, the new one is created as a sub-folder of the selected one.
Otherwise, the new folder is created at the first level in the project.

As Toolbar - Pop-up menu

Convert the Shortcuts View into a Toolbar

View

Can be launched from the **Shortcuts View**.

Context

Always available.

Result

A pop-up menu appears in the **Shortcuts** when right-clicking. The selection of the **As Toolbar** option allows to display the shortcut list as a toolbar.

To display the toolbar as an embedded listbar in the **Shortcuts View** again, click right and choose the **To Shortcut - Pop-up menu** option.

Back - Button palette

Navigate back in the Script Wizard or in an HTML view

View

Can be launched from the **Script Wizard** or an **HTML** view.

Context

At least one page has been previously displayed.

Seek command

- Button palette ⬅

Result

This command allows navigating back in the **Script Wizard** or in an **HTML** view: it displays the previous page.

Basic Metrics Values - Browse menu

Display the computed basic metrics values for the current project

View

Can be launched from any view.

Context

Always available when an *QualityChecker* project is loaded but significant only for a built project.

Result

This command displays the **Metrics Values** tab in the **Workspace View**. It gives the computed basic metrics values for the current project (depending on the project type). A double click on an application, a class or a method/function opens an html table in the result pane with the corresponding basic metrics names and values.

Note: For unbuilt projects, metrics values are not significant.

Browse - Pop-up menu

Hide or display the Browse bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Build - Project menu

Generate results (output and files) for the active project

View

Can be launched from the **File View**.

Context

Always available.

Seek command

- Shortcut: <F7>
- Button palette

Result

This command generates results (output and files) for the active project. If the project has already been built, this will only rebuild what has changed. The options used to generate are those of the active configuration (see the **Settings... - Project menu** command for more details). The output files are generated in the project directory.

Note: consult Telelogic Logiscope RuleChecker & QualityChecker Getting Started and Telelogic Logiscope TestChecker Getting Started for more information on the generation.

Cascade - Window menu

Arrange windows as overlapping tiles

View

Can be launched from any view.

Context

Always available.

Result

When the windows are in cascade, they appear in a stack. The window on the bottom of the stack occupies the upper left corner of the screen in the result pane, and the remaining windows are offset vertically and horizontally so that the left border and title bar of each window is visible.

Clear - Pop-up menu

Clear the active tab in the output window

View

Can be launched from the **Message** or **Build** tab in the **Output window**.

Context

Available if the active tab is not already cleared.

Result

The active tab content is cleared.

Close - File menu

Close the current window

View

Can be launched from all views except those in the **Workspace View**.

Context

Always available.

Dialog

If a file has been modified and not saved before closing the view, a dialog box is displayed and lets save it. Choose '**Yes**' to save and close, '**No**' to close, '**Cancel**' to cancel the command.

Result

The current view closes.

Close All - Window menu

Close all opened windows

View

Can be launched from all views except those in the **Workspace View**.

Context

Always available.

Dialog

If a file has been modified and not saved before closing the view, a dialog box is displayed and lets save it. Choose '**Yes**' to save and close, '**No**' to close, '**Cancel**' to stop the command.

Result

The views close.

Close Tab - Pop-up menu

Close a parameterizable view

View

Can be launched from any parameterizable view.

Context

Always available.

Result

The current parameterizable view is closed and disappears from the **Workspace View**.

Close - Window menu

Close the current window

View

Can be launched from all views except those in the **Workspace View**.

Context

Always available.

Dialog

If a file has been modified and not saved before closing the view, a dialog box is displayed and lets save it. Choose '**Yes**' to save and close, '**No**' to close, '**Cancel**' to cancel the command.

Result

The current window closes.

Close Workspace - File menu

Close the current workspace

View

Can be launched from all views.

Context

A workspace must be opened.

Dialog

If modifications have been made in the workspace or in files, a dialog box is opened and prompts to save the changes or not:

- Click '**Yes**' to save the changes and continue the close command,
- Click '**No**' to continue without saving,
- Click '**Cancel'** to stop the close command.

Result

The workspace is closed, with or without save action.

Possible errors

The possible errors are those linked to saving (see the **Save - File menu** command).

Components - Pop-up menu

Hide or display the Components view

View

Can be launched from a tab in the **Workspace View**.

Context

Always available when a *Test* project is loaded.

Result

With a right click on the tab, a pop-up menu appears. The view can be hidden or displayed depending on its current state.

Configurations - Project menu

Add or remove code generation configurations and designates the active project and configuration

View

Can be launched from any view.

Context

A project is opened.

Dialog

A **Configuration** dialog box is opened. It contains the following buttons:

- **Close** to close the dialog box.

- **Add** to create a new configuration based on an existing one: an **Add Configuration** dialog opens. Enter the new configuration name in the **New name** field, choose an existing configuration to initialize the new one in the **Copy from** list, select the project to attach the new configuration in the **Project** list. Click **OK** to confirm, or **Cancel** to cancel the command.

- **Remove** to delete a configuration: select a configuration then click **Remove**.

- **Set Active** to designate the active project and configuration.

Result

The modifications are applied to the projects. The project bar is updated (displays the active project and configuration).

Copy - Edit menu

Copy entities into the paste buffer

Context

Requires selection of the entities to be copied.

Seek command

- Shortcut: <Ctrl+C>
- Pop-up menu
- Button palette

Result

Internal changes

The selected entity (text) is copied into the paste buffer.

Changes to the views

The result of the operation is transparent to the user. The effect will only be visible after activation of the **Paste** command.

Copy All - Pop-up menu

Copy all the entities into the paste buffer

Result

Internal changes

The entities (text) are copied into the paste buffer.

Changes to the views

The result of the operation is transparent to the user. The effect will only be visible after activation of the **Paste** command.

Customize - Tools menu

Customize toolbars or the Tools menu

View

Can be launched from all views.

Context

Always available.

Seek command

- Pop-up menu on a toolbar

Dialog

The customized dialog box contains three tabs:

- a **Command** tab to add a button to a toolbar: in the **Categories** box, click the category that contains the command or the item to be added. Drag the button from the **Buttons** area to the toolbar.

- a **Toolbars** tab to:
    - Display or hide toolbars: click the toolbar names to be displayed. Clear the check boxes for any toolbar to be hidden. Each toolbar appears in a default location or in the last location it was moved to.

Note: the Menu bar cannot be hidden this way.

- Show the ToolTips: check **Show ToolTips**. ToolTips are on-screen descriptions of toolbar buttons that appear when the mouse points over a button.
- Modify the toolbar buttons appearance: check **Cool Look**.
- Enlarge the toolbar buttons: check **Large Buttons**.
- Create a new toolbar: click **New** and type a new toolbar name. Move the **Customize** dialog box aside to see the new toolbar, then use the **Command** tab to add buttons to the new toolbar.
- Restore a default toolbar: in the Toolbar box, click the toolbar name to restore to its default state, then click **Reset**. A custom toolbar cannot be reset.
- Delete a custom toolbar: in the Toolbar box, select the name of the custom toolbar to delete and click **Delete**. A default toolbar cannot be deleted.
- Rename a custom toolbar: in the Toolbar box, click the toolbar to rename. In the Toolbar name area, type the new name. Press **Enter**. A default toolbar cannot be renamed.

- a **Windows Layouts** tab to define named configuration (windows position).
  - n To define a configuration, click the New icon and give a new name.
  - n To restore a configuration, select its name in the list and click the Restore button.
  - n To suppress a configuration, select its name and click the Suppress icon .

- a **Tools** tab to add commands in the Tools menu. These commands can be associated with any program that will run on the operating system. Also specify arguments for any command to be added to the **Tools** menu:
  - n To add a command to the **Tools** menu: click the **New** icon or press **Insert key**. A blank line (indicated by an empty rectangle) appears in the **Menu Contents** box: type the name of the tool as it will appear on the **Tools** menu, and press **Enter**.
    For example, to add a command to Windows Notepad accessory, type **Notepad**. The remaining steps will be to specify what action will occur when the new Notepad command is selected.
    In the **Command** box browse or type the program pathname, for example C:\WINDOWS\NOTEPAD.EXE.
    In the **Arguments** text box, browse or type any arguments to be passed to the program. Use the drop-down arrow next to the **Arguments** text box to display a menu of arguments. Select an argument from the list to insert argument syntax into the **Arguments** text box.
    In the **Initial Directory** box, browse or type the file directory where the command is located.
    Click **OK** to validate. The command appears in the Tools menu: select it from the menu to run it.
  - n To edit a command of the **Tools** menu: in the **Menu Contents** box, select the command to edit. Perform one or more of the following actions:
    - p To move the command up one position in the menu, choose the **Move Up** button.
    - p To move the command down one position in the menu, choose the **Move Down** button.
    - p To change the menu text, command line (tool path and file name), command-line arguments, or the initial directory, type the new information in the appropriate text box.
    - p To specify a letter in the menu title as an access key, precede that letter in the **Menu Contents** box with an ampersand (&). By default, the first letter in the title is the keyboard access key.
    - p To specify sub-menu items, use a \.
    - p To specify that a console application output is displayed in the output window rather than in the console window, click **Use Output Window**.
    - p To be prompted for command-line arguments each time the tool is run, select **Prompt for Arguments**.
  - n To remove a command from the **Tools** menu: in the **Menu Contents** box, select the command to be deleted and press the **Delete** button.

- an **Add-ins** tab:
  - n To create an add-in module by clicking the Create button and filling the different fields in the Attributes, Binaries, Scripts or Other files tab. Once the add-in is created a new directory appears in the *<log_install_dir>***addins** directory.
  - n To activate or desactivate an add-in module by clicking on the check box and closing the dialog window (the activation/desactivation occurs once the window is closed).
  - n To modify an add-in module by clicking the Modify button and filling the different fields displayed in the window.

Result

The toolbars and/or the **Tools** menu are customized according to choices made.

Cut - Edit menu

Cut entities and transfer them to the paste buffer

View

Can be launched from any **Textual View**.

Context

The entities to be cut must be selected.

Seek command

- Shortcut: <Ctrl+X>
- Pop-up menu
- Button palette

Result

This command transfers the selected entity (text) to the paste buffer and removes it from the view (for more details, see the Result paragraph of the **Delete - Edit menu** command).

Delete - Edit menu

Delete entities

View

Can be launched from any **Textual View**.

Context

The entities to be deleted must be selected.

Seek command

- Shortcut: <Del>

- Button palette ✕

Result

This command deletes the selected entity (text).

Display Incoming Links - Link menu

Display the incoming links for an object

View

Can be launched from a file or a folder.

Context

Always available when the object has some links.

Seek command

- Pop-up menu on the object

Result

The incoming links are displayed in the Link tab of the Output window.

Display Outgoing Links - Link menu

Display the outcoming links for an object

View

Can be launched from a file or a folder.

Context

Always available when the object has some links.

Seek command

- Pop-up menu on the object

Result

The outcoming links are displayed in the Link tab of the Output window.

Edit Links - Link menu

Display the links or hyperlinks for an object

View

Can be launched from a file or a folder.

Context

Always available (even if the object hasn't any links).

Seek command

- Pop-up menu on the object
- Button palette

Result

The outgoing and incoming links and hyperlinks are displayed in a window. They can be selected and then set as default or deleted, but they cannot be changed this way.

Execute Script - Tools menu

Execute a Tcl script

View

Can be launched from a **Tcl view**.

Context

Always available.

Seek command

- Shortcut: <Ctrl+F5>
- Button palette

Result

This command executes the current Tcl script. The result can be displayed in the output window or in a parameterizable view, depending on the script.

Possible errors

If the Tcl script is syntactically incorrect, an error message is displayed in the script tab of the output window.

Exit - File menu

End a session

View

Can be launched from any view.

Context

Always available

Dialog

If modifications have been made in the workspace or in files, a dialog box is opened and prompts to save the changes or not:

- Click '**Yes**' to save the project and continue the exit command,
- Click '**No**' to continue without saving,
- Click '**Cancel**' to stop the exit command.

Result

All windows are closed and the *Studio* session is finished, with or without save action.

Possible errors

The possible errors are those linked to saving (see the **Save - File menu** command).

Export - File menu

Export *CodeReducer, QualityChecker* or *RuleChecker* result data in CSV format files

View

Can be launched from any view.

Context

Only available if the project loaded has been built.

Dialog

Activate the command **File - Export - CSV format** -

- **CodeReducer Data**: to export the *CodeReducer* results: location of all similarities found.

- **QualityChecker Data**: to export the ***RuleChecker*** results: values of metrics, rating at criteria and factor level for each component analysed.

- **RuleChecker Data**: to export the *RuleChceker* results: location of all violations found.

Result

CSV files containing the results data are generated in the report.dir folder in the Logiscope repository of the project loaded.

Possible errors

Lack of space on the disk.

FAQs - Help menu

Display Telelogic Logiscope FAQs (Frequently Asked Questions)

View

Can be launched from any view.

Context

Always available.

Result

An html file containing FAQs on Logiscope is displayed with links to online Telelogic support.

File View - Pop-up menu

Hide or display the File View

View

Can be launched from a tab in the **Workspace View**.

Context

Always available when a project is loaded.

Result

With a right click on the tab, a pop-up menu appears. The view can be hidden or displayed depending on its current state.

Find - Edit menu

Find a character string

View

Can be launched from a **Textual View**.

Context

Always available.

Seek command

- Shortcut: <Ctrl+F>

Dialog

A dialog box is opened. It contains the following fields:

- **Find what** enter the character string to be searched; a scrolling list containing the previous searched strings also lets choose one of them without re-entering it. If a string was selected in the text before starting the command, it appears in this field.

- **Match whole word only**: select it to search isolated occurrences of the string only.

- **Match case**: select it to discriminate between uppercase and lowercase characters.

- **Direction**: select **Up** or **Down** to indicate the direction of the search.

The dialog box contains the following buttons:

- **Find Next** to find the next occurrence of the string and close the dialog box. Once the dialog is closed, use the F3 key to continue the search downwards or press <Shift+F3> for searching upwards.

- **Mark All** to mark all string occurrences in the text. Once the string occurrences have been marked, jump from one string to another with the F2 key. Go down with the F2 key. Go up with the <Shift+F2> key. The <Ctrl+F2> key allows adding or deleting a marker and use the <Ctrl+Shift+F2> key to suppress all marks.

- **Cancel** to cancel the command.

Result

When an occurrence of the searched string is found, it is highlighted.

Possible errors

The searched string is not found. An error message box is opened: click '**OK**' to come back to the **Find** dialog box.

Full Screen - View menu

Allow to put the current view into full screen

Context

Always available.

Result

The current view is put into full screen, click on the Close FullScreen icon to go back.

Help Topics - Help menu

Give access to online help

View

Can be launched from any view.

Context

Always available.

Seek command

- Button palette

Result

This command allows consulting the help on line through a list of help chapters, an index or a research mechanism.

Hide - Pop-up menu

Hide a toolbar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The selection of the **Hide** option makes it disappear.

To display the hidden toolbar check its name either in the **Toolbars** tab of the **Customize - Tools menu** command or in the pop-up menu of another toolbar.

Home - Button palette

Navigate to a home page (HTML)

View

Can be launched from the **Script Wizard** or an **HTML** view.

Context

Always available.

Seek command

- Button palette 🏠

Result

This command displays the home page. In the **Script Wizard**, the home page contains the two entry points of the UML metamodel.

Insert Project into Workspace - Project menu

Add projects in the current workspace

View

Can be launched from the **File View**.

Context

Always available when a workspace is loaded.

Seek command

- Pop-up menu

Dialog

An **Open** dialog is opened (for more details, see the **Open - File menu** command). Select or enter the name of the project(s) to add.

Result

Internal changes

The workspace is updated: the new project is added to the current set of projects.

The entities of the new project are loaded.

Changes to the views

The new project appears in the **File View**.

Possible errors

An error may occur if the added project does not exist.

Kiviat graph - View menu

Display the Kiviat graph for a component

View

Can be launched from **Criteria** or **Factors** view in an *QualityChecker* project.

Context

A component must be selected.

Seek command

- Shortcut: <Ctrl+K>
- Pop-up menu

Result

The Kiviat graph is displayed in the result pane.

Large Icons - Pop-up menu

Increase the size of icons in the Shortcut view

View

Can be launched from the **Shortcut** view.

Context

Always available.

Result

Click right in the shortcut list and a pop-up menu appears. The selection of the **Large Icons** option allows to display large icons.

Link - Pop-up menu

Hide or display the Link bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click ⚭ a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Link Options - Link menu

Display the links and hyperlink options

View

Can be launched from any view.

Context

Always available.

Seek command

- Button palette

Result

The links and hyperlink options window is opened (see **Links** and **Hyperlink** tabs in **Options - Tools menu**).

Locate - Pop-up menu

Locate the active information by opening the corresponding source or html file in the result pane

View

Can be launched from the **Browse** tab in the **Output window**.

Context

Always available.

Seek command

- Click right on an item.
- Pop-up menu

Result

The source file or html file corresponding to the active information is opened in the result pane.

Logiscope - Pop-up menu

Hide or display the Logiscope bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Make Link from Start - Link menu

Allow to define an incoming link from the start object

View

Can be launched from a file or a folder.

Context

Always available when the Start Many Links mode is active.

Seek command

- Pop-up menu on the object
- Button palette
- Shortcut: <Ctrl+L>

Result

The first selected object is defined as the start point (it has incoming/outcoming link from/on itself), all other objects have an incoming link from the start.

Menu Bar - Pop-up menu

Hide or display the Menu bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

Metrics Dictionary - Browse menu

Display the basic metrics for all languages

View

Can be launched from any view.

Context

Always available when an *QualityChecker* project is loaded.

Seek command

*   Button palette

Result

This command displays the **Dictionary** tab in the **Workspace View**. It gives for each language (C++, C, Java and Ada), the basic metrics used for the different scopes: Application, Class (only available for C++ and Java) and Function.
A double click on a metric opens its description in html format in the result pane.

Mini Web - Pop-up menu

Hide or display the Mini Web bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Move - Pop-up menu

Move a toolbar

View

Can be launched from an undocked toolbar.

Context

Always available from the title bar.

Result

With a right click on the title bar of a toolbar, a pop-up menu appears. The selection of the **Move** option allows to change the position of the toolbar.

New – File menu

Create a new project or workspace

View

Can be launched from all views.

Context

Always available.

Seek command

- Shortcut: <Ctrl+N>

- Button palette

Result

Internal changes

The workspace and/or the project are created and loaded (the corresponding files are created at the specified location). The default folders of the new project are initialized.

Changes to the views

- The **File View** is updated.

- If a new workspace is created, all the modified files related to the previous workspace are prompted to save.

Possible errors

- No write access to a specified location.

New Application Criteria - Pop-up menu

Add a new application criterion

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new application criterion is created for the selected language and with the default name *ApplicationCriteria_i*. Then, start the Quality Model Editor to customize this new item (see the corresponding chapter).

New Application Factor - Pop-up menu

Add a new application factor

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new application factor is created for the selected language and with the default name *ApplicationFactor_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Application Metric - Pop-up menu

Add a new application metric

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new application metric is created for the selected language and with the default name *ApplicationMetric_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Class Criteria - Pop-up menu

Add a new class criterion

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new class criterion is created for the selected language and with the default name *ClassCriteria_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Class Factor - Pop-up menu

Add a new class factor

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new class factor is created for the selected language and with the default name *ClassFactor_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Class Metric - Pop-up menu

Add a new class metric

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new class metric is created for the selected language and with the default name *ClassMetric_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Function Criteria - Pop-up menu

Add a new function criterion

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new function criterion is created for the selected language and with the default name *FunctionCriteria_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Function Factor - Pop-up menu

Add a new function factor

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new function factor is created for the selected language and with the default name *FunctionFactor_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Function Metric - Pop-up menu

Add a new function metric

View

Can be launched from the Quality tab of any *QualityChecker* project.

Context

Always available.

Result

A new function metric is created for the selected language and with the default name *FunctionMetric_i*. Next start the quality model editor to customize this new item (see Chapter 4, *Quality Model Editor*).

New Window - Window menu

Duplicate a view

View

Can be launched from any view displayed in the result pane.

Context

Not available for the **File View** (if no other window is opened, the command will not be active).

Result

The current view is duplicated and its title changes (a number is added to differentiate each occurrence of the view). A new window is opened.

Next - Button palette

Navigate forward in the Script Wizard or an HTML view

View

Can be launched from the **Script Wizard** or an **HTML** view.

Context

It is possible to go forward.

Seek command

- Button palette ⇨

Result

This command allows navigating forward in the **Script Wizard** or an **HTML** view: It displays the following page.

Next Item - Button palette

Navigate down in the Browse tab of the Output Window

View

Effective in the **Browse** tab of the **Output window**.

Context

Items must be displayed in the **Browse** tab of the **Output window**.

Seek command

- Shortcut: <F4>

- Button palette ![icon]

Result

This command allows to navigate down in the **Browse** tab: it goes to the item placed after the highlighted item of the displayed list and opens the view associated with this new selection (for example, the metric issued from **Criteria** or **Factors** tab).

If no item has been previously selected, the command goes to the first item of the list; when the last item is reached, it is wrapped up around the list.

A click on a column header in the **Browse** tab sorts the list in ascending order.

Open - File menu

Open an existing file

View

Can be launched from any view.

Context

Always available.

Usage

This command is generally used to open an existing project (.ttp), html, source or Tcl file. To open a workspace, it is better to use the **Open Workspace - File menu** command.

A project can only be edited inside a workspace: if the opened project has no associated workspace, a new one is automatically created inside *Studio*.

If changes have been made in the previous workspace and not saved, a dialog prompts to save it before loading the new project: **Yes** to save and continue the project loading, **No** to continue the command without saving the previous workspace, and **Cancel** to stop the command.

Seek command

- Shortcut: <Ctrl+O>

- Button palette

Result

Internal changes

Opening a project loads the project and its workspace. All the project entities are loaded.

Changes to the views

- Opening a project or workspace: all the modified files relative to the previous workspace are prompted to save. The **File View** is updated.

- Opening a textual file (source or Tcl file): a view is opened in the result pane allowing changes.

- Opening an html file: a view is opened in the result pane to visualize it.

Possible errors

The following blocking errors may occur - due to Open dialog box entries:

• The file does not exist.

• The user does not have read access to the file to be opened.

In all these cases, the dialog box remains opened even after clicking **OK**. The user must enter another name or cancel the command.

Open Workspace - File menu

Open an existing workspace

View

Can be launched from any view.

Context

Always available.

Usage

This command is used to open an existing workspace (.ttw).

If the previous workspace was modified and not saved, a dialog lets save it before loading the new one (click **Yes**, **No** or **Cancel**).

Result

Internal changes

Opening a workspace loads the workspace and its project. All the project entities are loaded.

Changes to the views

All the modified files related to the previous workspace are prompted to save. The **File View** is updated.

Possible errors

See the **Open - File menu** command errors.

Options - Files - Project menu

Allow to choose the most significant option file

View

Can be launched from any view.

Context

Available only when the option files associated to the project exist.

Dialog

An Option Files dialog is displayed and allows using the arrows to select the most significant option file among all option files associated to the project. Click **Ok** to validate or **Cancel** to ignore.

Result

The file at the top list level in the dialog box becomes the most significant for the project.

Options - Save As - Project menu

Allow to save a project option in an option file

View

Can be launched from any view.

Context

Only available when a project is loaded.

Dialog

The **Save As** dialog box opens. Use this file browser to name a file to be saved. If an existing file is selected, a warning is displayed: click 'Yes' to override the existing file, or 'No' to cancel and come back to the **Save As** dialog box.

Then another dialog box prompts to include this new file in the project or not. Click **Yes** or **No**.

Result

A new option file (suffixed by **.tot**) is created and appears in the **File View** project if **Yes** has been selected. The project option (HTML tables or Java applets) is saved in this file. Once added it can be suppressed from the project as any other file. Double-clicking on a such file in the **File View**, activates the options editor which allows to change and save options.
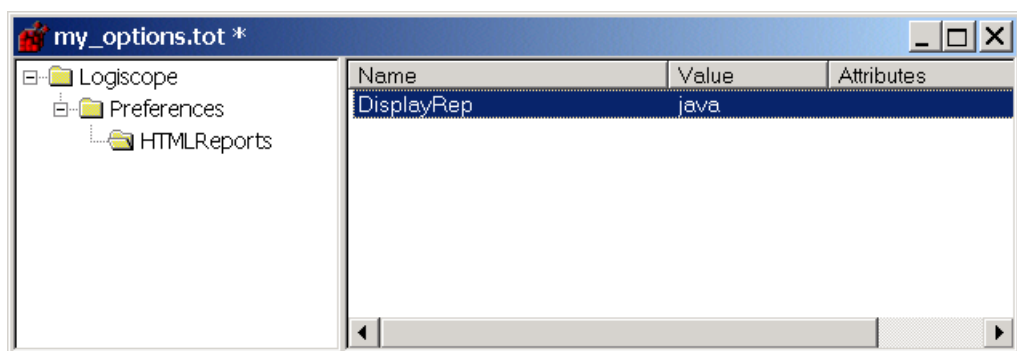
Options - Tools menu

Customize both Studio and HTML reports

View

Can be launched from any view.

Context

Always available.

Result

Customization will appear in the Logiscope session and HTML reports.

Dialog

A dialog box with several tabs is opened.

• The **General** tab allows to customize Studio behaviour regarding display options.
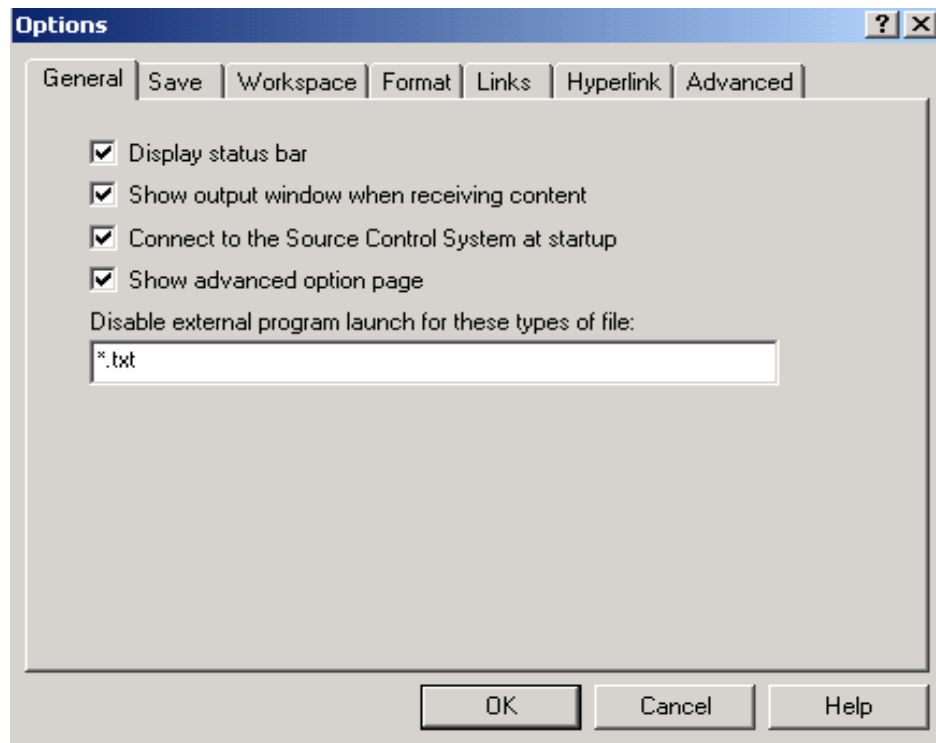
Display status bar: check/clear this option to show/hide the status bar.

Show output window when receiving content: check this option to show the output window when receiving content.

Connect to the Source Code Control System at startup: this option is available only if a source code control (SCC) system is installed. Check it to use the SCC system. If the option value is FALSE, the SCC menu and toolbar are not available.

Show advanced option page: check this option to display an additional property page (Advanced tab, described below) that displays all the options in a tree control.

Disable external program launch for these types of file: list in the field the file extensions (separated by a semi-colon) to open within Studio and not with an external program.

• The **Save** tab allows to customize Studio behaviour regarding save options.
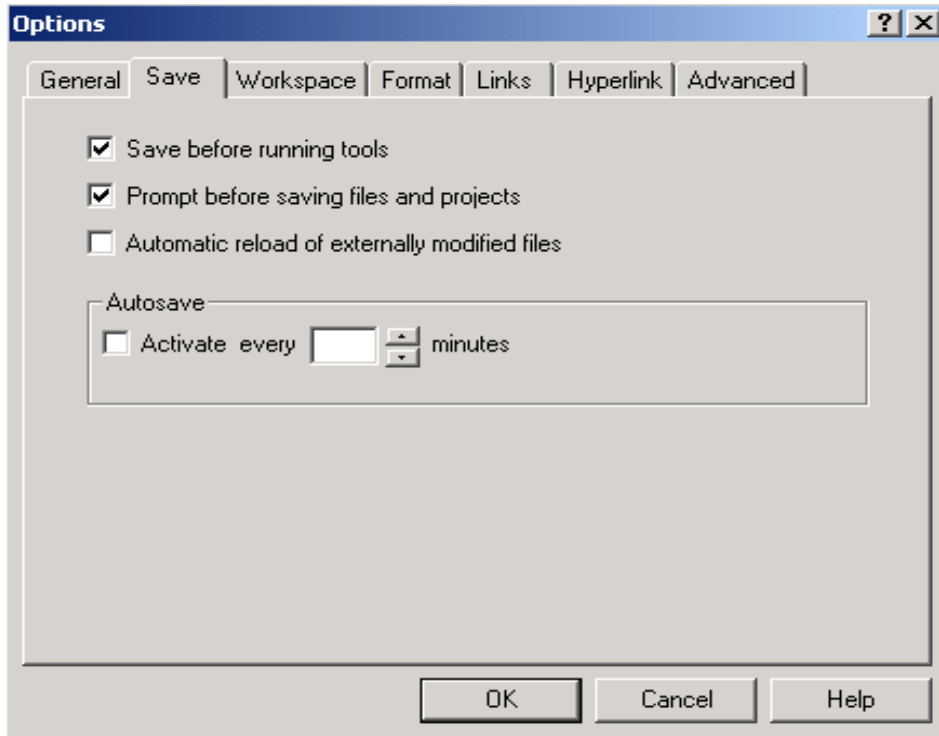
Save before running tools must be checked to automatically save the modifications each time an external tool is launched.

Prompt before saving files and projects allows to ask question before saving files before running tools. This choice is only effective if the previous Save option is also checked, else it has no effect.

Automatic reload of externally modified files when a file included in a project is modified outside Studio, it detects that the file has changed and prompts to reload it. Checked not to be prompted.

Autosave

Activate: if this option is checked, autosave is activated by default every 10 minutes, but this delay can be changed.

Every: allow to increase or decrease the autosave delay.

• The **Workspace** tab allows to customize Studio behaviour regarding workspace options.
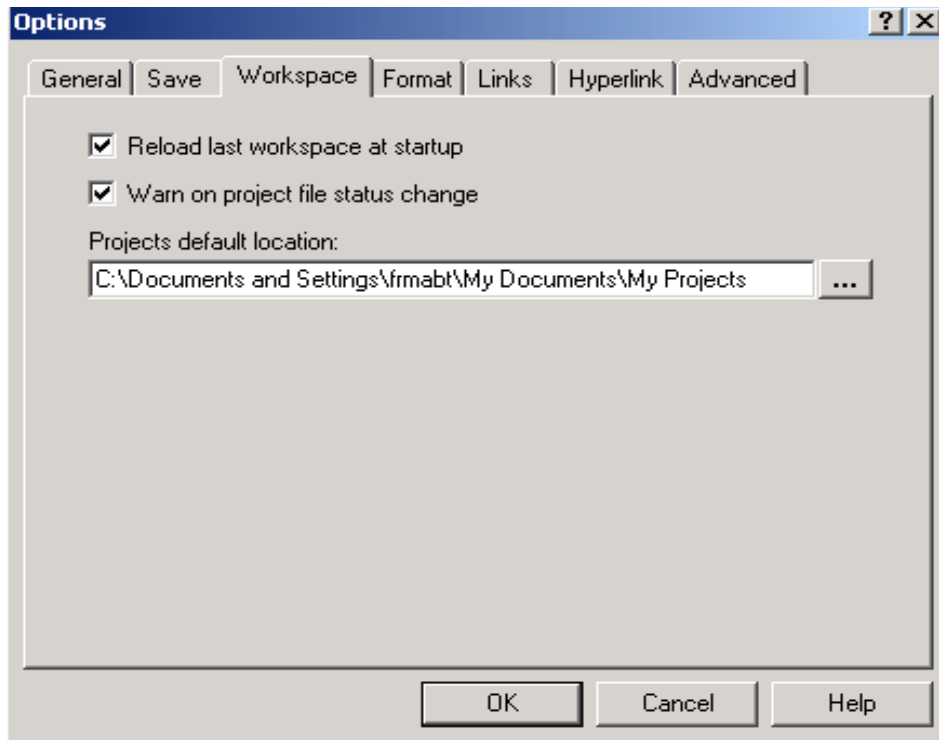
Reload last workspace at startup: if this option is checked, the last workspace is opened and loaded at startup.

Warn on project file status change: check this option to be warned when a project file (***.ttp**) access right passes to ReadOnly in order not to loose the changes.

Projects default location: fill this field with the projects default location path.

- The **Format** tab allows to customize Studio behaviour regarding format options.
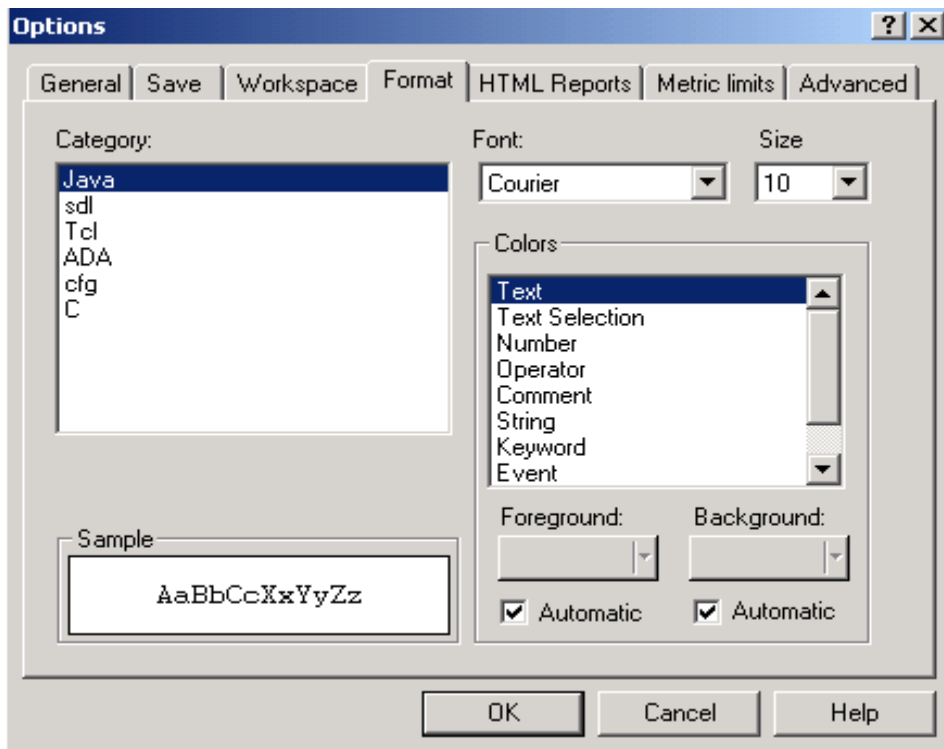
The user can define a font and a size for each type of file and a color to display each type of item in the file.

• The **Links** tab allows to customize Studio behaviour regarding link options.
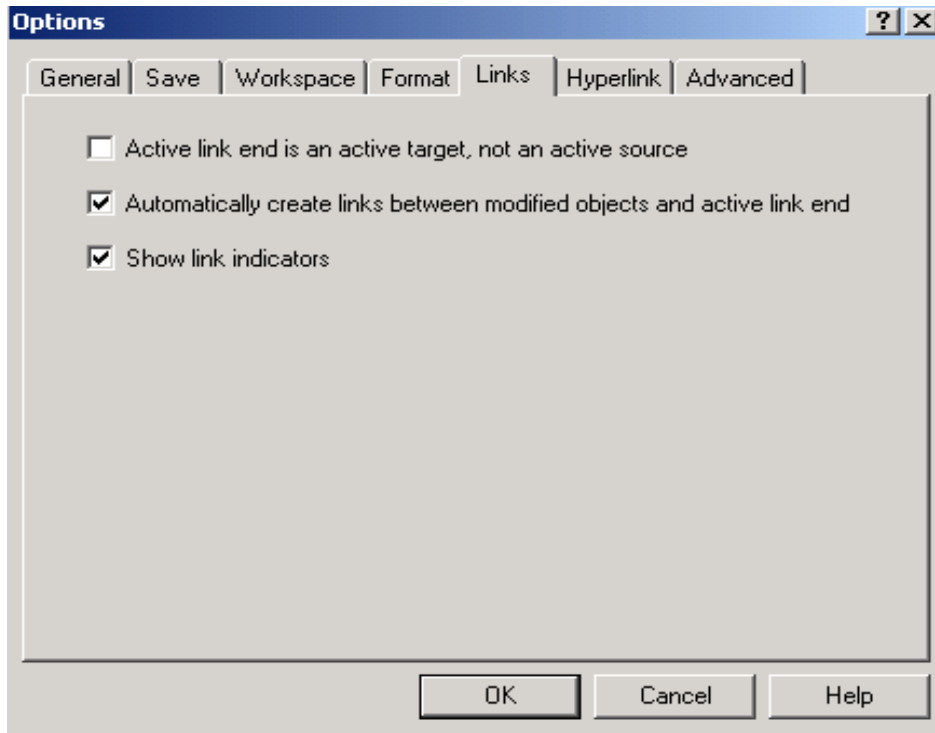
Active link end is an active target, not an active source: check this option to consider active end link as an active target.

Automatically create links between modified objects and active link end: check this option to create links between modified objects and active link end.

Show link indicators: check/clear this option to show/hide the link indicators.

• The **Hyperlink** tab allows to customize Studio behaviour regarding hyperlink options.
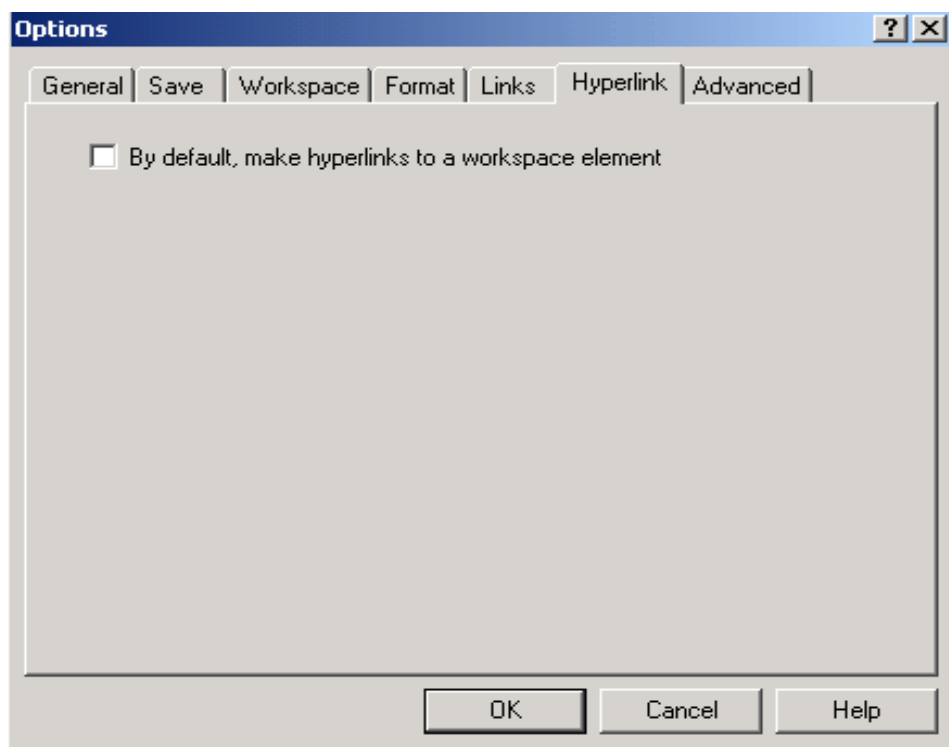
By default, make hyperlinks to a workspace element: check this option to make hyperlinks to a workspace element, by default.

• The **HTML Reports** tab allows a choice between Java Applets (not available on UNIX) or HTML Tables in reports.
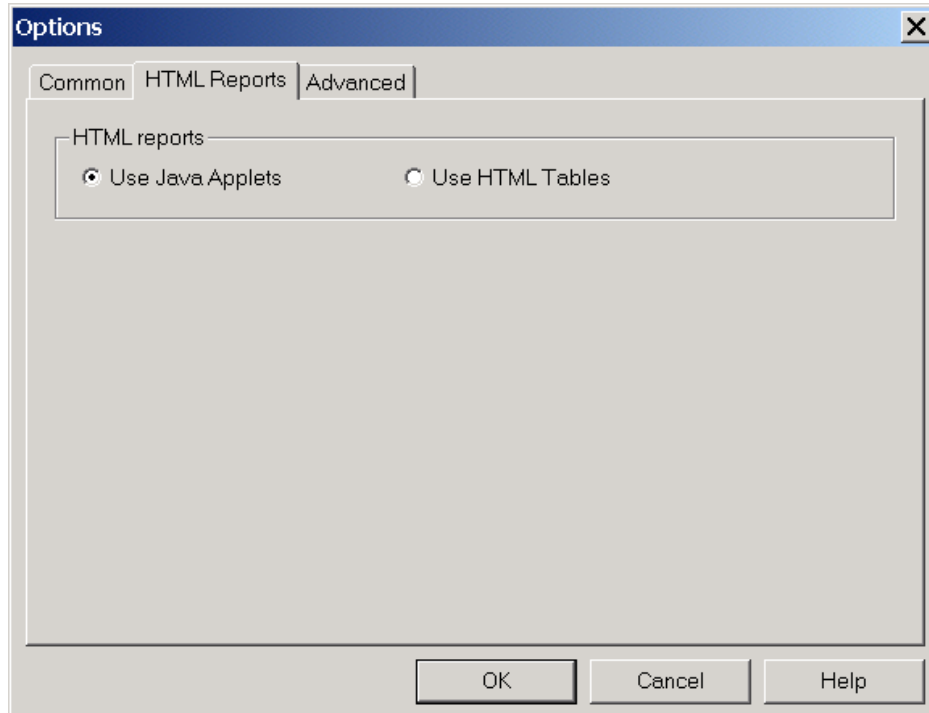
- The **Metric limits** tab allows to give a limit to the PATH metric, this value is used as a bound in computation algorithm to limit the time.
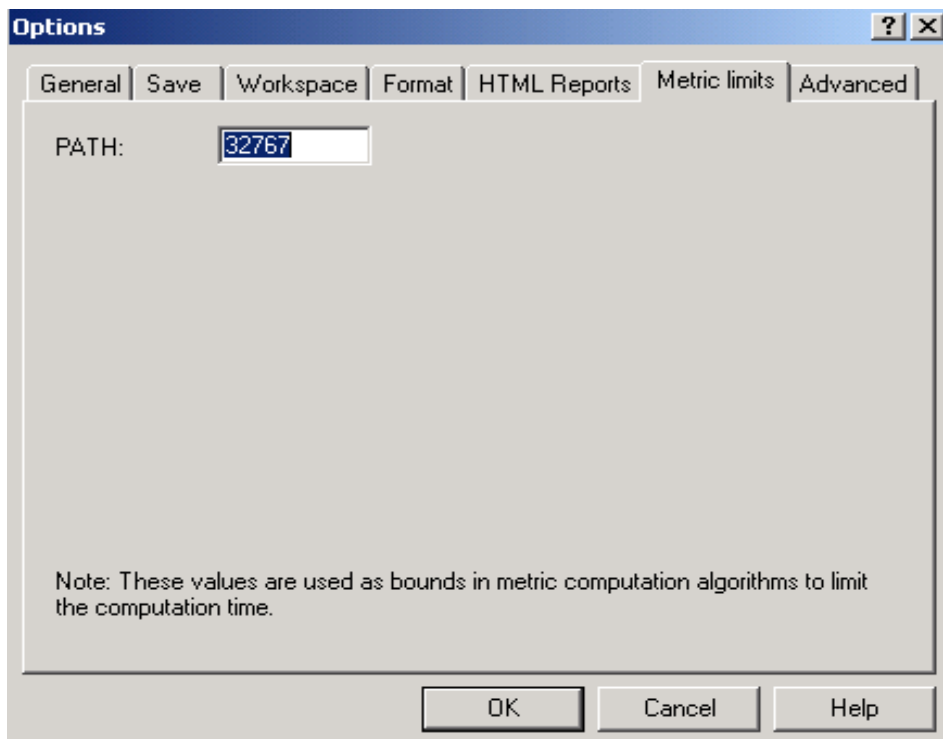
- The **Advanced** tab allows to edit the option values. MetricLimits and HTMLReports items are set using the previous tabs.
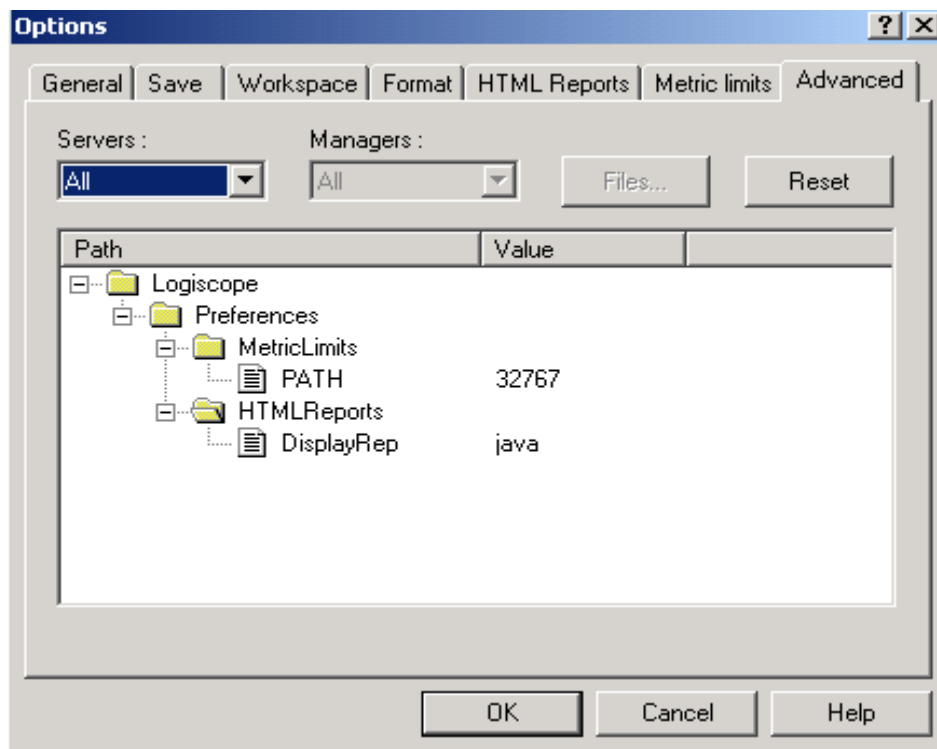
Output - View menu

Display or hide the output window

View

Can be launched from any view.

Context

Always available.

Seek command

- Shortcut: <Alt+2>
- Pop-up menu on a toolbar

Result

The output window (it also includes the **Messages**, **Build** and **Browse** tabs) is displayed or hidden, depending on whether it was hidden or displayed. A check box displayed before the command name in the **View** menu, indicates the current state of the output window: if the box is checked, the output window is currently displayed; if the box is not checked, it is hidden.

When hidden, the output window will be displayed again when a command using it is activated.

Paste - Edit menu

Paste the content of the paste buffer (entities previously copied or cut)

View

Can be launched from any **Textual View**.

Context

The paste buffer must contain entities that can be pasted in the view.

Usage

Click in the target view at a convenient place (or select an entity in the **File View**) before calling the **Paste** command.

Seek command

- Shortcut: <Ctrl+V>
- Pop-up menu
- Button palette

Result

Internal changes

The paste buffer is written in the **Textual View**.

Previous Item - Button palette

Navigate up in the Browse tab of the Output window

View

Effective in the **Browse** tab of the o**ut**put window.

Context

Items must be displayed in the **Browse** tab of the **Output window**.

Seek command

- Shortcut: <Shift+F4>

- Button palette

Result

This command allows to navigate in the **Browse** tab: it selects the item placed before the highlighted item of the displayed list and opens the view associated with it (for example the metric issued from **Criteria** or **Factors** tab).
If no item is currently selected, the command selects the last item of the list; when the first item is reached, it is wrapped up around the list.

A click on a column header in the **Browse** tab sorts the list in ascending order.

Print - File menu

Print or generate a print file of the current view

View

Can be launched from any view, except views of the **Workspace View**.

Context

Always available.

Seek command

*   Shortcut: <Ctrl+P>

*   Button palette 🖨

Dialog

This command displays a **Print** dialog box that allows to:

*   select a printer and modify its properties,
*   choose to print to a file (**.prn**); in this case a **Print to File** dialog box opens and lets select the target file.
*   choose to print the whole document or only some pages,
*   specify the number of printed copies.

Click **Cancel** to stop the command or **OK** to validate.

Result

The content of the current view is printed.

Possible errors

Insufficient access rights to the selected file.

Print Manager - File menu

Not applicable to Logiscope

Print Preview - File menu

Display the current view before printing

View

Can be launched from any **Textual View**.

Context

Always available.

Result

A print preview is displayed in a separate window. Controls are provided to zoom and scroll through the pages.

Print Setup - File menu

Set the printer options

View

Can be launched from any view.

Context

Always available.

Result

The current printer name and options are displayed and can be modified.

Project - Pop-up menu

Hide or display the Project bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Properties - View menu

Display or modify an entity characteristics

View

Can be launched from the **File View** and from any html file.

Context

Requires the selection of the entity whose characteristics are to be displayed or modified.

Seek command

- Shortcut: <Alt+Enter>
- Pop-up menu
- Button palette

Dialog

A **Properties Box** contains a tab for editing the entity characteristics (**General**). Once modifications are entered, close the dialog box or press **Return** to validate or **Esc** to cancel.

Properties of a workspace (File View):
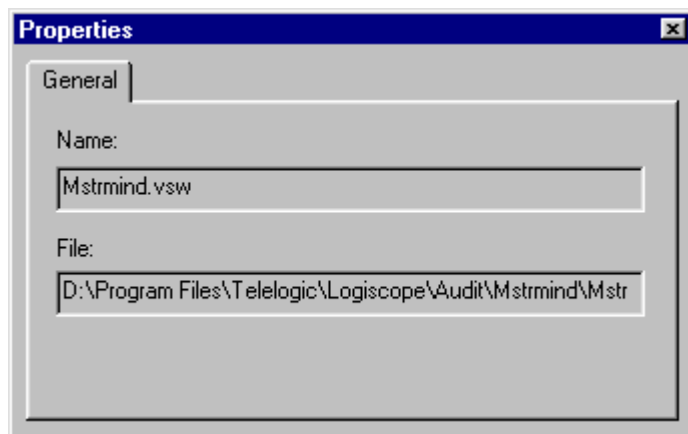
No modification allowed.

Properties of a project (File View):
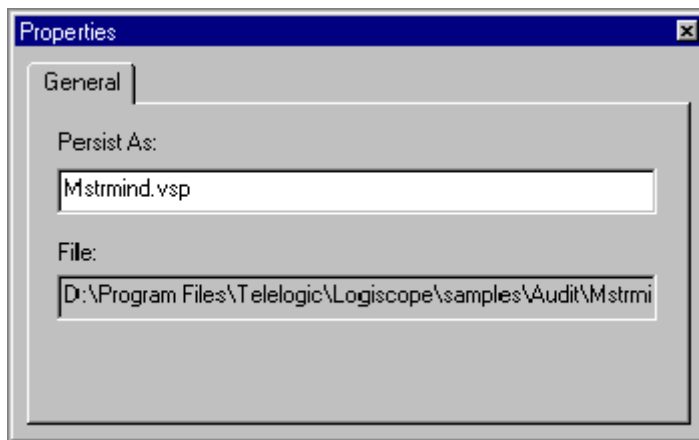
No modification is allowed.

Properties of a folder (File View):
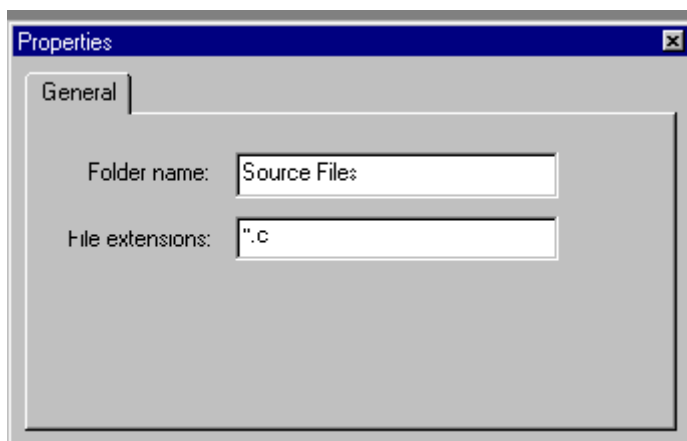
Once modifications are entered, close the dialog box, press **Return** to validate or **Esc** to cancel.

Properties of a textual file (File View):
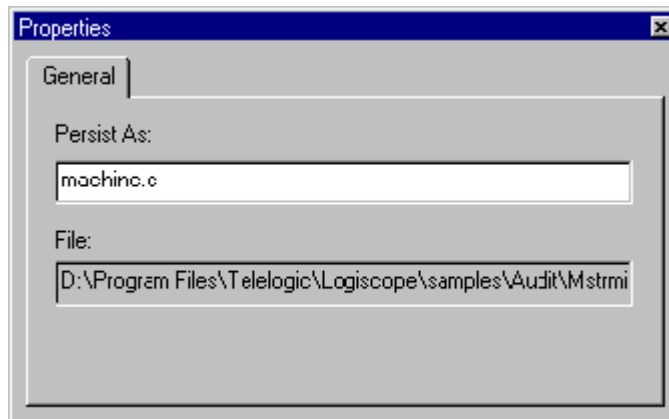
No modification is allowed.

For entities with no available properties, the following box is displayed:

Result

Allowed modification causes a new display in the view.

Possible errors

An illegal modification causes an error message display: "The new value of PersistAs is not correct". Click **OK** to close the dialog box if any change has been done.

Quality - Pop-up menu

Hide or display the Quality view

View

Can be launched from a tab in the **Workspace View**.

Context

Always available when an *QualityChecker* project is loaded.

Result

With a right click on the tab, a pop-up menu appears. The view can be hidden or displayed depending on its current state.

Quality - Criteria Level - Browse menu

Display the criteria values for the current project

View

Can be launched from any view.

Context

Only available when an *QualityChecker* project is loaded and built.

Seek command

- Button palette ⊕

Result

This command displays the **Criteria** tab in the **Workspace View**. It gives the criteria values for the current project.

A double click on a criterion opens its description in html format in the result pane.

A double click on a class or on a function opens its source file.

A double click on an item (application, class or function) in a criterion category displays the **Browse** tab in the output window with metrics used in the quality model for the related scope (Application, Class or Function).

A double click on a metric in the **Browse** tab opens its description in html format in the result pane.

A click on a column header in the **Browse** tab sorts the list in ascending order.

Quality - Factor Level - Browse menu

Display the factor values for the current project

View

Can be launched from any view.

Context

Only available when an *QualityChecker* project is loaded and built.

Seek command

- Button palette

Result

This command displays the **Factors** tab in the **Workspace View**. It gives the factor values for the current project.

A double click on a factor opens its description in html format in the result pane.

A double click on a class or on a function opens its source file.

A double click on an item (application, class or function) in a factor category displays the **Browse** tab in the output window with metrics used in the quality model for the related scope (Application, Class or Function).

A double click on a metric in the **Browse** tab opens its description in html format in the result pane.

A click on a column header in the **Browse** tab sorts the list in ascending order.

Quality - Quality Report - Browse menu

Display the quality report for the current project

View

Can be launched from any view.

Context

Only available when an *QualityChecker* project is loaded and built.

Seek command

- Shortcut: <F9>

- Button palette

Result

This command opens the **Quality Report** in the result pane. The links in html format allow access to quality elements related to the project:

- the value and the description of factors, criteria and metrics for application, class and function scopes,

- the list of source files.

Note: The report is generated in the tmp and is deleted when the report window is closed, but it can be saved using the "Save As" command.

Rebuild all - Project menu

Regenerate all results (output and files) for the active project

View

Can be launched from the **File View**.

Context

Always available.

Seek command

- Shortcut: <F7>
- Button palette

Result

This command generates results (output and files) for the active project. Even if the project has already been built, this forces a rebuild of the entire project. The options used to generate are those of the active configuration (see the **Settings... - Project menu** command for more details). The output files are generated in the project directory.

Note: consult Telelogic Logiscope RuleChecker & QualityChecker Getting Started and Telelogic Logiscope TestChecker Getting Started for more information on the generation.

Recent Files - File menu

List the last eight files used and allow to choose a file to open in this list

View

Can be launched from any **Textual View**.

Context

Always available.

Result

The chosen file is opened in the result pane.

Recent Workspaces - File menu

List the last eight workspaces used and allow to choose a workspace to load in this list

View

Can be launched from any **Textual View**.

Context

Always available.

Result

The chosen workspace is loaded.

Redo - Edit menu

Restore a previously undone action

View

Can be launched from all views.

Context

Available after the **Undo - Edit menu** command.

*Note: if an action has been performed after the **Undo** command, the **Redo** command is no longer available.*

Seek command

- Shortcut: <Ctrl+Y>
- Pop-up menu
- Button palette

Result

The last actions canceled with the **Undo** command are restored.

Refresh - Button palette

Reload an HTML page

View

Can be launched from the **Script Wizard** or an **HTML** view.

Context

Always available.

Seek command

- Button palette

Result

The current page is loaded again and the display is refreshed.

Replace - Edit menu

Find a character string and replace it by another one

View

Can be launched from a **Textual View**.

Context

Always available.

Seek command

- Shortcut: <Ctrl+H>

Dialog

A dialog box is opened. It contains the following fields:

- **Find what** to enter the character string to be searched; a scrolling list containing the previous searched strings also allows to choose one of them without re-entering it. If a string was selected in the text before starting the command, this string appears in this field.

- **Replace with** to enter the replacement for the searched character string; a scrolling list containing the previous replacement strings also allows to choose one of them without re-entering it.

- **Match whole word only**: select it to search isolated occurrences of the string only.

- **Match case**: select it to discriminate between uppercase and lowercase characters.

- **Replace In**: select **Selection** to limit the find and replace operation to the previously selected text; select **Whole file** to use the "find and replace" method in the whole document.

The dialog box contains the following buttons:

- **Find Next** to find the next occurrence of the string.

- **Replace** to replace the current occurrence and find the next occurrence of the string.

- **Replace All** to replace all occurrences.

- **Cancel** to cancel the command.

Revert - File menu

Revert to the previous state of the file

View

Can be launched on any file.

Context

Always available.

Result

The previous state of the file is restored and the display is refreshed.

Rule - Rule Violations - Browse menu

Display the rule violations for the current project

View

Can be launched from any view.

Context

Only available when a *RuleChecker* project is loaded and built.

Seek command

- Button palette 🚦

Result

This command displays the **Violations** tab in the **Workspace View**. It gives the violated, the clean (not violated) and ignored rules for the current project.

A double click on a rule opens its description in html format in the result pane.

A double click on a violated line opens the corresponding source file and highlights the violation.

Rule - Rule Violations by Files - Browse menu

Display the rule violations by files for the current project

View

Can be launched from any view.

Context

Only available when a *RuleChecker* project is loaded and built.

Result

This command displays the **File Violations** tab in the **Workspace View**. It displays the files in the current project where the violations occurred.

A double click on a rule opens its description in the result pane.

A double click on a violated line opens the corresponding source file and highlights the violation.

Rule - Rule Violations List - Browse menu

Display the rule violation list for the current project

View

Can be launched from any view.

Context

Only available when a *RuleChecker* project is loaded and built.

Seek command

- Button palette

Result

This command displays the rule violation list in the **Browse** tab of the output window. It gives for each rule the violations location (line).

A double click on a violated line opens the corresponding source file and highlights the violation.

A click on a column header in the **Browse** tab sorts the list in ascending order.

Rule - Rule Violations Report - Browse menu

Display the rule violation report for the current project

View

Can be launched from any view.

Context

Only available when a *RuleChecker* project is loaded and built.

Seek command

- Button palette

Result

This command displays the rule violation report in the result pane the links in html format allow access to violations by file and by rule for the current project.

Note: The report is generated in the tmp and is deleted when the report window is closed, but it can be saved using the "Save As" command.

Save - File menu

Save the selected file and associated files

View

Can be launched from the **File View**.

Context

A file (**.ttw**, **.ttp** or **textual**) must be selected.

Seek command

- Shortcut: <Ctrl+S>
- Button palette 💾

Result

- If a textual file is selected, only this file is saved.
- If a workspace is selected, this command is equivalent to the **Save Workspace - File menu** command (the workspace, its projects and all the modified Telelogic Logiscope files are saved).
- If a project is selected, it is saved as well as all the modified Telelogic Logiscope files.

Possible errors

One file cannot be write accessed. A message is displayed in an **Error** box. The command is canceled.

Save All - File menu

Save all modified files

View

Can be launched from any view.

Context

A workspace is opened.

Seek command

- Button palette

Result

All modified files of the current workspace are saved.

Possible errors

One file cannot be write accessed. A message is displayed in an **Error** box. The command is canceled.

Save As - File menu

Save a description in a new file

View

Can be launched from any **Textual View**.

Context

Always available.

Seek command

- Pop-up menu

Dialog

The **Save As** dialog box opens. Use this file browser to name a file to be saved. If an existing file is selected, a warning is displayed: click 'Yes' to override the existing file, or 'No' to cancel and come back to the **Save As** dialog box.

Result

The current description is saved in the designated file. This command has no impact on the current project: the new file is not referenced by the current project (meaning this command can be seen as an export command).

Possible errors

The designated file cannot be write accessed. A message is displayed in an **Error** box. The command is canceled.

Save Workspace - File menu

Save the current workspace

View

Can be launched from any view.

Context

A workspace is opened.

Result

The current workspace, its projects and all modified Telelogic Logiscope files are saved.

Note: The workspace and the projects are saved even if no change is made.

Possible errors

One file cannot be write accessed. A message is displayed in the **Error** box. The command is canceled.

Script Wizard - Tools menu

Open the Script Wizard

View

Can be launched from any view.

Context

Always available.

Seek command

- Shortcut: <Ctrl+W>

- Button palette

Result

This command opens the **Script Wizard** that allows to navigate in the **UML** representation of a Telelogic Logiscope model and to create Tcl scripts to extract information from the model (for more details see paragraph 2.17, *Script Wizard*).

Select All - Edit menu

Select all entities in the current Textual View

View

Can be launched from a **Textual View**.

Context

Always available.

Seek command

- Shortcut: <Ctrl+A>
- Pop-up menu

Result

All current view entities are selected.

Select from Properties - Browse menu

Allow to select classes or methods/functions for the current project according to a criterion or a factor evaluation degree

View

Can be launched from any view.

Context

Only available when an *QualityChecker* project is loaded and built. If in a multi-project context, only the active (currently selected) project is considered.

Result

This command displays a **User Request** tab in the **Workspace View**. It gives for the current project the list of classes or methods/functions for the chosen criterion or factor evaluation degree.

A double click on a class or on a method/function opens its source file in the result pane and its metrics values in the **Browse** tab of the output window.

A double click on a metric in the **Browse** tab opens its description in html format in the result pane.

A click on a column header in the **Browse** tab sorts the list in ascending order.

Note: make as many requests as needed by adding them in the User Request tab or by creating a new tab.

Settings... - Project menu

Set current tool options for the current project and configuration

View

Can be launched from any view.

Context

A project is active.

Seek command

•   Shortcut: <Alt+F7>

Dialog

A **Settings** dialog box opens. It displays the project settings with several tabs depending on the type of the selected project: i.e. the programming language and the applicable Logiscope verification modules.
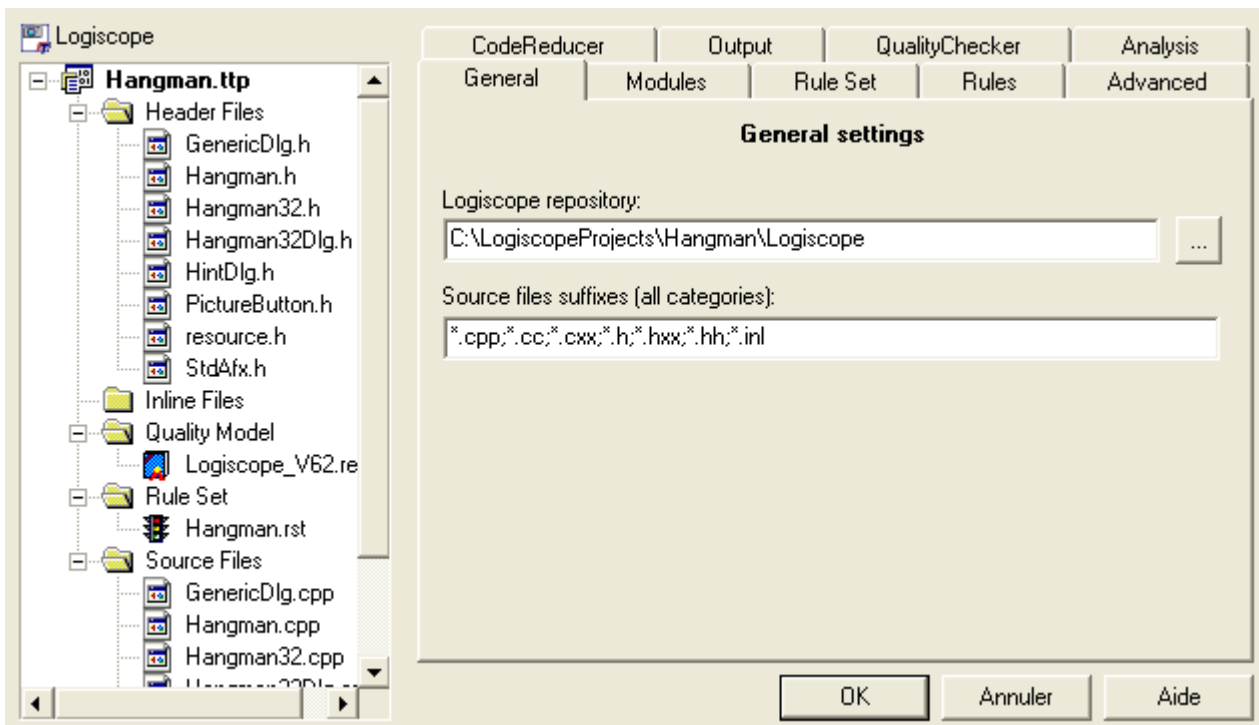
More information on project settings are available in *Telelogic Logiscope RuleChecker & QualityChecker - Getting Started* and *Logiscope TestChecker - Getting Started*.

Shortcuts - View menu

Display or hide the Shortcuts

View

Can be launched from any view.

Context

Always available.

Seek command

- Pop-up menu on a toolbar

Result

Shortcuts are displayed or hidden, depending on whether they were hidden or displayed. A check box displayed before the command name in the **View** menu indicates the current state of the **Shortcuts**: if the box is checked, they are currently displayed; if the box is cleared, they are hidden.

Small Icons - Pop-up menu

Reduce the size of the icons in the Shortcuts View

View

Can be launched from the **Shortcuts View**.

Context

Always available.

Result

A pop-up menu appears when right-clicking in a shortcut list. The selection of the **Small Icons** option allows to display the icons of the list as small icons.

Source Control - Pop-up menu

Hide or display the Source Control bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Source Control - Add to Source Control - Project menu

Add an item into the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Source Control - Check In - Project menu

Save an item into the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Source Control - Check Out - Project menu

Restore an item from the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Source Control - Get lastest version - Project menu

Get the lastest version of an item in the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Source Control - Import from Source Control - Project menu

Import from the source code control system

View

Can be launched from any view.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

* Button palette

Result

The source code control system is invoked and the requested action is executed.

Source Control - Refresh Status - Project menu

Refresh status according to the source code control system

View

Can be launched from any view.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette

Result

The source code control system is invoked and the requested action is executed.

Source Control - Remove from Source Control - Project menu

Remove an item from the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Source Control - Show Differences - Project menu

Show differences on an item between its current state and its state in the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Source Control - Show History - Project menu

Show an item's history archived in the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Source Control - Source Control - Project menu

Execute the source code control system

View

Can be launched from any view.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

• Button palette

Result

The source code control system is invoked and the requested action is executed.

Source Control - Source Control Properties - Project menu

Display the properties of the source code control system

View

Can be launched from any view.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

* Button palette

Result

The source code control system is invoked and the requested action is executed.

Source Control - Undo Check Out - Project menu

Undo the operation in the source code control system

View

Can be launched from a file or a folder.

Context

Always available when a source code control system is installed on the machine and if the "**Connect to the Source Code Control System at startup**" option is checked (see **General** tab in **Options - Tools menu**).

Seek command

- Button palette
- Pop-up menu on a file or a folder

Result

The source code control system is invoked and the requested action is executed.

Standard - View menu

Display or hide the Standard toolbar

View

Can be launched from any view.

Context

Always available.

Seek command

• Pop-up menu on a toolbar

Result

The **Standard** toolbar is displayed or hidden, depending on whether it is hidden or displayed. A check box displayed before the command name in the **View** menu, indicates the current state of the toolbar: if the box is checked, the toolbar is currently displayed; if it is cleared, the toolbar is hidden.

Start Link - Link menu

Allow to define a link on an object

View

Can be launched from a file or a folder.

Context

Always available (even if the object has already some links).

Seek command

- Pop-up menu on the object
- Button palette
- Shortcut: <Ctrl+K>

Result

A dialog window is opened, allowing to define a link on the object.

Start Many Links - Link menu

Allow to make many links from start

View

Can be launched from any view.

Context

Available if not already active.

Seek command

- Button palette

Result

This command allows to make links from start. Once it is activated, it is replaced by the **Clear Start Many Links** command and allows the access to the **Make Link from Start** command in the Link menu.

The **Clear Start Many Links** command restores the initial state.

Start *TestChecker* **-** Project menu

Start Telelogic Logiscope *TestChecker*

View

Can be launched from any built project.

Context

Only available for a *TestChecker* built project.

Seek command

- Button palette

Result

This command launches *TestChecker*.

Start Viewer - Project menu

Start Telelogic Logiscope Viewer

View

Can be launched from any built project.

Context

Always available for a built project except for a *RuleChecker* project.

Seek command

- Button palette

Result

This command launches the *Viewer*.

Stop - Button palette

Stop the loading of an HTML page

View

Can be launched from the **Script Wizard** or an **HTML** view.

Context

A page is being loaded.

Seek command

- Button palette

Result

This command interrupts a page loading.

Stop - Project menu

Stop the execution of the build command

View

Can be launched from any view.

Context

Only available during a **Build** command execution.

Seek command

- Button palette

Result

This **Build** command is stopped and the following message is displayed in the output window (**Build** tab): Build canceled by user.

Style - Tools menu

Not available

Tcl Script - Pop-up menu

Hide or display the Tcl Script bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Telelogic Direct Contact - Help menu

Display Telelogic Logiscope feedback

View

Can be launched from any view.

Context

Always available.

Result

An html file containing links to contact Telelogic directly or to get information on products is displayed.

Test - Component Coverage - Browse menu

Display tested components with DDPs

View

Can be launched from any view.

Context

Only available when a *TestChecker* project with a dynamic result file is loaded.

Seek command

- Button palette

Result

Components are listed in the **Browse** tab of the output window with their DDP coverage and the corresponding test.

A double click on a line in the **Browse** tab displays the corresponding source file in the result pane.

A click on a column header in the **Browse** tab sorts the list in ascending order.

Test - Test Report - Browse menu

Display the test report for the current project

View

Can be launched from any view.

Context

Only available when a *TestChecker* project with a dynamic result file is loaded.

Seek command

- Button palette

Result

This command opens the test report in the result pane.

Note: The report is generated in the tmp and is deleted when the report window is closed, but it can be saved using the "Save As" command.

Tests - Pop-up menu

Hide or display the Tests view

View

Can be launched from a tab in the **Workspace View**.

Context

Always available when a *Test* project is loaded.

Result

With a right click on the tab, a pop-up menu appears. The view can be hidden or displayed depending on its current state.

Tile Horizontally - Window menu

Arrange windows as non horizontally overlapping tiles

View

Can be launched from any view.

Context

Always available.

Result

When the windows are horizontally tiled, *Studio* displays entirely each window - no windows overlapping. All windows are resized as necessary to fit inside the result pane.

Tile Vertically - Window menu

Arrange windows as non vertically overlapping tiles

View

Can be launched from any view.

Context

Always available.

Result

When the windows are vertically tiled, *Studio* displays entirely each window - no windows overlapping. All windows are resized as necessary to fit inside the result pane.

To Shortcut - Pop-up menu

Convert a toolbar into an embedded listbar in the Shortcuts View

View

Can be launched from Logiscope toolbar.

Note: The Menu, Standard, Browse, Tcl Script, UML Browser and Project bars cannot be converted into Shortcuts View.

Context

Always available.

Seek command

- Pop-up menu on the Logiscope toolbar

Result

A pop-up menu appears with a right-click on a toolbar. The selection of the **To Shortcut** option allows to display the toolbar as a shortcut list.

To display the shortcuts as a toolbar again, click right and choose the **As Toolbar - Pop-up menu** option.

Undo - Edit menu

Cancel a previous action

View

Can be launched from any view.

Context

Always available, providing that at least one editing action has been performed.

Seek command

- Shortcut: <Ctrl+Z>
- Pop-up menu
- Button palette

Result

The previous action is canceled. The previous internal context and the previous concerned view are restored. If a view is closed after actions have been performed in it, the actions can be undone but the view is not opened again.

The **Undo** command can be launched as many times as there are previously performed actions.

Actions involving file saving cannot be undone.

Use the **Redo - Edit menu** command to restore undone actions.

Web - Pop-up menu

Hide or display the Web bar

View

Can be launched from a toolbar.

Context

Always available.

Result

With a right click on a toolbar, a pop-up menu appears. The toolbar can be hidden or displayed depending on its current state.

See **Toolbars** tab in **Customize - Tools menu** command.

Workspace - View menu

Display or hide the Workspace Window

View

Can be launched from any view.

Context

Always available.

Seek command

- Shortcut: <Alt+0>
- Pop-up menu on a toolbar

Result

The **Workspace View** (which includes the **File View** tab) is displayed or hidden, depending on whether it is hidden or displayed. A check box displayed before the command name in the View menu indicates the current state of the workspace area: if the box is checked, the window is currently displayed; if cleared it is hidden.

Telelogic Logiscope

Telelogic Logiscope

Telelogic Logiscope

同

Telelogic Logiscope

Telelogic Logiscope

Chapter 4

Quality Model Editor

4.1 Starting the Quality Model Editor

The Quality Model Editor allows to display and modify the Quality Model used for a *Logiscope QualityChecker* project.

With *Logiscope Studio*, load or create a *QualityChecker* project. Double-click on the quality model file in the project workspace. The quality model editor is displayed together with the metrics, criteria and factors used in the quality model for the required language.

The picture below represents *Studio* with the quality model editor:
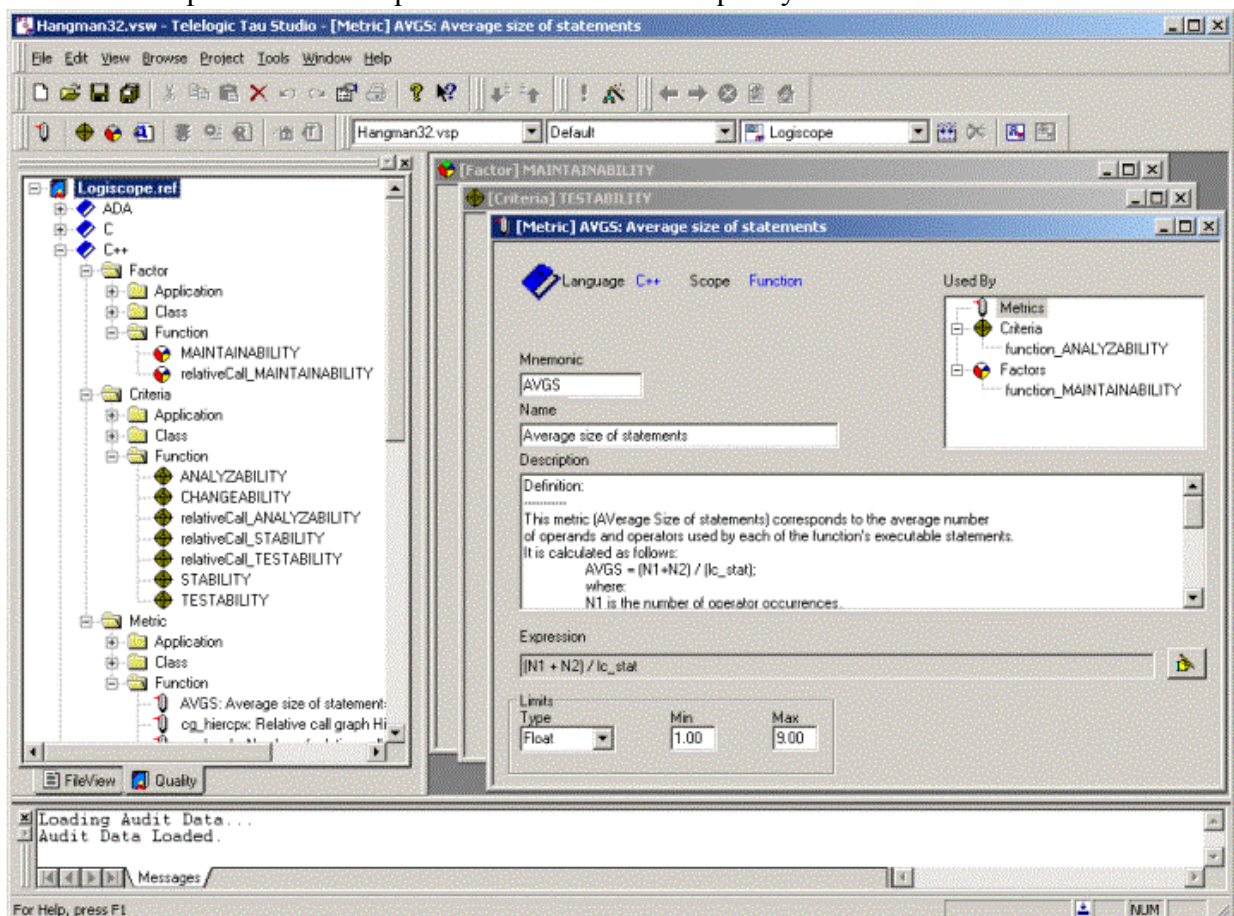
Default Quality Model files are supplied: e.g. **Logiscope.ref** in the **Ref** folder of the Logiscope standard installation directory.

4.2   Overview of Quality Model Editor

Views

When the quality model editor is activated, three windows are displayed in the result pane:

- **[Factor]** quality editor,
- **[Criteria]** quality editor,
- **[Metric]** quality editor.

With the use of these editors, all the existing items in the quality model can be modified and new items can be added.

[Factor] quality editor

There are four main user definable areas:

- **Name:** contains the name of the factor item.
- **Description:** contains a comment about the factor item.

- **Expression:** contains the expression used to calculate the factor item. To modify this area, the expression editor must be used. This editor is opened with the ![button] button.
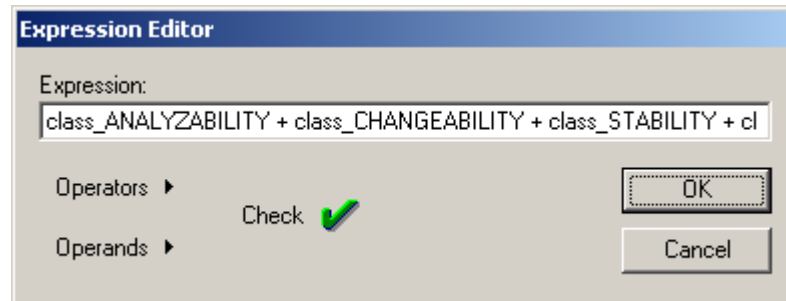
Operands and *Operators* displays a pop-up list of all the criteria items and operators that are available to construct the required factor level expression.

Check can be selected to verify the syntax of the edited expression. Any errors found are displayed in the output window of *Studio*.

OK updates the expression area in the **[Factor]** window.

- **Categories:** contains the evaluation degrees together with their maximum and minimum values. These values can be modified by selecting the value with the left mouse button.
If conflicting values are selected, an error is displayed in the output window.

New evaluation degrees can be added to the categories list with the use of the *Insert* key and can be deleted with the use of the *Delete* key. Renaming is also possible by clicking twice on the evaluation degree.

Note: To add new evaluation degrees, there must be a Min/Max gap between the evaluation degrees where the insertion is required.

Criteria] quality editor
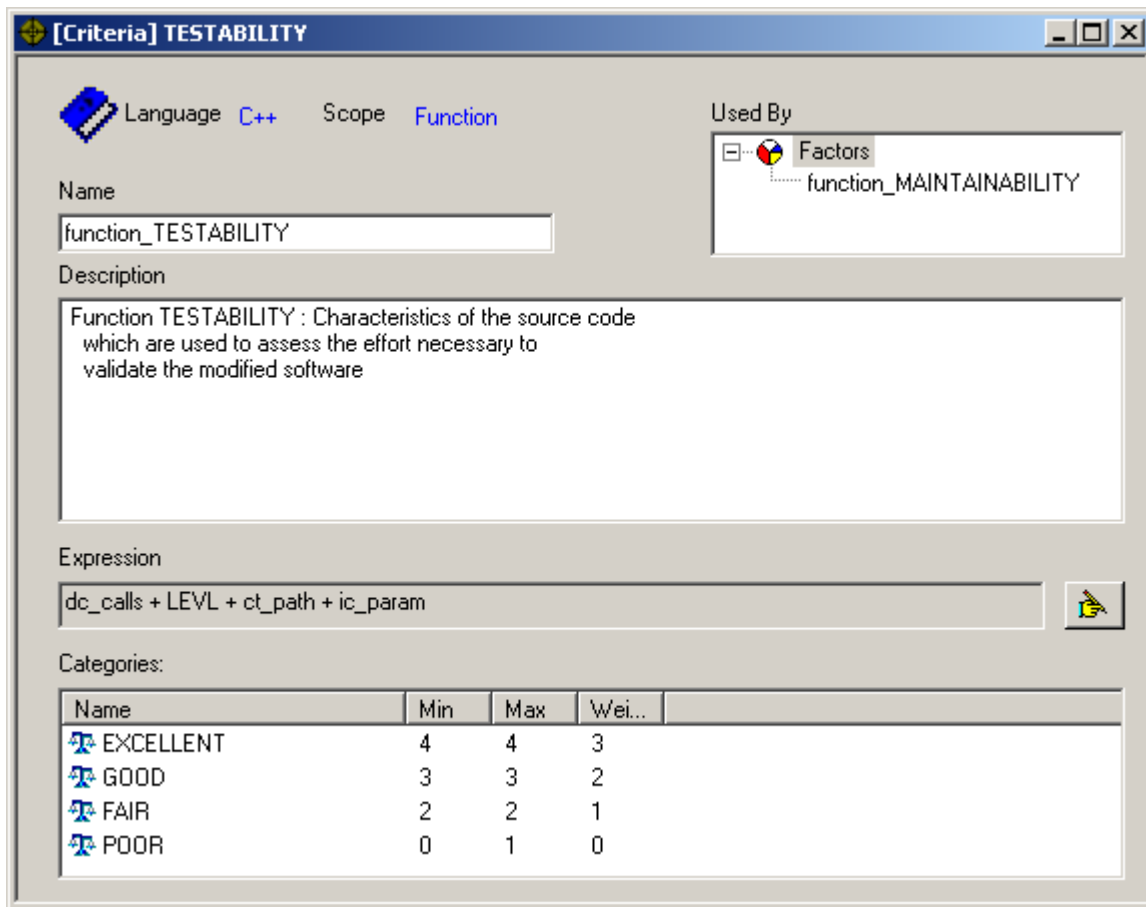
[There are five main user definable areas:

- **Name:** contains the name of the criteria item.
- **Description:** contains a comment about the criteria item.
- **Used By:** displays all the factor items that use the criteria item in question.
- **Expression:** contains the expression used to calculate the criteria item. To modify this area, the expression editor must be used. This editor is opened with the button.
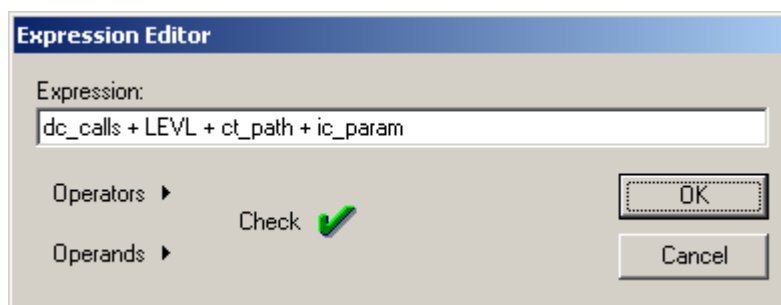
Operands and *Operators* displays a pop-up list of all the metric items and operators that are available to construct the required criteria level expression.

Check can be selected to verify the syntax of the edited expression. Any errors found are displayed in the output window of *Studio*.
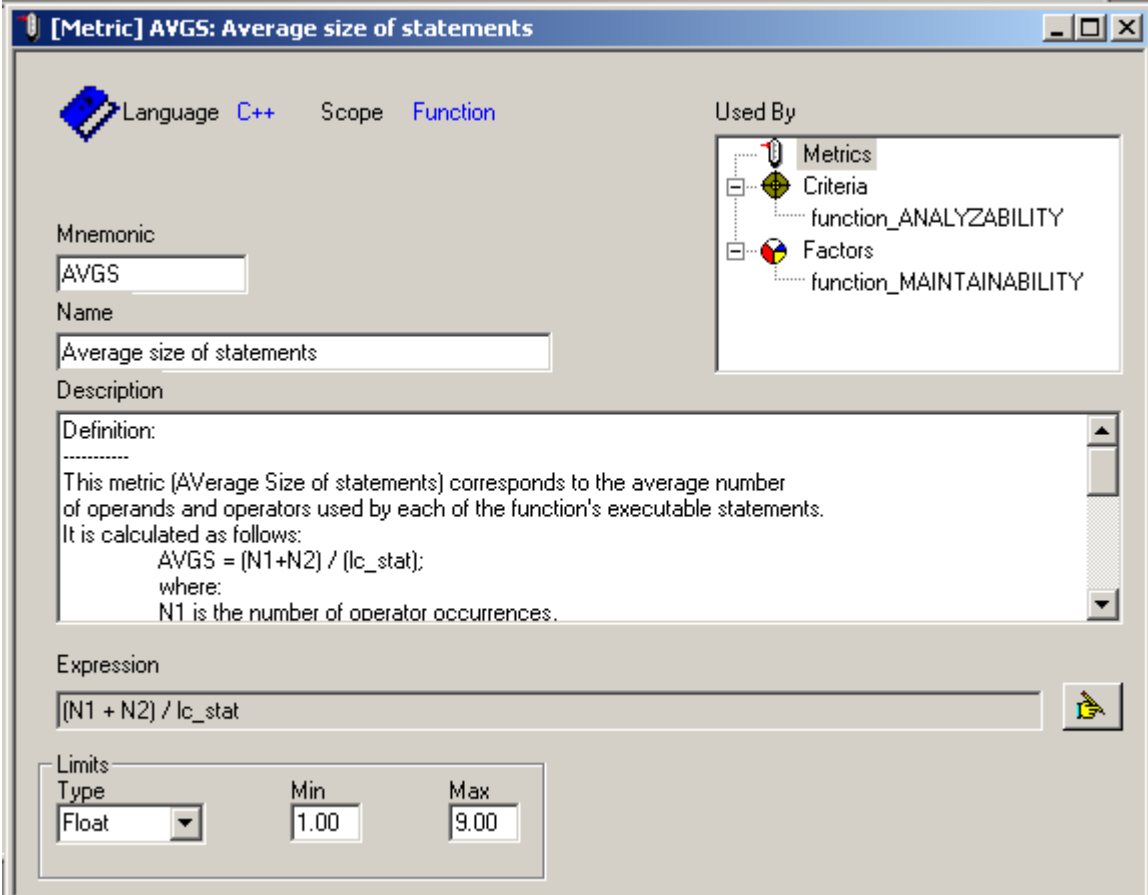
OK updates the expression area in the **[Criteria]** window.

- **Categories:** contains the evaluation degrees together with their maximum, minimum and weight values. These values can be modified by selecting the value with the left mouse button.

- Evaluation degrees can be added, deleted and renamed. See **[Factor]** quality editor for a full explanation.

[Metric] quality editor

There are six main user definable areas:

- **Mnemonic:** contains the mnemonic of the metric item.
- **Name:** contains the full name of the metric item.
- **Description:** contains a comment about the metric item.
- **Used By:** displays all the factor and criteria items that use the metric item in question.
- **Expression:** contains the expression used to calculate the metric item. To modify this area, the expression editor must be used. This editor is opened with the [icon] button.

> *Note: the expression area and expression editor are disabled for basic metric items.*
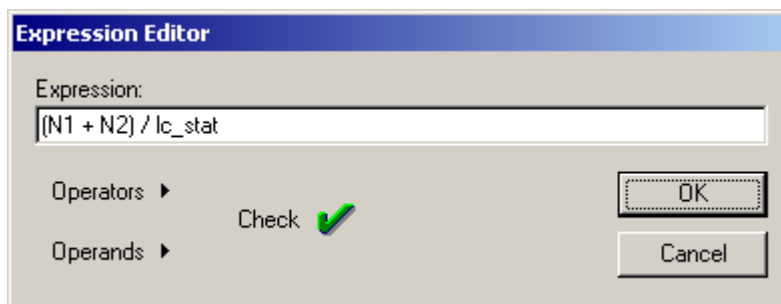
Expression Editor

Expression:

(N1 + N2) / lc_stat

Operators ▶

Check ✔

Operands ▶

OK

Cancel

Operands and *Operators* *displays a pop-up list of all the metric items and operators that are available to construct the required metric level expression.*

Check can be selected to verify the syntax of the edited expression. Any errors found are displayed in the output window of *Studio*.

OK updates the expression area in the **[Metric]** window.

- Limits: contains the type, together with its maximum and minimum, that can be used with the selected metric.

4.2.1 Quality (Tab) view
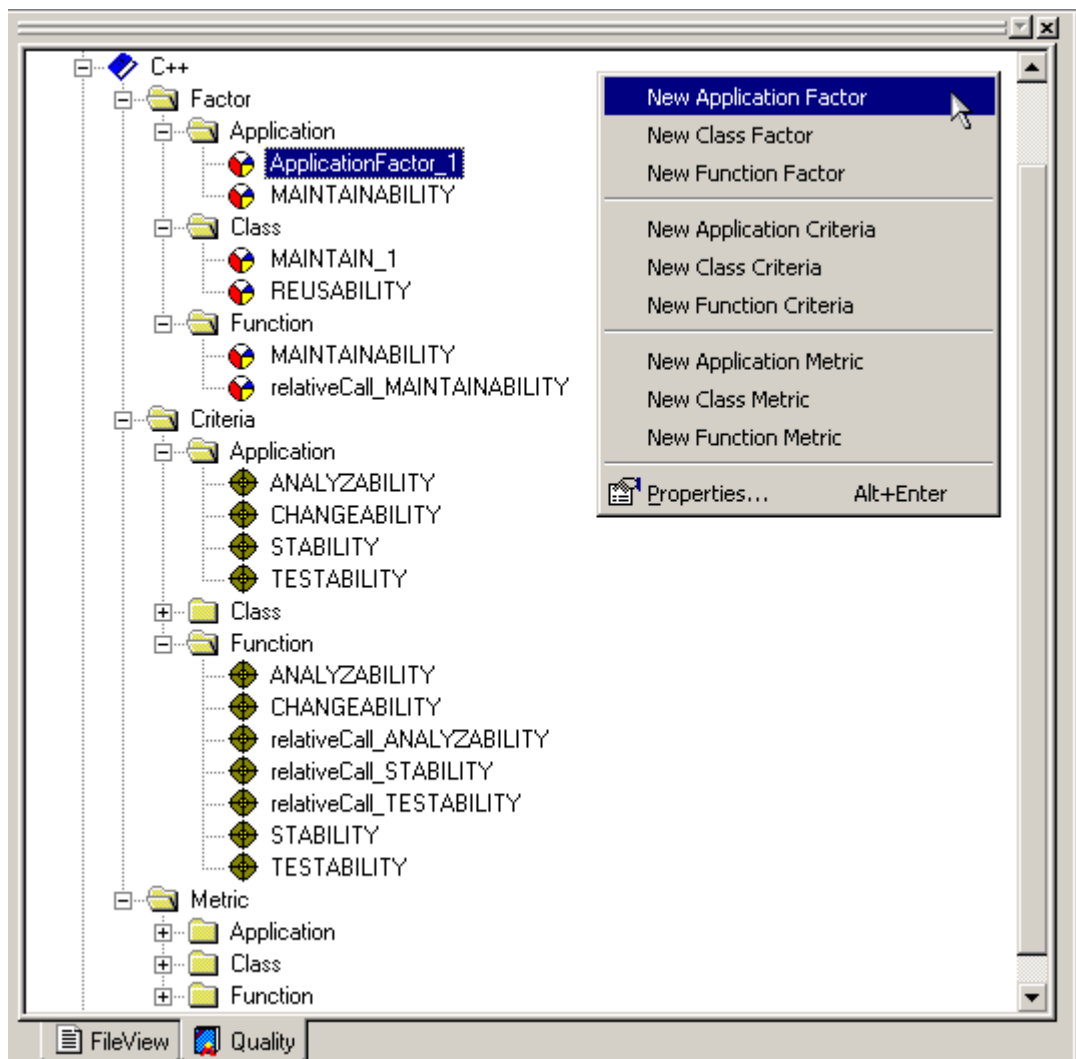
This quality view presents all the factor, criteria and metric items used for each language scope. The following actions can be performed on these items:

Add

New items can be added to the quality model. This is done by selecting a location in the quality view with the right mouse button. A pop-up menu is displayed with a list of item types that can be added. The items in the menu list are dependent on the cursor location.

When a new item is added, a corresponding blank editor window is displayed in the result pane. Modifications in this editor window can be performed as described above.

Delete

Existing items can be removed from the quality model. This is done by selecting the item in the quality view and pressing the **Delete** or **Backspace** keys. The item is removed from the quality view and the corresponding editor is cleared.

Only items with no links to other items can be deleted. For example, the item Criteria-Application-ANALYZABILITY can only be deleted if Factor-Application-MAINTAINABILITY is deleted first.

Rename

Existing items can be renamed in the quality model. This is done by clicking the item twice. Modification to the name in the quality view are duplicated in the corresponding editor window.

Appendix A

Data Model

A.1 Introduction

The Telelogic Logiscope API (Application Programming Interface) is based on a data model as described below.This UML metamodel describes the project structure and Logiscope entities. To see Logiscope data model, browse through **Script Wizard**. This function allows to create or adapt Tcl scripts to customize interface or generated results (for more details on the **Script Wizard** function see Section 2.17).

At first the symbols used in a data model representation will be explained. Then the data model itself will be shown in graphic form and in text format.

Examples of Tcl scripts using the Logiscope API as described in Appendix A can be found in the *<log_install_dir>/***Scripts** directory.

A.2 Concepts and symbolism

A data model is represented as a class diagram. Here is the definition and representation of object-oriented concepts appearing in graphic form in a data model.

A.2.1 Class

A class is a set of objects with similar properties (attributes), common behaviors (operations) and share relations with other objects.
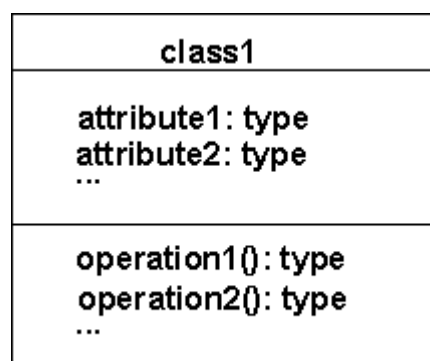
A.2.2 Attribute

An attribute is a data item specific to objects of a given class. Each attribute name is unique in its class. Each attribute has a value of the specified type (string, integer, etc.) for every object instance.

A.2.3 Operation

An operation is a function or a transformation that can be applied to objects of a class or carried out by them. All the objects in a given class share the same operations. The type associated to an operation indicates the type of value returned by the operation.

A.2.4 Link and association
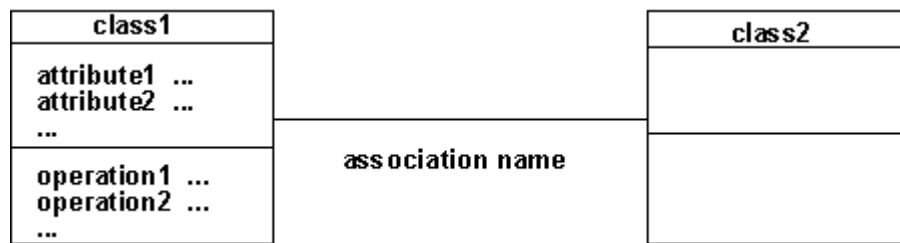
A link is a physical or conceptual connection between two instances of an object:

• A-to-B link and B-to-A link:
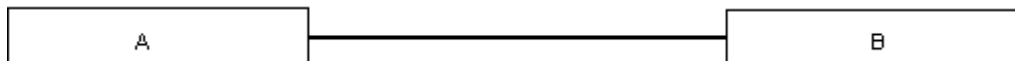
• A-to-B link only (the origin side of the link is indicated by the exclamation point !):
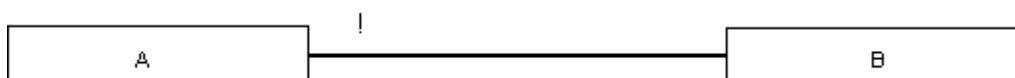
An association describes a set of links just as a class describes a set of objects.

A.2.5 Multiplicity

The multiplicity specifies the number of class instances related to an instance of the associated class. Multiplicity (or cardinality) can be a range of values, a set of values or a specific number.

*   1 instance of *A* is linked to 0 or 1 instance of *B*:

A			B
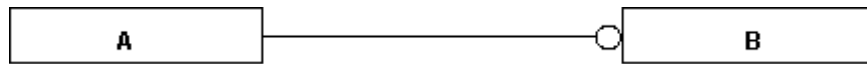
*   1 instance of A is linked to 0 or more instances of B:

A			B
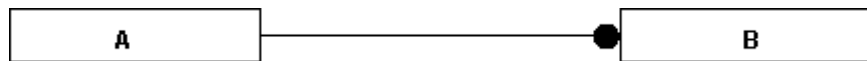
*   1 instance of *A* is linked to at least *n* instances of *B* ($n > 0$):

A	n+		B
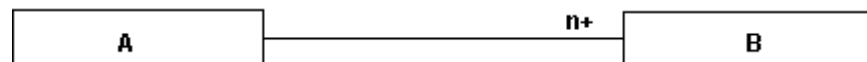
*   1 instance of *A* is linked to a number of instances of *B* between *m* and *n* inclusive:

A	m-n		B
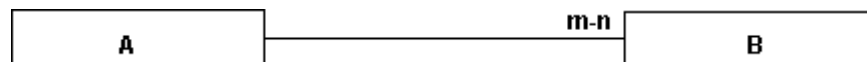
A.2.6 Role

A role is one end of an association. A binary association has two roles, each with its own name. The name of a role is a name that clearly identifies one end of an association. Roles make possible to consider a binary association as the link of one object to an associated set of objects. Each role in a binary association identifies an object or set of objects associated with an object at the other end.

class1		class2
attribute1 ... attribute2	role1　　　　　role2	
operation1 ... operation2	association name	
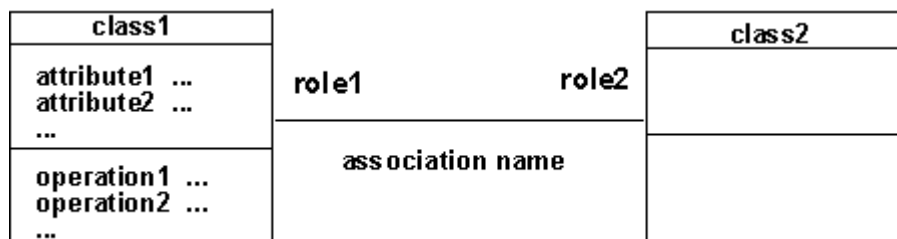
The name of a role is a derivative attribute whose value is a set of associated objects.

There are two cases for which roles must absolutely be named:

*   recursive associations,
*   several associations involving the same classes.

If roles are not named, the class name is taken as the role name, with the first letter changed to lower-case.

A.2.7 Inheritance

The "is a kind of" relation allows classes to share similarities and retain their differences.
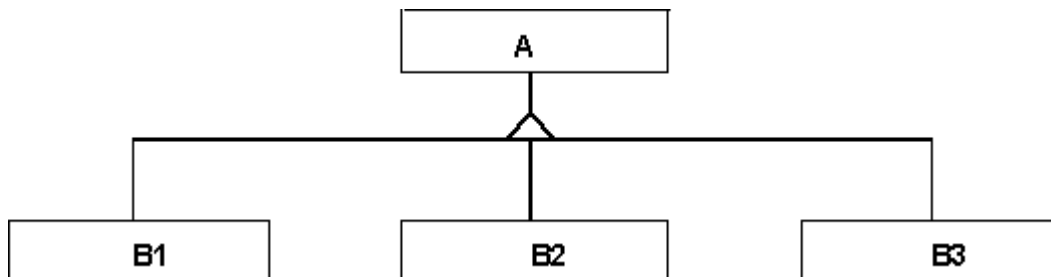
A.2.8 Abstract class
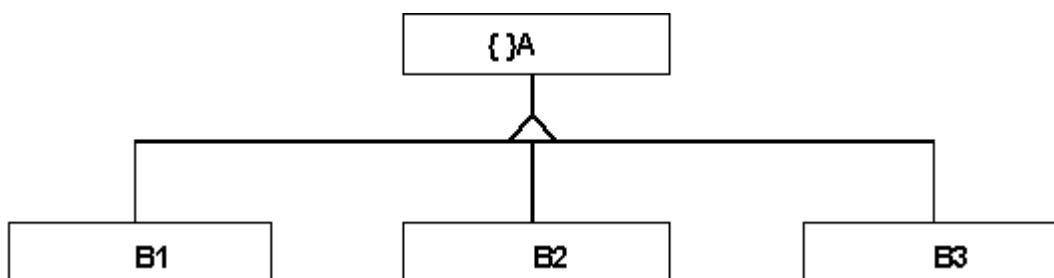
An abstract class is a class with no instantiated objects. Attributes and operations it describes are inherited by its sub-classes.

A.3 The data model

A.3.1 Graphical Representation
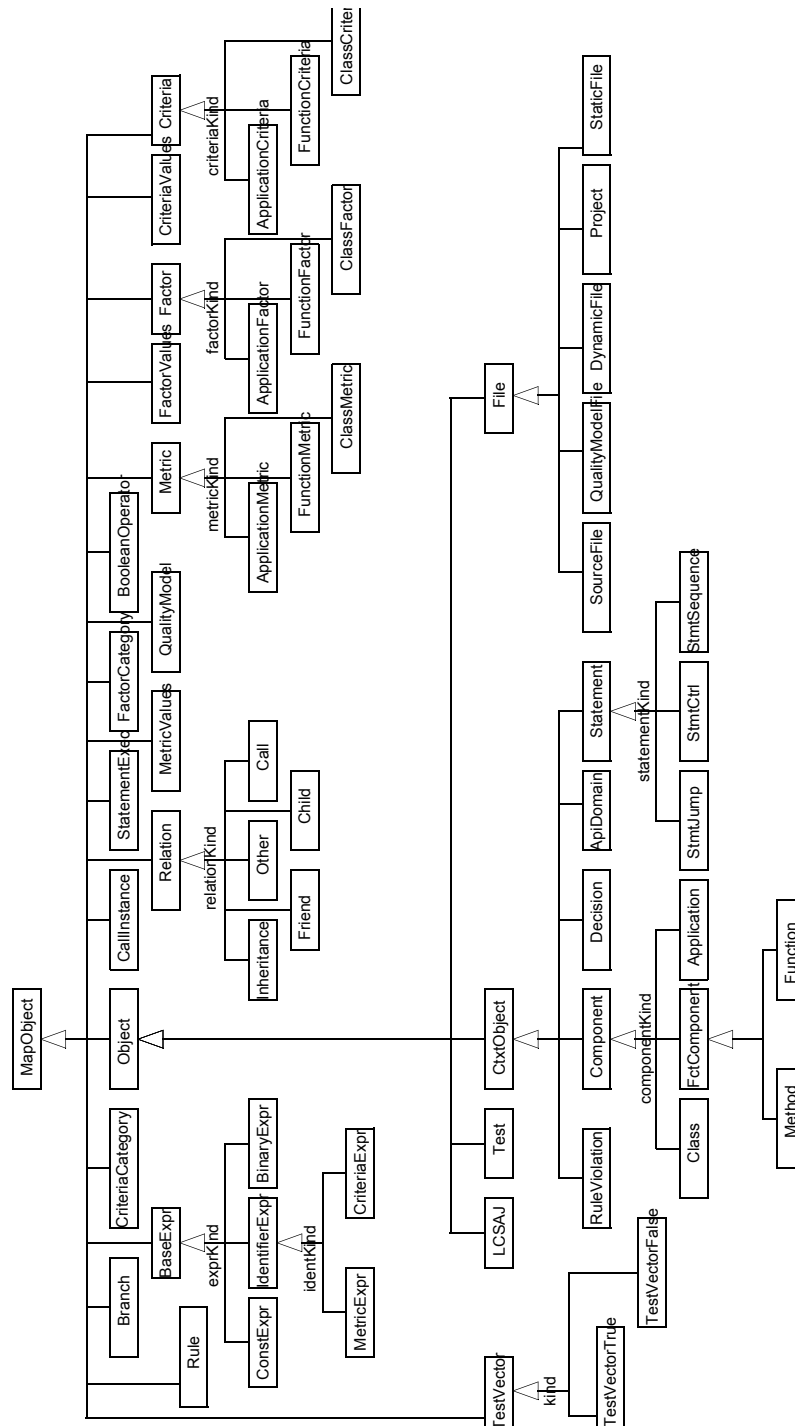
Figure 1 Inheritance overview diagram

CtxtObject

code : string
line : integer
start : integer
length : integer

Component

name : string
crc : integer
date : string
listEntries : CStringList

GetMetricValue(metric : Metric) : real
EvaluateComputedMetrics()
GetBasicMetricValue(basicMetric : BasicMetric) : real
GetCriteriaValue(criteria : Criteria) : CriteriaCategory
GetFactorValue(factor : Factor) : FactorCategory
EvaluateFactors()
EvaluateCriterias()
EvaluateQualityModel()
IsExternal() : boolean
GetLanguageName() : LPCTSTR

Statement

kind : integer
num : integer
instruction : string

Decision

notOperator : boolean
decisionNumber : integer

IsMcdc() : boolean
condCount() : integer

RuleViolation

Figure 2 ContextObject diagram
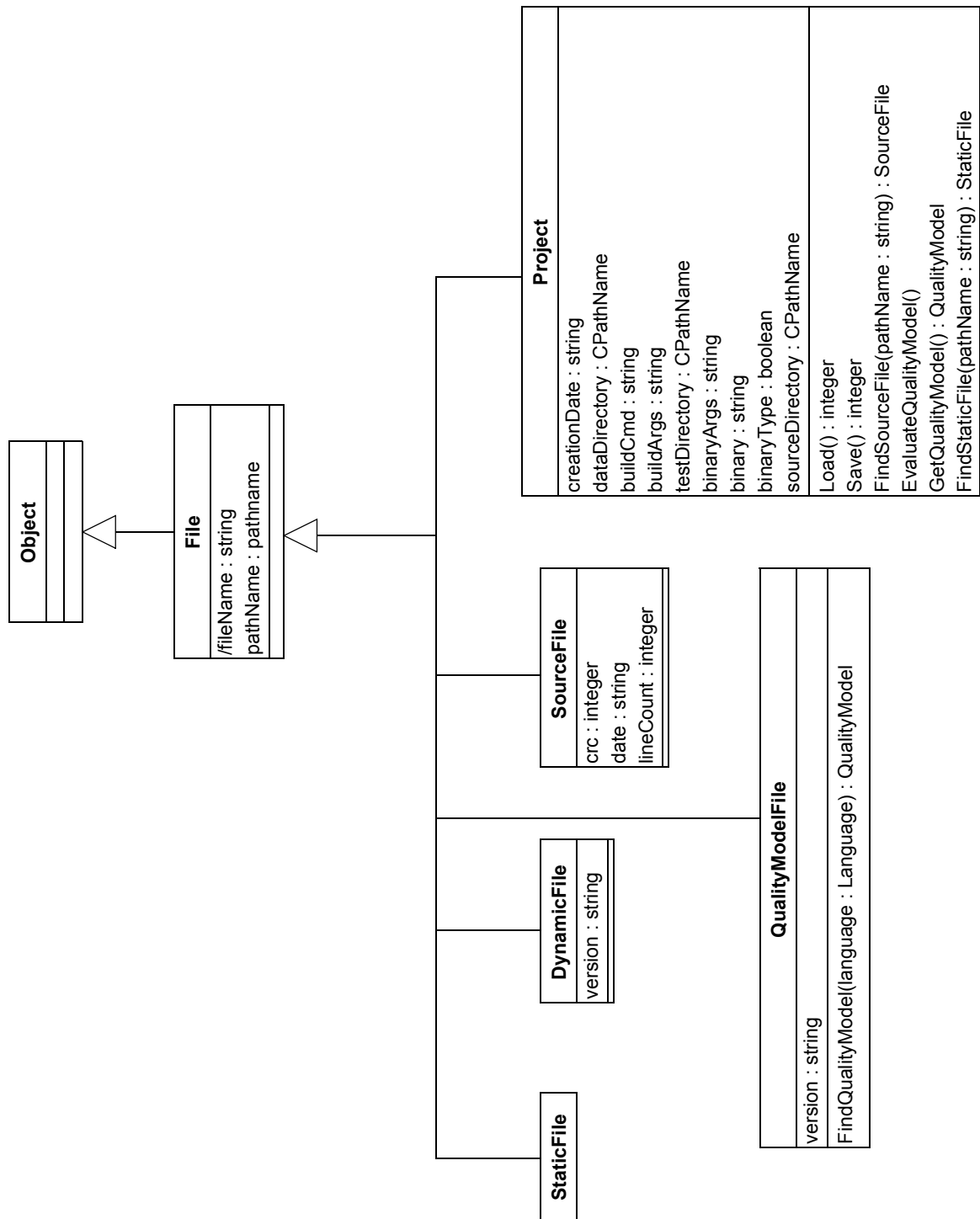
Figure 3 File diagram
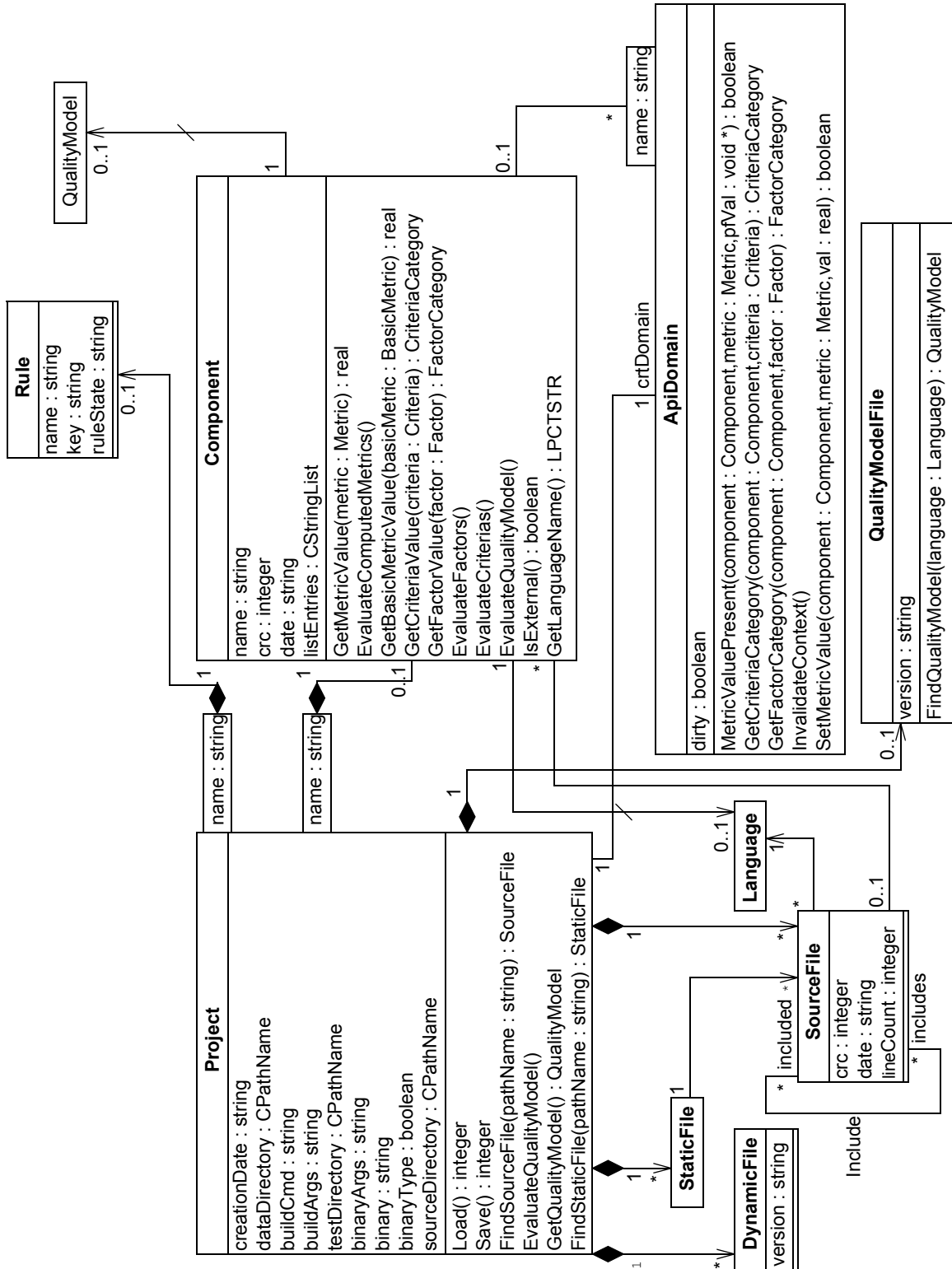
Figure 4 Project diagram
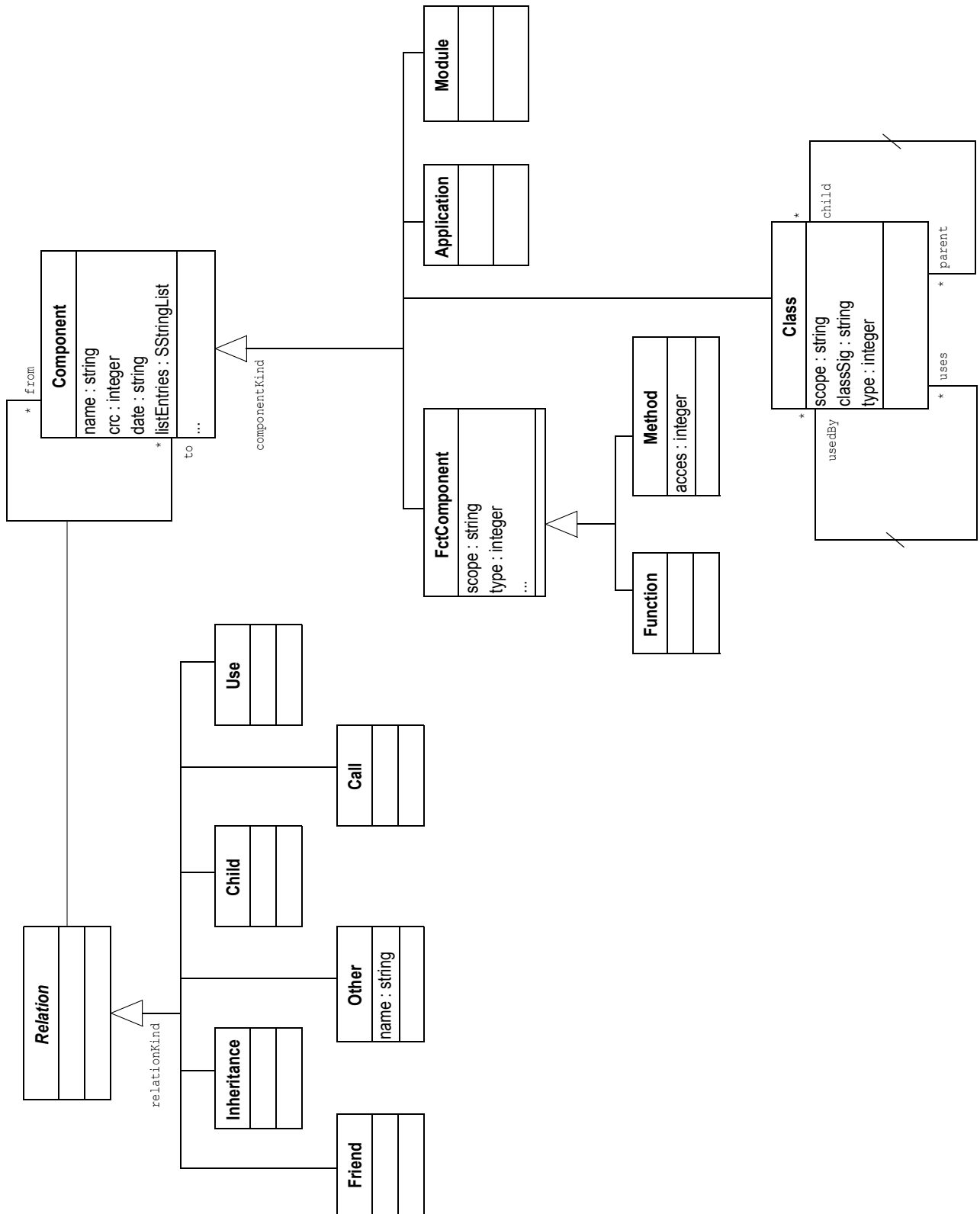
Figure 5 Component diagram

Figure 6 Statement diagram
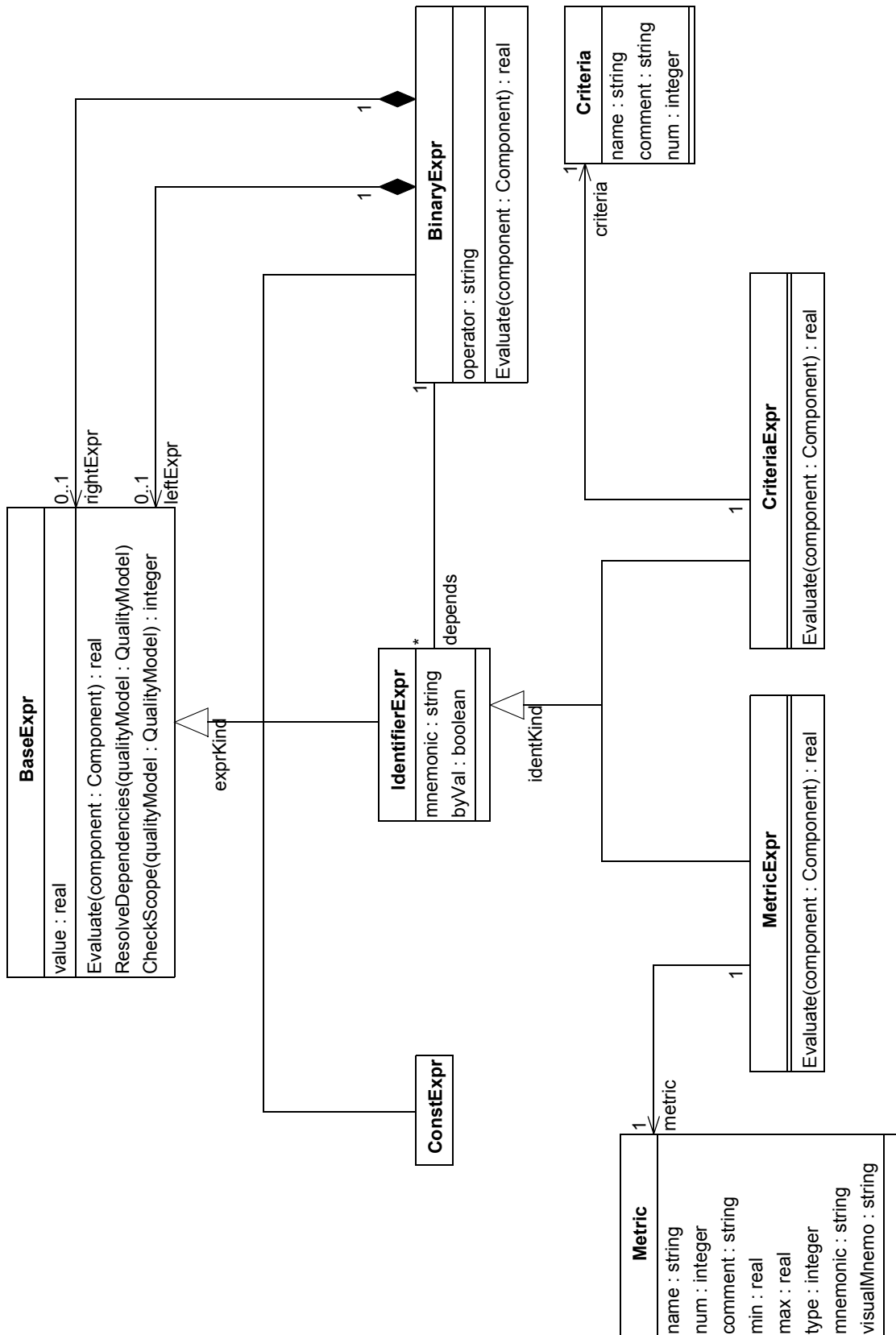
Figure 7 Expression diagram
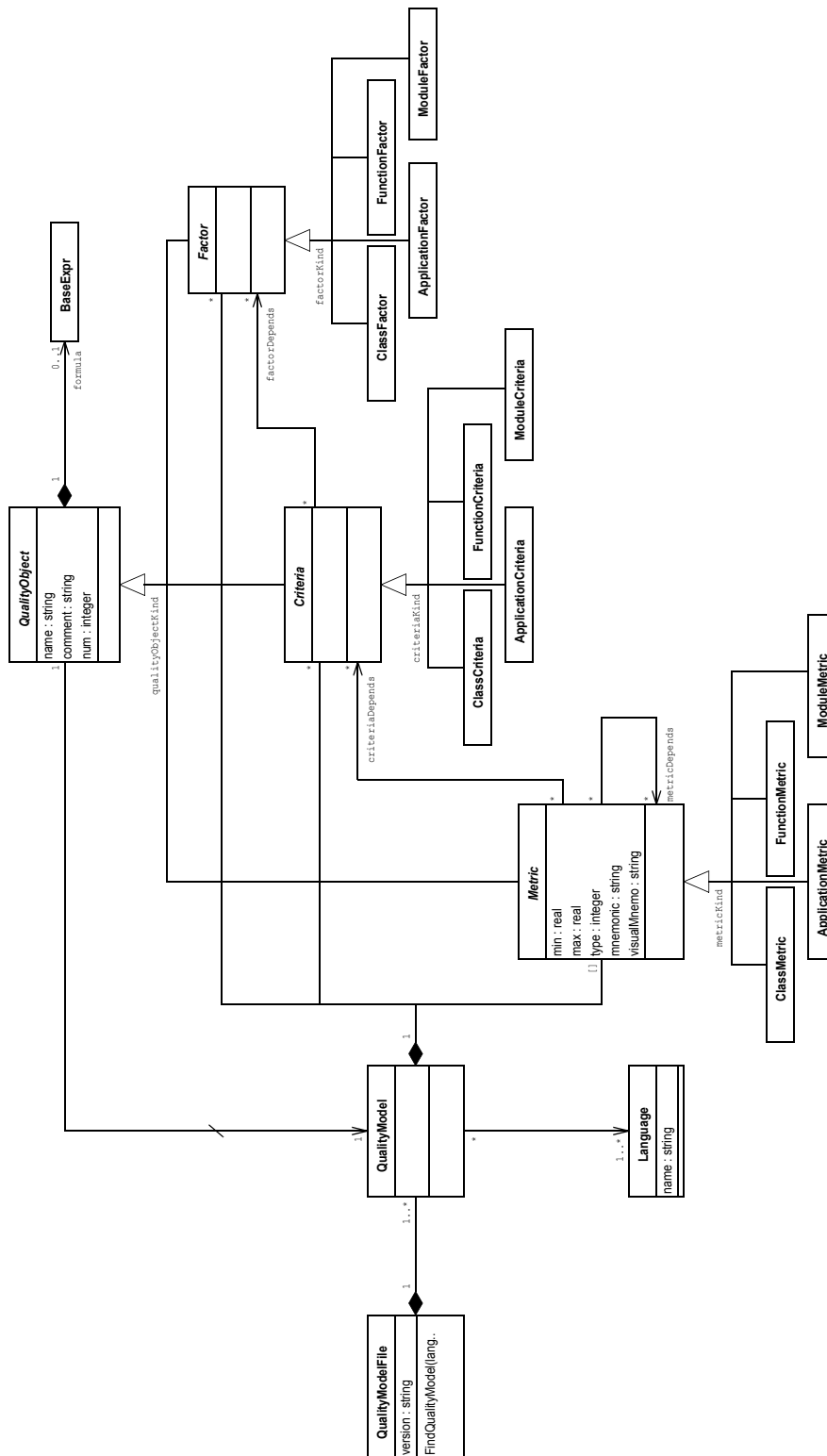
Figure 8 QualityModel diagram
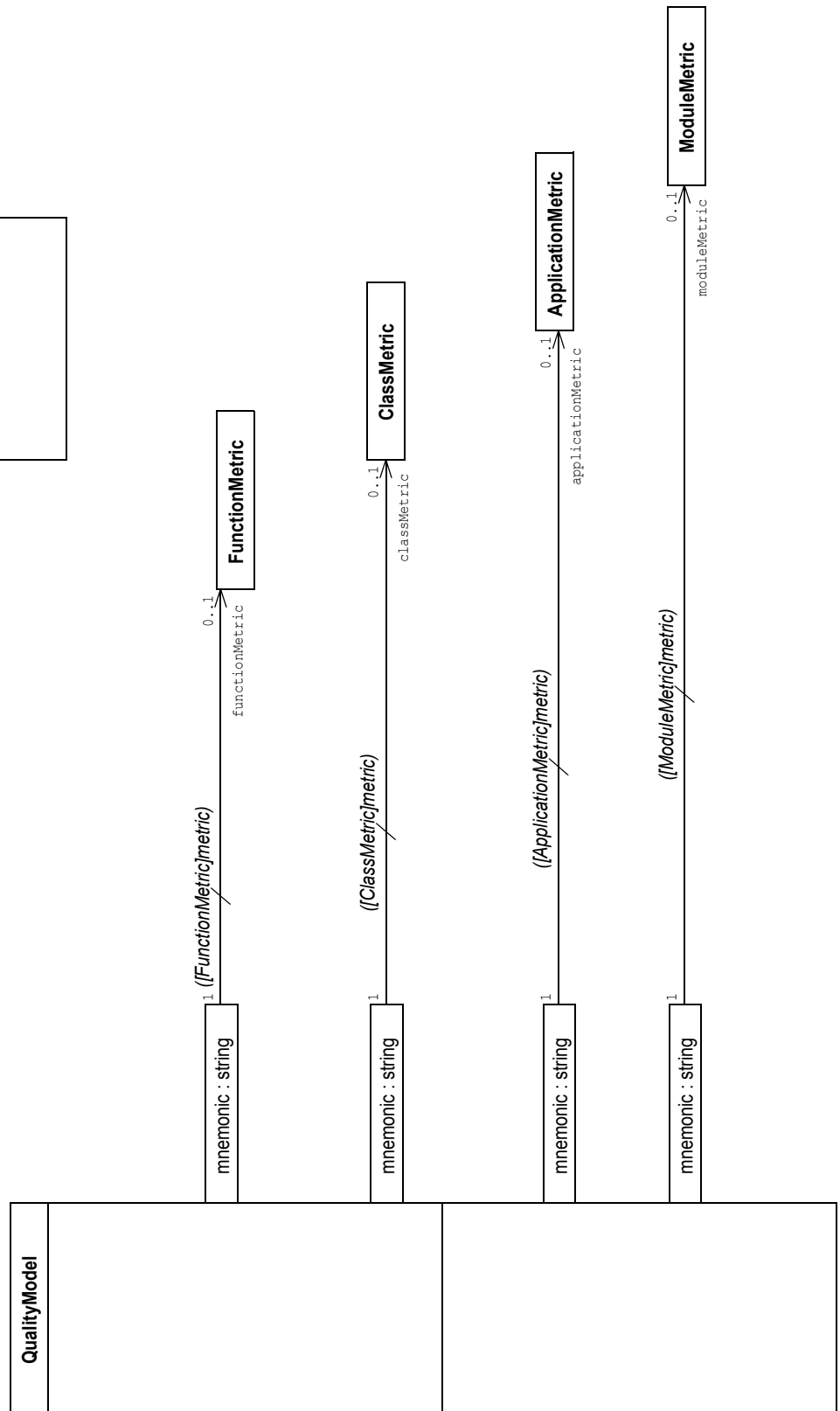
Figure 9 Metric Maps diagram
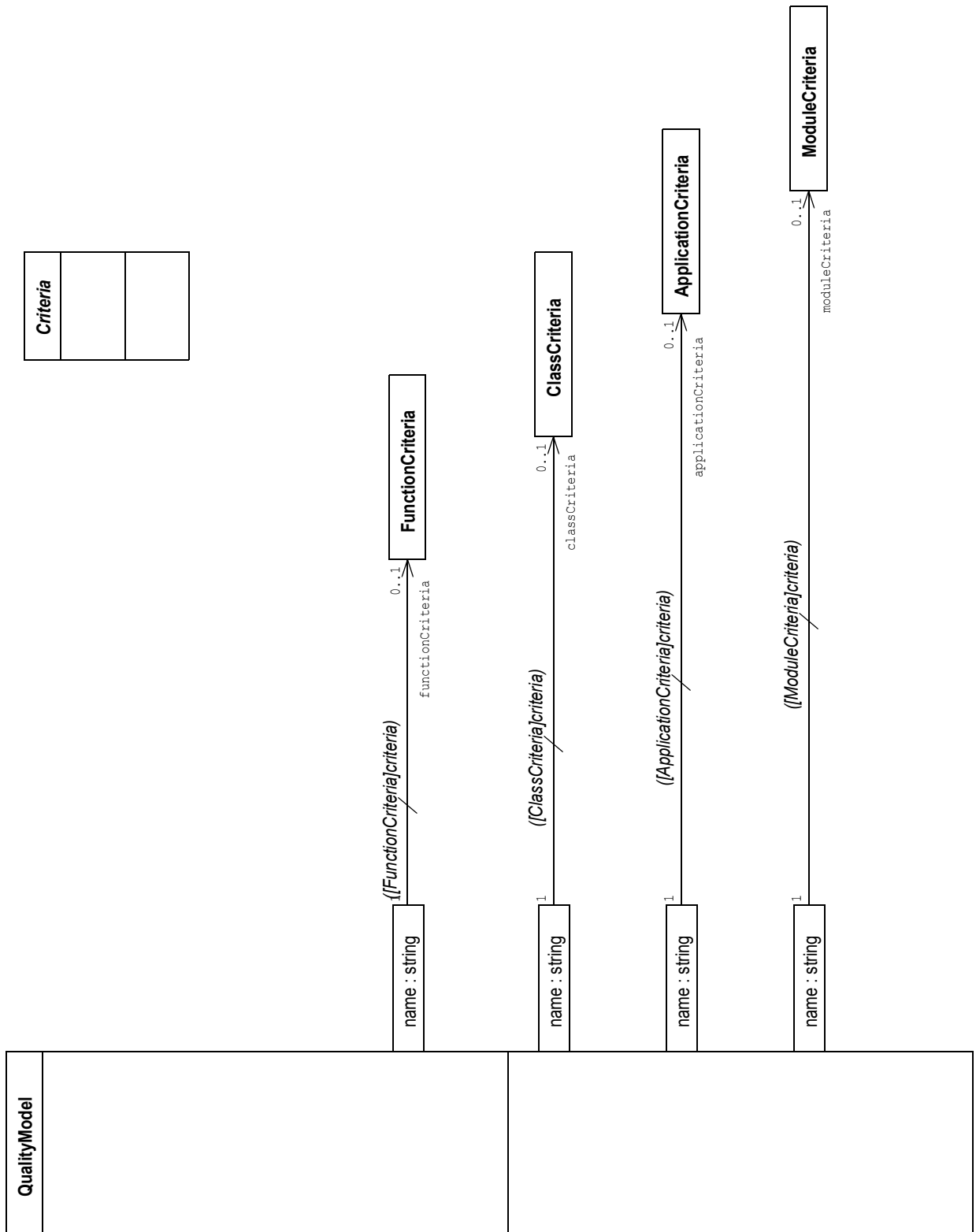
Figure 10 Criteria Maps diagram
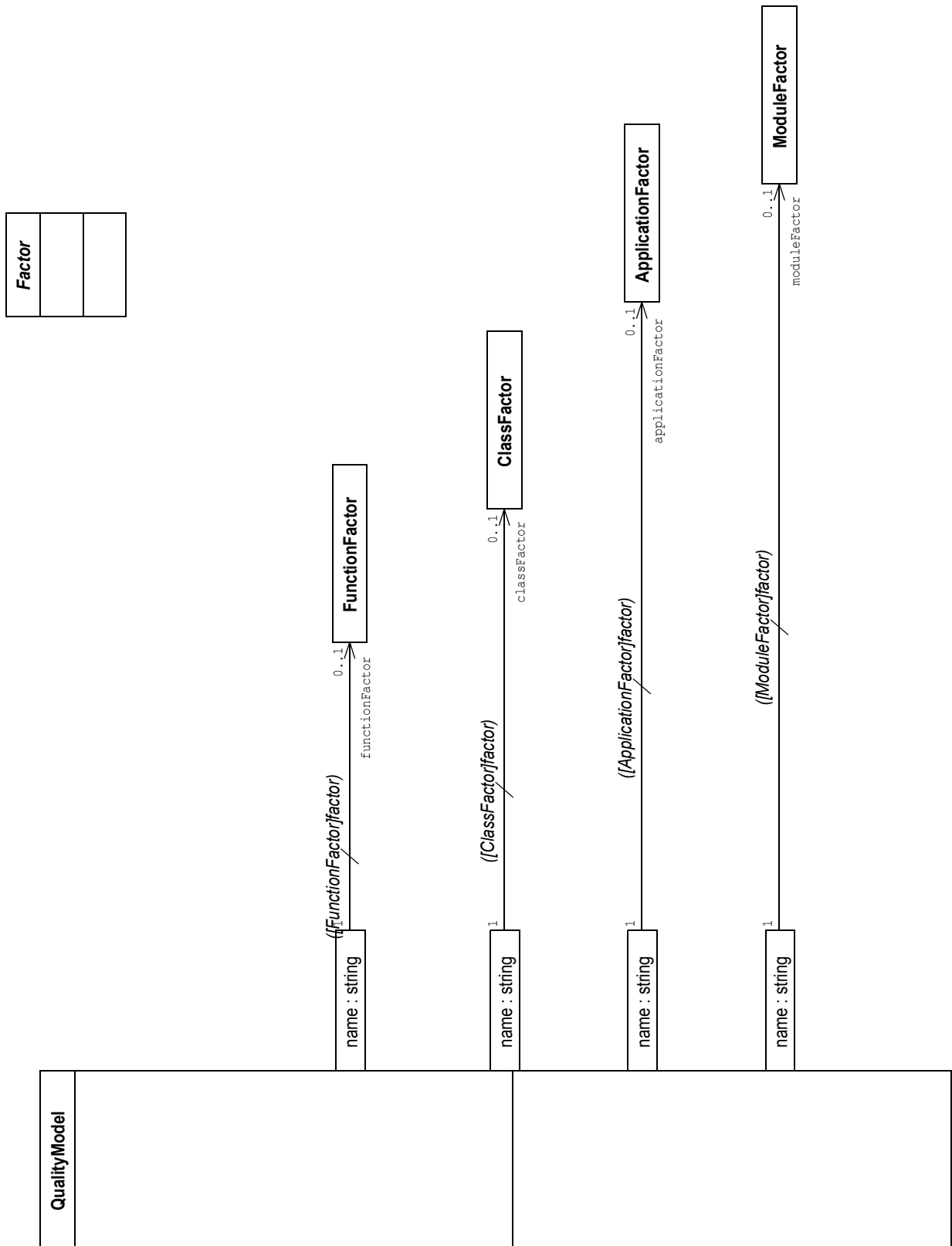
Figure 11 Factor Maps diagram

```
┌──────────────────────┐         ┌──────────────────────┐
│    CriteriaCategory  │         │    FactorCategory    │
├──────────────────────┤         ├──────────────────────┤
│ name : string        │         │ name : string        │
│ min : integer        │         │ min : integer        │
│ max : integer        │         │ max : integer        │
│ weight : integer     │         │                      │
└──────────────────────┘         └──────────────────────┘
        │ 1..*                            │ 1..*
        │                                 │
        │ 1                               │ 1
        ◆                                 ◆
┌──────────────────────┐         ┌──────────────────────┐
│       Criteria       │         │        Factor        │
├──────────────────────┤         ├──────────────────────┤
│ name : string        │         │ name : string        │
│ comment : string     │         │ comment : string     │
│ num : integer        │         │ num : integer        │
└──────────────────────┘         └──────────────────────┘
```
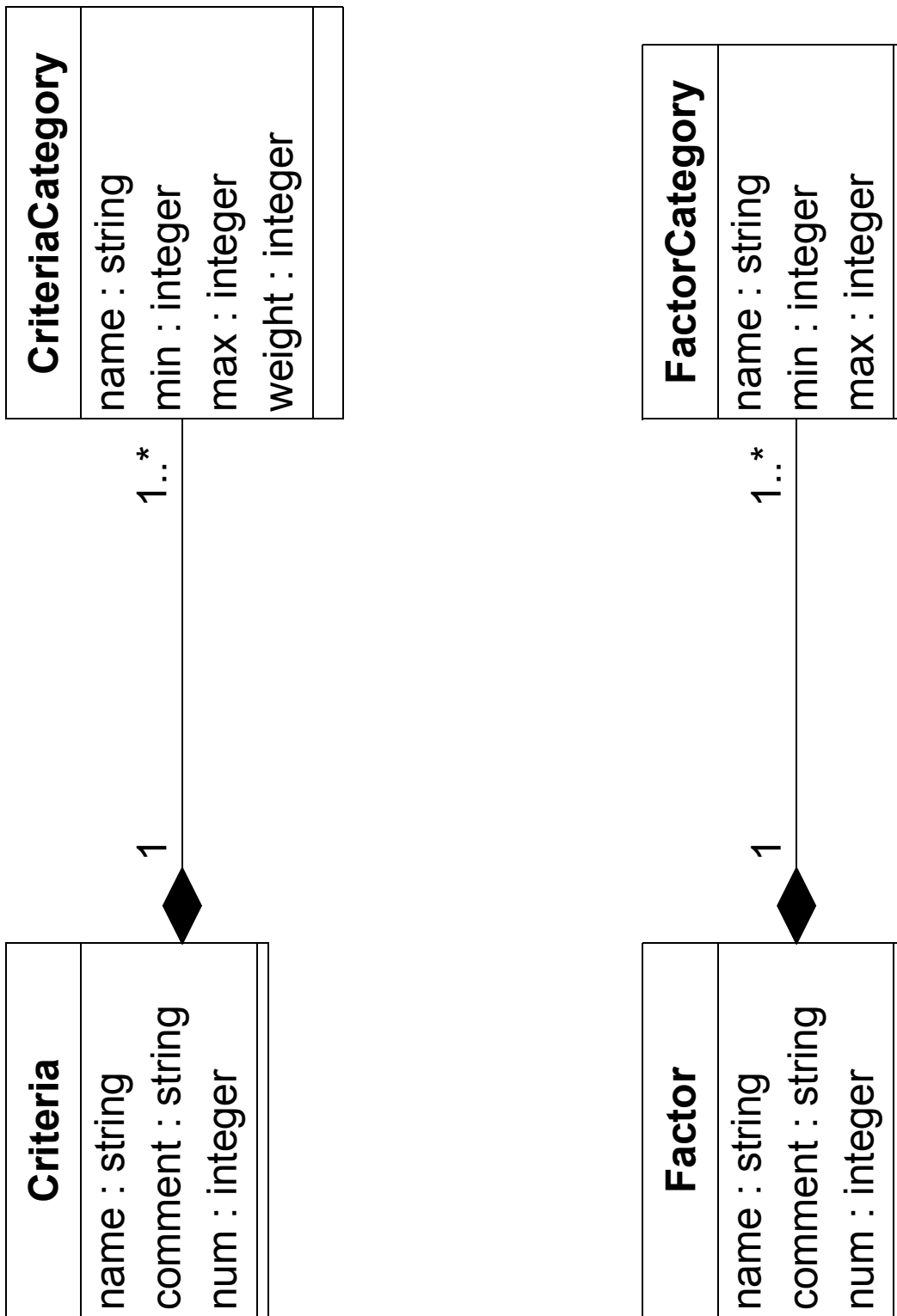
Figure 12 Category diagram

BasicMetric

name : string
comment : string
scope : integer
index : integer
pfnMetricEngine : LGS_METRIC_ENGINE_PFN = NULL

0..1

Metric

name : string
num : integer
comment : string
min : real
max : real
type : integer
mnemonic : string
visualMnemo : string

*

Component

name : string
crc : integer
date : string
listEntries : CStringList

GetMetricValue(metric : Metric) : real
EvaluateComputedMetrics()
GetBasicMetricValue(basicMetric : BasicMetric) : real
GetCriteriaValue(criteria : Criteria) : CriteriaCategory
GetFactorValue(factor : Factor) : FactorCategory
EvaluateFactors()
EvaluateCriterias()
EvaluateQualityModel()
IsExternal() : boolean
GetLanguageName() : LPCTSTR

*

Figure 13 Base Metric diagram
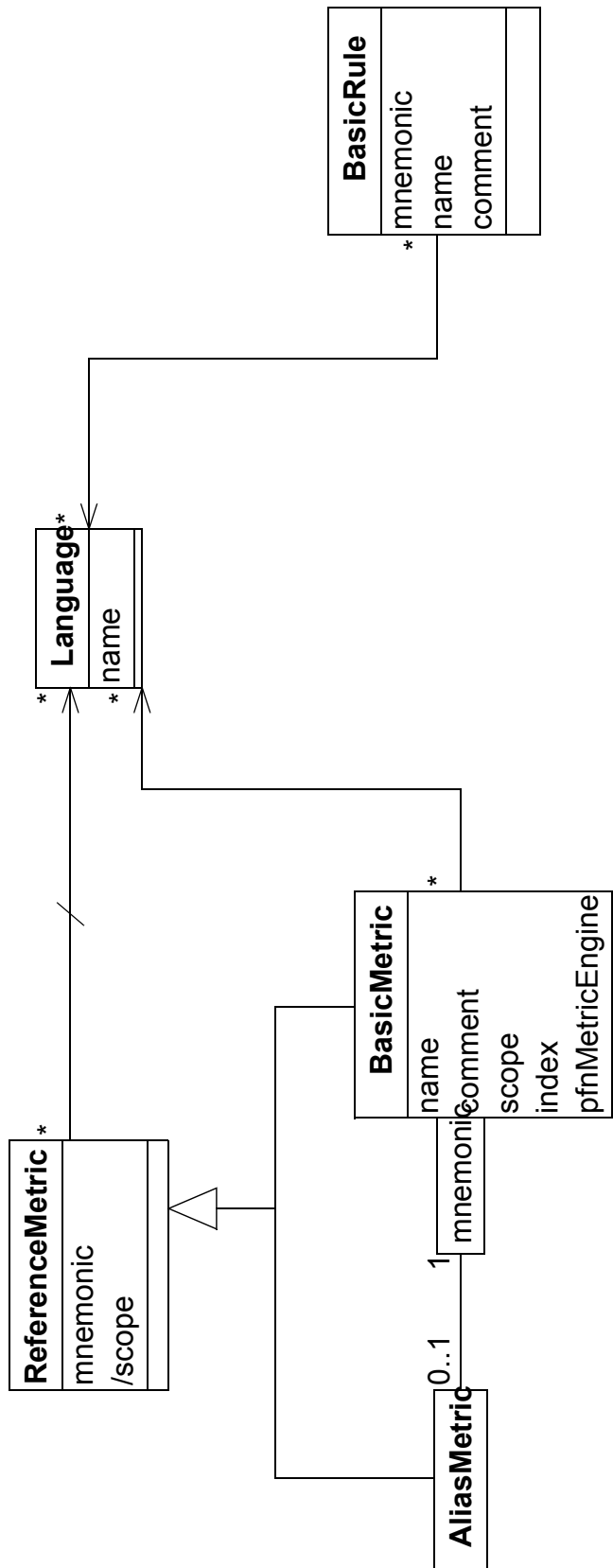
Figure 14 Base Rule diagram

Component

name : string
crc : integer
date : string
listEntries : CStringList

GetMetricValue(metric : Metric) : real
EvaluateComputedMetrics()
GetBasicMetricValue(basicMetric : BasicMetric) : real
GetCriteriaValue(criteria : Criteria) : CriteriaCategory
GetFactorValue(factor : Factor) : FactorCategory
EvaluateFactors()
EvaluateCriterias()
EvaluateQualityModel()
IsExternal() : boolean
GetLanguageName() : LPCTSTR

Rule

name : string
key : string
ruleState : string

BasicRule

mnemonic : string
name : string
comment : string

RuleViolation

SourceFile

crc : integer
date : string
lineCount : integer

Figure 15 Programming Rules diagram
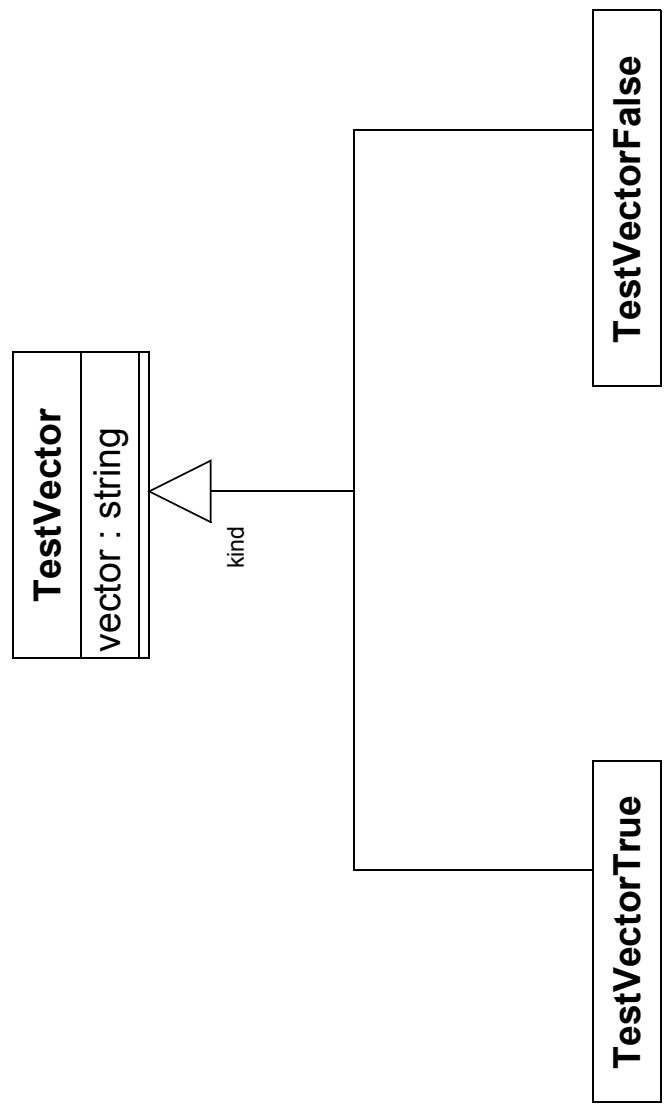
Figure 16 Test Vector diagram
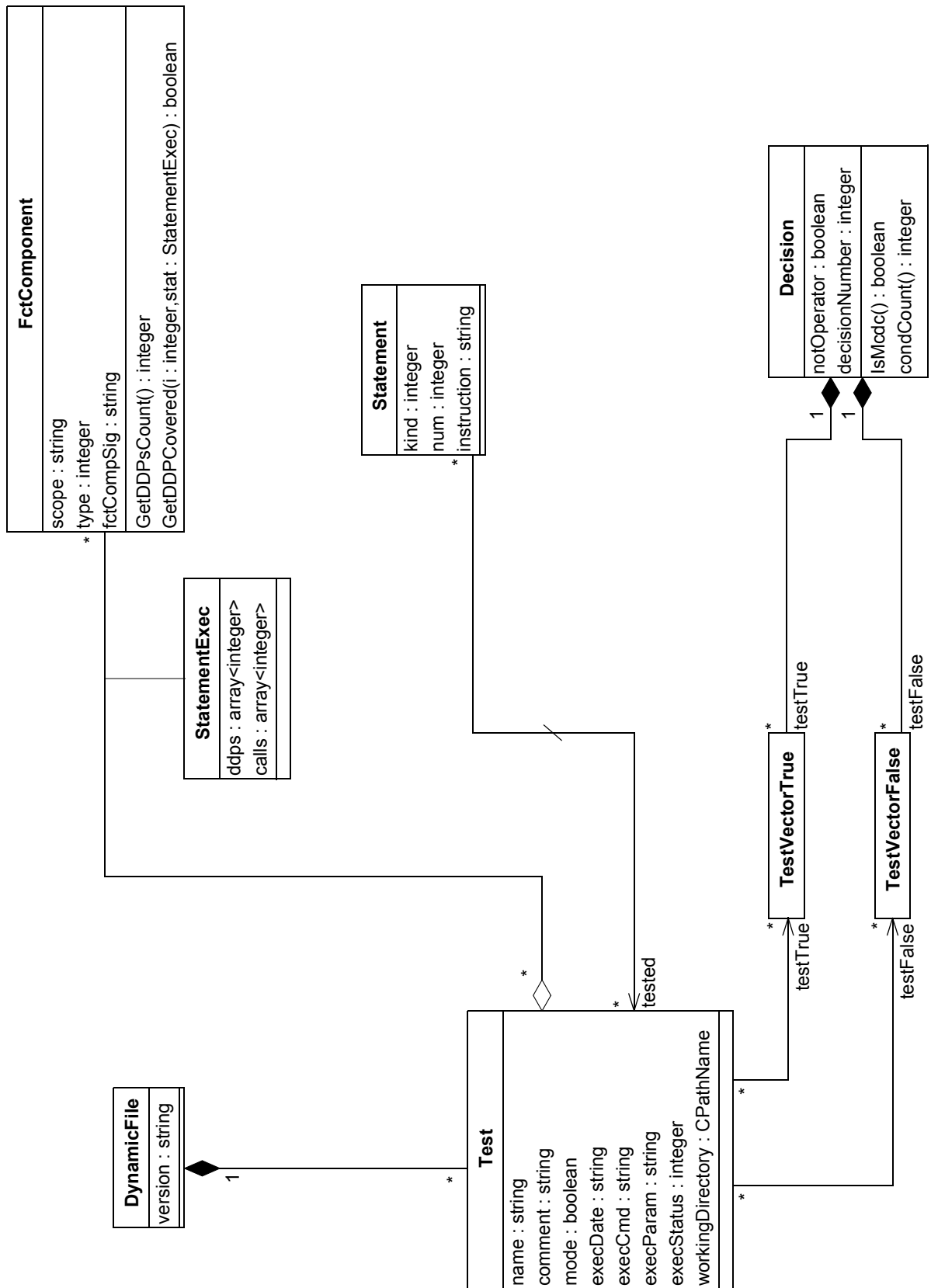
Figure 17 Dynamic diagram

A.3.2 Text Representation

The data model is shown class by class. Classes appear in alphabetical order.
For each class, existing associations, attributes, methods and heirs are listed in the following format:

class_name class

    Associations with:

        target_class_name   target_role number_instances_target_class

    Attributes:

        attribute_name : type

    Methods:

        method_name

    Heirs:

        class_name

AliasMetric class

    Associations with:

```
Language        language n
BasicMetric     basicMetric1
```

    Attributes:

        mnemonic : string

        scope: string

    Methods:

        LanguagePresent (language : Language) : boolean

        GetLanguageCount () : integer

ApiDomain class

Associations with:

```
Project        project 1
Component      componentn
```

Attributes:

code : string

line : integer

start : integer

length : integer

dirty : boolean

Methods:

MetricValuePresent (component : Component, metric : Metric, pfVal : void*): boolean

GetCriteriaCategory (component : Component, criteria : Criteria) : CriteriaCategory

GetFactorCategory (component : Component, factor : Factor) : FactorCategory

InvalidateContext ()

SetMetricValue (component : Component, metric : Metric, val : real) : boolean

GetComponentCount () : integer

ComponentPresent (component : Component) : boolean

Application class

Associations with:

```
Relation         to                n
Relation         from              n
ApiDomain        apiDomain         n
RuleViolation    ruleViolation     n
BasicMetric      basicMetric       n
Factor           factor            n
Criteria         criteria          n
Metric           metric            n
MetricValues     metricValues      1
CriteriaValues   criteriaValues    1
FactorValues     factorValues      1
Project          project           1
Language         language          1
QualityModel     qualityModel      1
SourceFile       sourceFile        1
```

Attributes:

code : string
line : integer
start : integer
length : integer
name : string
crc : integer
date : string
listEntries : *CstringList*

Methods:

GetMetricValue (metric : Metric) : real
EvaluateComputedMetrics ()
GetBasicMetricValue (basicMetric : BasicMetric) : real
GetCriteriaValue (criteria : Criteria) : CriteriaCategory
GetFactorValue (factor : Factor) : FactorCategory
EvaluateFactors ()
EvaluateCriterias ()
EvaluateQualityModel ()
IsExternal () : boolean
GetLanguageName () : LPCTSTR
IsClass () : boolean
IsFctComponent () : boolean
IsApplication () : boolean
IsModule () : boolean
IsFunction () : boolean

IsMethod () : boolean

MetricPresent (metric : Metric) : boolean

GetMetricCount () : integer

CriteriaPresent (criteria : Criteria) : boolean

GetCriteriaCount () : integer

FactorPresent (factor : Factor) : boolean

GetFactorCount () : integer

BasicMetricPresent (basicMetric : BasicMetric) : boolean

GetBasicMetricCount () : boolean

RuleViolationPresent (ruleViolation : RuleViolation) : boolean

GetRuleViolationCount () : integer

ApiDomainPresent (apiDomain : ApiDomain) : boolean

GetApiDomainCount () : integer

FromPresent (relation : Relation) : boolean

GetFromCount () : integer

ToPresent (relation : Relation) : boolean

GetToCount () : integer

ApplicationCriteria class

Associations with:

```
Component       component        n
BaseExpr        formula          1
QualityModel    qualityModel     1
CriteriaCategorycriteriaCategory n
```

Attributes:

name : string

comment : string

num : integer

Methods:

IsClassCriteria () : boolean

IsApplicationCriteria () : boolean

IsModuleCriteria () : boolean

IsFunctionCriteria () : boolean

GetCriteriaCategoryCount () : integer

CriteriaCategoryPresent (criteriaCategory : CriteriaCategory) : boolean

ComponentPresent (component : Component) : boolean

*GetComponentCount () : intege*r

ApplicationFactor class

Associations with:

Component	component	n
BaseExpr	formula	1
QualityModel	qualityModel	1
FactorCategory	factorCategory	n

Attributes:

name : string
comment : string
num : integer

Methods:

IsClassFactor () : boolean
IsApplicationFactor () : boolean
IsModuleFactor () : boolean
IsFunctionFactor () : boolean
GetFactorCategoryCount () : integer
FactorCategoryPresent (factorCategory : FactorCategory) : boolean
ComponentPresent (component : Component) : boolean
GetComponentCount () : integer

ApplicationMetric class

    Associations with:

```
BasicMetric     basicMetric1
Component       componentn
BaseExpr        formula  1
QualityModel    qualityModel1
```

    Attributes:

      name : string

      num : integer

      comment : string

      min : real

      max : real

      type : integer

      mnemonic : string

      visualMnemo : string

    Methods:

      IsClassMetric () : boolean

      IsApplicationMetric () : boolean

      IsModuleMetric () : boolean

      IsFunctionMetric () : boolean

      ComponentPresent (component : Component) : boolean

      GetComponentCount () : integer

BaseExpr class

    Associations with:

    Attributes:

    value : real

    Methods:

      Evaluate (component : Component) : real

      ResolveDependencies (qualityModel : QualityModel)

      CheckScope (qualityModel : QualityModel) : integer

      IsMetricExpr () : boolean

      IsIdentifierExpr () : boolean

      IsConstExpr () : boolean

      IsBinaryExpr () : boolean

      IsCriteriaExpr () : boolean

    Heirs:

      BinaryExpr

      ConstExpr

      IdentifierExpr

BasicMetric class

Associations with:

```
Language        language n
Component       componentn
AliasMetric     aliasMetricn
```

Attributes:

mnemonic : string

scope : integer

name : string

comment : string

index : integer

Methods:

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

AliasMetricPresent (aliasMetric : AliasMetric) : boolean

GetAliasMetricCount () : integer

LanguagePresent (language : Language) : boolean

GetLanguageCount () : integer

BasicRule class

Associations with:

```
Language        language n
```

Attributes:

mnemonic : string

name : string

comment : string

Methods:

LanguagePresent (language : Language) : boolean

GetLanguageCount () : integer

BinaryExpr class

Associations with:

```
IdentifierExpr  depends  n
BaseExpr        rightExpr1
BaseExpr        leftExpr 1
```

Attributes:

value : real

operator : string

Methods:

Evaluate (component : Component) : real

ResolveDependencies (qualityModel : QualityModel)

CheckScope (qualityModel : QualityModel) : integer

IsMetricExpr () : boolean

IsIdentifierExpr () : boolean

IsConstExpr () : boolean

IsBinaryExpr () : boolean

IsCriteriaExpr () : boolean

BooleanOperator class

Associations with:

```
Decision        next     1
Decision        previous 1
Attributes:
kind : integer
```


Call class

Associations with:

```
Component       from     1
Component       to       1
CallInstance    callInstancen
```

Methods:

IsOther () : boolean

IsFriend () : boolean

IsInheritance () : boolean

IsChild () : boolean

IsCall () : boolean

CallInstancePresent (callInstance : CallInstance) : boolean

GetCallInstanceCount () : integer

CallInstance class

Associations with:
```
Call            call      1
Statement       statement1
```
Attributes:
numCall : integer
line : integer

Child class

Associations with:
```
pComponent      from      1
Component       to        1
```
Methods:
IsOther () : boolean
IsFriend () : boolean
IsInheritance () : boolean
IsChild () : boolean
IsCall () : boolean

Class class

Associations with:
```
Relation        to        n
Relation        from      n
ApiDomain       apiDomainn
RuleViolation   ruleViolationn
BasicMetric     basicMetricn
Factor          factor    n
Criteria        criteria n
Metric          metric    n
MetricValues    metricValues1
CriteriaValues  criteriaValues1
FactorValues    factorValues1
Project         project  1
Language        language 1
QualityModel    qualityModel1
SourceFile      sourceFile1
```

```
Class          child     n
Class          parent    n
```

Attributes:

    code : string

    line : integer

    start : integer

    length : integer

    name : string

    crc : integer

    date : string

    listEntries : CstringList

    scope : string

    classSig : string

    type : integer

Methods:

    GetMetricValue (metric : Metric) : real

    EvaluateComputedMetrics ()

    GetBasicMetricValue (basicMetric : BasicMetric) : real

    GetCriteriaValue (criteria : Criteria) : CriteriaCategory

    GetFactorValue (factor : Factor) : FactorCategory

    EvaluateFactors ()

    EvaluateCriterias ()

    EvaluateQualityModel ()

    IsExternal () : boolean

    GetLanguageName () : LPCTSTR

    IsClass () : boolean

    IsFctComponent () : boolean

    IsApplication () : boolean

    IsModule () : boolean

    IsFunction () : boolean

    IsMethod () : boolean

    MetricPresent (metric : Metric) : boolean

    GetMetricCount () : integer

    CriteriaPresent (criteria : Criteria) : boolean

    GetCriteriaCount () : integer

    FactorPresent (factor : Factor) : boolean

    GetFactorCount () : integer

    BasicMetricPresent (basicMetric : BasicMetric) : boolean

    GetBasicMetricCount () : boolean

    RuleViolationPresent (ruleViolation : RuleViolation) : boolean

    GetRuleViolationCount () : integer

ApiDomainPresent (apiDomain : ApiDomain) : boolean
GetApiDomainCount () : integer
FromPresent (relation : Relation) : boolean
GetFromCount () : integer
ToPresent (relation : Relation) : boolean
GetToCount () : integer
ChildPresent (class : Class) : boolean
GetChildCount () : integer
ParentPresent (class : Class) : boolean
GetParentCount () : integer

ClassCriteria class

Associations with:

```
Component       componentn
BaseExpr        formula  1
QualityModel    qualityModel1
CriteriaCategorycriteriaCategoryn
```

Attributes:

name : string
comment : string
num : integer

Methods:

IsClassCriteria () : boolean
IsApplicationCriteria () : boolean
IsModuleCriteria () : boolean
IsFunctionCriteria () : boolean
GetCriteriaCategoryCount () : integer
CriteriaCategoryPresent (criteriaCategory : CriteriaCategory) : boolean
ComponentPresent (component : Component) : boolean
GetComponentCount () : integer

ClassFactor class

Associations with:

```
Component      componentn
BaseExpr       formula 1
QualityModel   qualityModel1
FactorCategory factorCategoryn
```

Attributes:

name : string

comment : string

num : integer

Methods:

IsClassFactor () : boolean

IsApplicationFactor () : boolean

IsModuleFactor () : boolean

IsFunctionFactor () : boolean

GetFactorCategoryCount () : integer

FactorCategoryPresent (factorCategory : FactorCategory) : boolean

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

ClassMetric class

Associations with:

```
BasicMetric     basicMetric1
Component       componentn
QualityModel    qualityModel1
BaseExpr        formula  1
```

Attributes:

name : string

num : integer

comment : string

min : real

max : real

type : integer

mnemonic : string

visualMnemo : string

Methods:

IsClassMetric () : boolean

IsApplicationMetric () : boolean

IsModuleMetric () : boolean

IsFunctionMetric () : boolean

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

Component class

Associations with:

```
Relation        to       n
Relation        from     n
ApiDomain       apiDomainn
RuleViolation   ruleViolationn
BasicMetric     basicMetricn
Factor          factor   n
Criteria        criteria n
Metric          metric   n
MetricValues    metricValues1
CriteriaValues  criteriaValues1
FactorValues    factorValues1
Project         project  1
Language        language 1
QualityModel    qualityModel1
SourceFile      sourceFile1
```

Attributes:

code : string

line : integer

start : integer

length : integer

name : string

crc : integer

date : string

listEntries : CstringList

Methods:

GetMetricValue (metric : Metric) : real

EvaluateComputedMetrics ()

GetBasicMetricValue (basicMetric : BasicMetric) : real

GetCriteriaValue (criteria : Criteria) : CriteriaCategory

GetFactorValue (factor : Factor) : FactorCategory

EvaluateFactors ()

EvaluateCriterias ()

EvaluateQualityModel ()

IsExternal () : boolean

GetLanguageName () : LPCTSTR

IsClass () : boolean

IsFctComponent () : boolean

IsApplication () : boolean

IsModule () : boolean

IsFunction () : boolean

IsMethod () : boolean

MetricPresent (metric : Metric) : boolean
GetMetricCount () : integer
CriteriaPresent (criteria : Criteria) : boolean
GetCriteriaCount () : integer
FactorPresent (factor : Factor) : boolean
GetFactorCount () : integer
BasicMetricPresent (basicMetric : BasicMetric) : boolean
GetBasicMetricCount () : boolean
RuleViolationPresent (ruleViolation : RuleViolation) : boolean
GetRuleViolationCount () : integer
ApiDomainPresent (apiDomain : ApiDomain) : boolean
GetApiDomainCount () : integer
FromPresent (relation : Relation) : boolean
GetFromCount () : integer
ToPresent (relation : Relation) : boolean
*GetToCount () : intege*r

Heirs:
Class
Application
FctComponent
Module

ConstExpr class

Attributes:
value : real
Methods:
Evaluate (component : Component) : real
ResolveDependencies (qualityModel : QualityModel)
CheckScope (qualityModel : QualityModel) : integer
IsMetricExpr () : boolean
IsIdentifierExpr () : boolean
IsConstExpr () : boolean
IsBinaryExpr () : boolean
IsCriteriaExpr () : boolean

Criteria class

Associations with:

```
Component      component      n
BaseExpr       formula        1
QualityModel   qualityModel   1
CriteriaCategorycriteriaCategory n
```

Attributes:

name : string

comment : string

num : integer

Methods:

IsClassCriteria () : boolean

IsApplicationCriteria () : boolean

IsModuleCriteria () : boolean

IsFunctionCriteria () : boolean

GetCriteriaCategoryCount () : integer

CriteriaCategoryPresent (criteriaCategory : CriteriaCategory) : boolean

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

Heirs:

ClassCriteria

ApplicationCriteria

FunctionCriteria

ModuleCriteria

CriteriaCategory class

Associations with:

```
Criteria       criteria 1
```

Attributes:

name : string

min : integer

max : integer

weight : integer

CriteriaExpr class

Associations with:
```
BinaryExpr       binaryExpr1
Criteria         criteria 1
```
Attributes:
value : real
mnemonic : string
byVal : boolean

Methods:
Evaluate (component : Component) : real
ResolveDependencies (qualityModel : QualityModel)
CheckScope (qualityModel : QualityModel) : integer
IsMetricExpr () : boolean
IsIdentifierExpr () : boolean
IsConstExpr () : boolean
IsBinaryExpr () : boolean
IsCriteriaExpr () : boolean

CriteriaValues class

Associations with:
```
Component        component1
```
Attributes:
values : array <ClgCriteriaCategory>*

Methods:
AddCriteriaValue (criteriaCategory : CriteriaCategory) : integer
GetCriteriaValue (index : integer) : CriteriaCategory

CtxtObject class

Attributes:
code : string
line : integer
start : integer
length : integer

Heirs :
RuleViolation
Component
Decision
ApiDomain
Statement

Decision class

    Associations with:

```
BooleanOperator next      1
BooleanOperator previous 1
Decision        up        1
Decision        down      1
Statement       statement1
TestVectorTrue  testTrue n
TestVectorFalse testFalsen
```

    Attributes:

       code : string

       line : integer

       start : integer

       length : integer

       notOperator : boolean

       decisionNumber : integer

    Methods:

       IsMcdc () : boolean

       CondCount () : integer

       TestTruePresent (testVectorTrue : TestVectorTrue) : boolean

       GetTestTrueCount () : integer

       TestFalsePresent (testVectorFalse : TestVectorFalse) : boolean

       GetTestFalseCount () : integer

DynamicFile class

    Associations with:

```
Test            test    n
```

    Attributes:

       fileName : string

       pathName : pathName

       version : string

       Methods:

       TestPresent (test : Test) : boolean

       GetTestCount () : integer

Factor class

Associations with:

```
Component       componentn
BaseExpr        formula  1
QualityModel    qualityModel1
FactorCategory  factorCategoryn
```

Attributes:

name : string
comment : string
num : integer

Methods:

IsClassFactor () : boolean
IsApplicationFactor () : boolean
IsModuleFactor () : boolean
IsFunctionFactor () : boolean
GetFactorCategoryCount () : integer
FactorCategoryPresent (factorCategory : FactorCategory) : boolean
ComponentPresent (component : Component) : boolean
GetComponentCount () : integer

Heirs:

ClassFactor
Applicationfactor
FunctionFactor
ModuleFactor

FactorCategory class

Associations with:

```
Factor          factor   1
```

Attributes:

name : string
min : integer
max : integer

Methods:

FactorValues class

Associations with:

Component   component1

Attributes:

values : array <ClgFactorCategory>*

Methods:

AddFactorValue (factorCategory : FactorCategory) : integer

GetFactorValue (index : integer) : FactorCategory

FctComponent class

Associations with:

```
Relation        to      n
Relation        from    n
ApiDomain       apiDomainn
RuleViolation   ruleViolationn
BasicMetric     basicMetricn
Factor          factor  n
Criteria        criteria n
Metric          metric  n
MetricValues    metricValues1
CriteriaValues  criteriaValues1
FactorValues    factorValues1
Project         project 1
Language        language 1
QualityModel    qualityModel1
SourceFile      sourceFile1
FctComponent    called  n
FctComponent    calling n
Statement       statementn
StatementExec   statementExecn
```

Attributes:

code : string
line : integer
start : integer
length : integer
name : string
crc : integer
date : string
listEntries : CstringList
scope : string
type : integer
fctCompSig : string

Methods:

GetMetricValue (metric : Metric) : real
EvaluateComputedMetrics ()
GetBasicMetricValue (basicMetric : BasicMetric) : real
GetCriteriaValue (criteria : Criteria) : CriteriaCategory
GetFactorValue (factor : Factor) : FactorCategory
EvaluateFactors ()
EvaluateCriterias ()
EvaluateQualityModel ()

IsExternal () : boolean
GetLanguageName () : LPCTSTR
IsClass () : boolean
IsFctComponent () : boolean
IsApplication () : boolean
IsModule () : boolean
IsFunction () : boolean
IsMethod () : boolean
MetricPresent (metric : Metric) : boolean
GetMetricCount () : integer
CriteriaPresent (criteria : Criteria) : boolean
GetCriteriaCount () : integer
FactorPresent (factor : Factor) : boolean
GetFactorCount () : integer
BasicMetricPresent (basicMetric : BasicMetric) : boolean
GetBasicMetricCount () : boolean
RuleViolationPresent (ruleViolation : RuleViolation) : boolean
GetRuleViolationCount () : integer
ApiDomainPresent (apiDomain : ApiDomain) : boolean
GetApiDomainCount () : integer
FromPresent (relation : Relation) : boolean
GetFromCount () : integer
ToPresent (relation : Relation) : boolean
GetToCount () : integer
GetDDPsCount () : integer
GetDDPCovered (i : integer, stat : StatementExec) : boolean
FindStatementIndex (statement : Statement) : integer
StatementPresent (statement : Statement) : boolean
GetStatementCount () : integer
CallingPresent (fctComponent : FctComponent) : boolean
GetCallingCount () : integer
CalledPresent (fctComponent : FctComponent) : boolean
GetCalledCount () : integer
StatementExecPresent (statementExec : StatementExec) : boolean
GetStatementExecCount () : integer

File class

Attributes:
fileName : string
pathName : pathName
Heirs:
StaticFile
SourceFile
DynamicFile
QualityModelFile

Friend class

Associations with:
```
Component        from      1
Component        to        1
```
Methods:
IsOther () : boolean
IsFriend () : boolean
IsInheritance () : boolean
IsChild () : boolean
IsCall () : boolean

Function class

Associations with:
```
Relation         to        n
Relation         from      n
ApiDomain        apiDomainn
RuleViolation    ruleViolationn
BasicMetric      basicMetricn
Factor           factor    n
Criteria         criteria n
Metric           metric    n
MetricValues     metricValues1
CriteriaValues   criteriaValues1
FactorValues     factorValues1
Project          project   1
Language         language 1
QualityModel     qualityModel1
```

```
SourceFile      sourceFile1
FctComponent    called    n
FctComponent    calling   n
Statement       statementn
StatementExec   statementExecn
```

Attributes:

code : string

line : integer

start : integer

length : integer

name : string

crc : integer

date : string

listEntries : CstringList

scope : string

type : integer

fctCompSig : string

Methods:

GetMetricValue (metric : Metric) : real

EvaluateComputedMetrics ()

GetBasicMetricValue (basicMetric : BasicMetric) : real

GetCriteriaValue (criteria : Criteria) : CriteriaCategory

GetFactorValue (factor : Factor) : FactorCategory

EvaluateFactors ()

EvaluateCriterias ()

EvaluateQualityModel ()

IsExternal () : boolean

GetLanguageName () : LPCTSTR

IsClass () : boolean

IsFctComponent () : boolean

IsApplication () : boolean

IsModule () : boolean

IsFunction () : boolean

IsMethod () : boolean

MetricPresent (metric : Metric) : boolean

GetMetricCount () : integer

CriteriaPresent (criteria : Criteria) : boolean

GetCriteriaCount () : integer

FactorPresent (factor : Factor) : boolean

GetFactorCount () : integer

BasicMetricPresent (basicMetric : BasicMetric) : boolean

GetBasicMetricCount () : boolean

RuleViolationPresent (ruleViolation : RuleViolation) : boolean
GetRuleViolationCount () : integer
ApiDomainPresent (apiDomain : ApiDomain) : boolean
GetApiDomainCount () : integer
FromPresent (relation : Relation) : boolean
GetFromCount () : integer
ToPresent (relation : Relation) : boolean
GetToCount () : integer
GetDDPsCount () : integer
GetDDPCovered (i : integer, stat : StatementExec) : boolean
FindStatementIndex (statement : Statement) : integer
StatementPresent (statement : Statement) : boolean
GetStatementCount () : integer
CallingPresent (fctComponent : FctComponent) : boolean
GetCallingCount () : integer
CalledPresent (fctComponent : FctComponent) : boolean
GetCalledCount () : integer
StatementExecPresent (statementExec : StatementExec) : boolean
GetStatementExecCount () : integer

FunctionCriteria class

Associations with:

```
Component       component       n
BaseExpr        formula         1
QualityModel    qualityModel    1
CriteriaCategorycriteriaCategory n
```

Attributes:

name : string
comment : string
num : integer

Methods:

IsClassCriteria () : boolean
IsApplicationCriteria () : boolean
IsModuleCriteria () : boolean
IsFunctionCriteria () : boolean
GetCriteriaCategoryCount () : integer
CriteriaCategoryPresent (criteriaCategory : CriteriaCategory) : boolean
ComponentPresent (component : Component) : boolean
GetComponentCount () : integer

FunctionFactor class

Associations with:

```
Component        componentn
BaseExpr         formula 1
QualityModel     qualityModel1
FactorCategory   factorCategoryn
```

Attributes:

name : string

comment : string

num : integer

Methods:

IsClassFactor () : boolean

IsApplicationFactor () : boolean

IsModuleFactor () : boolean

IsFunctionFactor () : boolean

GetFactorCategoryCount () : integer

FactorCategoryPresent (factorCategory : FactorCategory) : boolean

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

FunctionMetric class

Associations with:

```
BasicMetric      basicMetric1
Component        componentn
QualityModel     qualityModel1
BaseExpr         formula 1
```

Attributes:

name : string

num : integer

comment : string   min : real

max : real

type : integer

mnemonic : string

visualMnemo : string

Methods:

IsClassMetric () : boolean

IsApplicationMetric () : boolean

IsModuleMetric () : boolean

IsFunctionMetric () : boolean

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

IdentifierExpr class

Associations with:
```
BinaryExpr        binaryExpr1
```
Attributes:
value : real
mnemonic : string
byVal : boolean
Methods:
Evaluate (component : Component) : real
ResolveDependencies (qualityModel : QualityModel)
CheckScope (qualityModel : QualityModel) : integer
IsMetricExpr () : boolean
IsIdentifierExpr () : boolean
IsConstExpr () : boolean
IsBinaryExpr () : boolean
IsCriteriaExpr () : boolean
Heirs:
MetricExpr
CriteriaExpr

Inheritance class

Associations with:
```
Component        from      1
Component        to        1
```
Methods:
IsOther () : boolean
IsFriend () : boolean
IsInheritance () : boolean
IsChild () : boolean
IsCall () : boolean

Language class

Attributes:
name : string

LCSAJ class

Associations with:

```
Statement       jump    1
StmtCtrl        end     1
StmtCtrl        start   1
```

Method class

Associations with:

```
Relation        to       n
Relation        from     n
ApiDomain       apiDomain    n
RuleViolation   ruleViolation    n
BasicMetric     basicMetric  n
Factor          factor   n
Criteria        criteria n
Metric          metric   n
MetricValues    metricValues 1
CriteriaValues  criteriaValues   1
FactorValues    factorValues 1
Project         project  1
Language        language 1
QualityModel    qualityModel 1
SourceFile      sourceFile   1
FctComponent    called   n
FctComponent    calling  n
Statement       statement    n
StatementExec   statementExec    n
```

Attributes:

code : string
line : integer
start : integer
length : integer
name : string
crc : integer
date : string
listEntries : CstringList
scope : string
type : integer
fctCompSig : string
access : integer

Methods:

GetMetricValue (metric : Metric) : real
EvaluateComputedMetrics ()
GetBasicMetricValue (basicMetric : BasicMetric) : real
GetCriteriaValue (criteria : Criteria) : CriteriaCategory
GetFactorValue (factor : Factor) : FactorCategory
EvaluateFactors ()
EvaluateCriterias ()

EvaluateQualityModel ()
IsExternal () : boolean
GetLanguageName () : LPCTSTR
IsClass () : boolean
IsFctComponent () : boolean
IsApplication () : boolean
IsModule () : boolean
IsFunction () : boolean
IsMethod () : boolean
MetricPresent (metric : Metric) : boolean
GetMetricCount () : integer
CriteriaPresent (criteria : Criteria) : boolean
GetCriteriaCount () : integer
FactorPresent (factor : Factor) : boolean
GetFactorCount () : integer
BasicMetricPresent (basicMetric : BasicMetric) : boolean
GetBasicMetricCount () : boolean
RuleViolationPresent (ruleViolation : RuleViolation) : boolean
GetRuleViolationCount () : integer
ApiDomainPresent (apiDomain : ApiDomain) : boolean
GetApiDomainCount () : integer
FromPresent (relation : Relation) : boolean
GetFromCount () : integer
ToPresent (relation : Relation) : boolean
GetToCount () : integer
GetDDPsCount () : integer
GetDDPCovered (i : integer, stat : StatementExec) : boolean
FindStatementIndex (statement : Statement) : integer
StatementPresent (statement : Statement) : boolean
GetStatementCount () : integer
CallingPresent (fctComponent : FctComponent) : boolean
GetCallingCount () : integer
CalledPresent (fctComponent : FctComponent) : boolean
GetCalledCount () : integer
StatementExecPresent (statementExec : StatementExec) : boolean
GetStatementExecCount () : integer

Metric class

Associations with:

```
BasicMetric     basicMetric1
Component       componentn
QualityModel    qualityModel1
BaseExpr        formula  1
```

Attributes:

name : string

num : integer

comment : string

min : real

max : real

type : integer

mnemonic : string

visualMnemo : string

Methods:

IsClassMetric () : boolean

IsApplicationMetric () : boolean

IsModuleMetric () : boolean

IsFunctionMetric () : boolean

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

Heirs:

ClassMetric

ApplicationMetric

FunctionMetric

ModuleMetric

MetricExpr class

Associations with:

```
BinaryExpr      binaryExpr1
Metric          metric   1
```

Attributes:

value : real

mnemonic : string

byVal : boolean

Methods:

Evaluate (component : Component) : real

ResolveDependencies (qualityModel : QualityModel)

CheckScope (qualityModel : QualityModel) : integer

IsMetricExpr () : boolean

IsIdentifierExpr () : boolean

IsConstExpr () : boolean

IsBinaryExpr () : boolean

IsCriteriaExpr () : boolean

MetricValues class

Associations with:

```
Component       component1
```

Attributes:

values : array <real>

Module class

Associations with:

```
Relation        to              n
Relation        from            n
ApiDomain       apiDomain       n
RuleViolation   ruleViolation   n
BasicMetric     basicMetric     n
Factor          factor          n
Criteria        criteria        n
Metric          metric          n
MetricValues    metricValues    1
CriteriaValues  criteriaValues  1
FactorValues    factorValues    1
Project         project         1
Language        language        1
QualityModel    qualityModel    1
SourceFile      sourceFile      1
```

Attributes:

code : string
line : integer
start : integer
length : integer
name : string
crc : integer
date : string
listEntries : *CstringList*

Methods:

GetMetricValue (metric : Metric) : real
EvaluateComputedMetrics ()
GetBasicMetricValue (basicMetric : BasicMetric) : real
GetCriteriaValue (criteria : Criteria) : CriteriaCategory
GetFactorValue (factor : Factor) : FactorCategory
EvaluateFactors ()
EvaluateCriterias ()
EvaluateQualityModel ()
IsExternal () : boolean
GetLanguageName () : LPCTSTR
IsClass () : boolean
IsFctComponent () : boolean
IsApplication () : boolean
IsModule () : boolean
IsFunction () : boolean

IsMethod () : boolean

MetricPresent (metric : Metric) : boolean

GetMetricCount () : integer

CriteriaPresent (criteria : Criteria) : boolean

GetCriteriaCount () : integer

FactorPresent (factor : Factor) : boolean

GetFactorCount () : integer

BasicMetricPresent (basicMetric : BasicMetric) : boolean

GetBasicMetricCount () : boolean

RuleViolationPresent (ruleViolation : RuleViolation) : boolean

GetRuleViolationCount () : integer

ApiDomainPresent (apiDomain : ApiDomain) : boolean

GetApiDomainCount () : integer

FromPresent (relation : Relation) : boolean

GetFromCount () : integer

ToPresent (relation : Relation) : boolean

GetToCount () : integer

ModuleCriteria class

Associations with:

```
Component          component          n
BaseExpr           formula            1
QualityModel       qualityModel       1
CriteriaCategorycriteriaCategory n
```

Attributes:

name : string

comment : string

num : integer

Methods:

IsClassCriteria () : boolean

IsApplicationCriteria () : boolean

IsModuleCriteria () : boolean

IsFunctionCriteria () : boolean

GetCriteriaCategoryCount () : integer

CriteriaCategoryPresent (criteriaCategory : CriteriaCategory) : boolean

ComponentPresent (component : Component) : boolean

*GetComponentCount () : intege*r

ModuleFactor class

Associations with:

Component	component	n
BaseExpr	formula	1
QualityModel	qualityModel	1
FactorCategory	factorCategory	n

Attributes:

name : string

comment : string

num : integer

Methods:

IsClassFactor () : boolean

IsApplicationFactor () : boolean

IsModuleFactor () : boolean

IsFunctionFactor () : boolean

GetFactorCategoryCount () : integer

FactorCategoryPresent (factorCategory : FactorCategory) : boolean

ComponentPresent (component : Component) : boolean

GetComponentCount () : integer

ModuleMetric class

   Associations with:

```
BasicMetric      basicMetric1
Component        componentn
BaseExpr         formula  1
QualityModel     qualityModel1
```

   Attributes:

     name : string

     num : integer

     comment : string

     min : real

     max : real

     type : integer

     mnemonic : string

     visualMnemo : string

   Methods:

     IsClassMetric () : boolean

     IsApplicationMetric () : boolean

     IsModuleMetric () : boolean

     IsFunctionMetric () : boolean

     ComponentPresent (component : Component) : boolean

     GetComponentCount () : integer

Other class

   Associations with:

```
Component        from     1
Component        to       1
```

   Attributes:

     name : string

   Methods:

     IsOther () : boolean

     IsFriend () : boolean

     IsInheritance () : boolean

     IsChild () : boolean

     IsCall () : boolean

Project class

Associations with:

```
Rule            rule              n
Component       component         n
DynamicFile     dynamicFile       n
StaticFile      staticFile        n
SourceFile      sourceFile        n
ApiDomain       crtDomain         1
QualityModelFilequalityModelFile 1
```

Attributes:

fileName : string
pathName : pathname
creationDate : string
dataDirectory : CPathName
buildCmd : string
buildArgs : string
testDirectory : CpathName
binaryArgs : string
binary : string
binaryType : boolean
sourceDirectory : CPathName

Methods:

Load () : integer
Save () : integer
FindSourceFile (pathName : string) : SourceFile
EvaluateQualityModel ()
GetQualityModel () : QualityModel
FindStaticFile (pathName : string) : StaticFile
DynamicFilePresent (dynamicFile : DynamicFile) : boolean
GetDynamicFileCount () : integer
SourceFilePresent (sourceFile : SourceFile) : boolean
GetSourceFileCount () : integer
StaticFilePresent (staticFile : StaticFile) : boolean
GetStaticFileCount () : integer
ComponentPresent (component : Component) : boolean
GetComponentCount () : integer
RulePresent (rule : Rule) : boolean
GetRuleCount () : integer

QualityModel class

Associations with:
```
FunctionCriteriafunctionCriterian
ClassCriteria    classCriterian
ApplicationCriteriaapplicationCriterian
ModuleCriteria  moduleCriterian
FunctionFactor  functionFactorn
ClassFactor     classFactorn
ApplicationFactorapplicationFactorn
ModuleFactor    moduleFactorn
FunctionMetric  functionMetricn
ClassMetric     classMetricn
ApplicationMetricapplicationMetricn
ModuleMetric    moduleMetricn
QualityModelFilequalityModelFile1
Factor          factor                  n
Criteria        criteria                n
Metric          metric                  n
Language        language                n
```
Methods:

FindMetricIndex (metric : Metric) : integer

GetMetricCount () : integer

MetricPresent (metric : Metric) : boolean

GetCriteriaCount () : integer

CriteriaPresent (criteria : Criteria) : boolean

GetFactorCount () : integer

FactorPresent (factor : Factor) : boolean

GetLanguageCount () : integer

LanguagePresent (language : Language) : boolean

QualityModelFile class

Associations with:
```
QualityModel    qualityModeln
```
Attributes:

fileName : string

pathName : pathName

version : string

Methods:

FindQualityModel (language : Language) : QualityModel

QualityModelPresent (qualityModel : QualityModel) : boolean

Get QualityModelCount () : integer

ReferenceMetric class

Associations with:

```
Language        language n
```

Attributes :

mnemonic : string
scope : string

Methods :

LanguagePresent (language : Language) : boolean
GetLanguageCount () : integer

Heirs :

BasicMetric
AliasMetric

Relation class

Associations with:

```
Component       from    1
Component       to      1
```

Methods:

IsOther () : boolean
IsFriend () : boolean
IsInheritance () : boolean
IsChild () : boolean
IsCall () : booleanAssociations with:

Heirs:

Friend
Other
Inheritance
Child
Call

Rule class

Associations with:

```
RuleViolation    ruleViolationn
BasicRule        basicRule1
```

Attributes:

name : string
key : string
ruleState : string
Methods:
RuleViolationPresent (ruleViolation : RuleViolation) : boolean
Get RuleViolationCount () : integer

RuleViolation class

Associations with:

```
Component        component1
Rule             rule      1
SourceFile       sourceFile1
```

Attributes:

code : string
line : integer
start : integer
length : integer

SourceFile class

Associations with:

```
SourceFile      included n
SourceFile      includes n
Component       component  n
RuleViolation   ruleViolationn
Language        language 1
```

Attributes:

fileName : string
pathName : pathName
crc : integer
date : string
lineCount : integer

Methods:

IncludedPresent (sourceFile : SourceFile) : boolean
GetIncludedCount () : integer
IncludesPresent (sourceFile : SourceFile) : boolean
GetIncludesCount () : integer
RuleViolationPresent (ruleViolation : RuleViolation) : boolean
GetRuleViolationCount () : integer
ComponentPresent (component : Component) : boolean
GetComponentCount () : integer

Statement class

Associations with:

```
CallInstance    callInstancen
FctComponent    fctComponent1
Decision        decision n
StmtJump        stmtJump n
Branch          srcBranchn
Test            tested   n
```

Attributes:

code : string
line : integer
length : integer
kind : integer
num : integer
instruction : string

Methods:

DecisionPresent (decision : Decision) : boolean
GetDecisionCount () : integer
SrcBranchPresent (branch : Branch) : boolean
GetSrcBranchCount () : integer
StmtJumpPresent (stmtJump : StmtJump) : boolean
GetStmtJumpCount () : integer
TestedPresent (test : Test) : boolean
GetTestedCount () : integer
CallInstancePresent (callInstance : CallInstance) : boolean
GetCallInstanceCount () : integer
IsStmtCtrl () : boolean
IsStmtSequence () : boolean
IsStmtJump () : boolean

Heirs:

StmtCtrl
StmtSequence
StmtJump

StatementExec class

Associations with:

```
FctComponent    fctComponent    1
Test            test            1
```

Attributes:

ddps : array <integer>calls : array <integer>

StaticFile class

Associations with:

SourceFile   sourceFile   n

Attributes:

fileName : string

pathName : pathName

Methods:

SourceFilePresent (sourceFile : SourceFile) : boolean

GetSourceFileCount () : integer

StmtCtrl class

Associations with:

```
CallInstance    callInstance       n
FctComponent    fctComponent       1
Decision        decision           n
StmtJump        stmtJump           n
Branch          srcBranch          n
Test            tested             n
Branch          destinationBranch n
LCSAJ           lCSAJ              n
```

Attributes:

code : string

line : integer

start : integer

length : integer

kind : integer

num : integer

instruction : string

Methods:

DecisionPresent (decision : Decision) : boolean

GetDecisionCount () : integer

SrcBranchPresent (branch : Branch) : boolean

GetSrcBranchCount () : integer

StmtJumpPresent (stmtJump : StmtJump) : boolean

GetStmtJumpCount () : integer

TestedPresent (test : Test) : boolean

GetTestedCount () : integer

CallInstancePresent (callInstance : CallInstance) : boolean

GetCallInstanceCount () : integer

IsStmtCtrl () : boolean

IsStmtSequence () : boolean

IsStmtJump () : boolean
LCSAJPresent (lCSAJ : LCSAJ) : boolean
GetLCSAJCount () : integer
DestinationBranchPresent (branch : Branch) : boolean
GetDestinationBranchCount () : integer

StmtJump class

Associations with:

CallInstance	callInstance	n
FctComponent	fctComponent	1
Decision	decision	n
StmtJump	stmtJump	n
Branch	srcBranch	n
Test	tested	n
Statement	destination	1

Attributes:

code : string
line : integer
start : integer
length : integer
kind : integer
num : integer
instruction : string
label : string
instrJump : integer

Methods:

DecisionPresent (decision : Decision) : boolean
GetDecisionCount () : integer
SrcBranchPresent (branch : Branch) : boolean
GetSrcBranchCount () : integer
StmtJumpPresent (stmtJump : StmtJump) : boolean
GetStmtJumpCount () : integer
TestedPresent (test : Test) : boolean
GetTestedCount () : integer
CallInstancePresent (callInstance : CallInstance) : boolean
GetCallInstanceCount () : integer
IsStmtCtrl () : boolean
IsStmtSequence () : boolean
IsStmtJump () : boolean

StmtSequence class

Associations with:

```
CallInstance    callInstance    n
FctComponent    fctComponent    1
Decision        decision        n
StmtJump        stmtJump        n
Branch          srcBranch       n
Test            tested          n
```

Attributes:

code : string

line : integer

start : integer

length : integer

kind : integer

num : integer

instruction : string

stmtCount : integer

Methods:

DecisionPresent (decision : Decision) : boolean

GetDecisionCount () : integer

SrcBranchPresent (branch : Branch) : boolean

GetSrcBranchCount () : integer

StmtJumpPresent (stmtJump : StmtJump) : boolean

GetStmtJumpCount () : integer

TestedPresent (test : Test) : boolean

GetTestedCount () : integer

CallInstancePresent (callInstance : CallInstance) : boolean

GetCallInstanceCount () : integer

IsStmtCtrl () : boolean

IsStmtSequence () : boolean

IsStmtJump () : boolean

Test class

Associations with:

```
TestVectorTrue   testTrue       n
TestVectorFalse testFalse       n
DynamicFile     dynamicFile     1
StatementExec   statementExec   n
```

Attributes:
name : string
comment : string
mode : boolean
execDate : string
execCmd : string
execParam : string
execStatus : integer
workingDirectory : CPathName

Methods:
TestTruePresent (testVectorTrue : TestVectorTrue) : boolean
GetTestTrueCount () : integer
TestFalsePresent (testVectorFalse : TestVectorFalse) : boolean
GetTestFalseCount () : integer
StatementExecPresent (statementExec: StatementExec) : boolean
GetStatementExecCount () : integer

TestVector class

Attributes:
vector : string

Methods:
IsTestVectorTrue () : boolean
IsTestVectorFalse () : boolean

Heirs:
TestVectorTrue
TestVectorFalse

TestVectorFalse class

Associations with:
```
Decision        decision        1
```
Attributes:
vector : string

Methods:
IsTestVectorTrue () : boolean
IsTestVectorFalse () : boolean

TestVectorTrue class

Associations with:

    Decision   decision 1

Attributes:

vector : string

Methods:

IsTestVectorTrue () : boolean

IsTestVectorFalse () : boolean

A.4 Access commands

Access to the class attribute

Class*object*

Returns the name of the class of *object*. An error is reported if *object* is not a valid key.

Access to other attributes

Get *object attribute*

Returns the value of attributes of *object* designated by *attribute*. An error is reported if *attribute* does not designate an attribute of *object* or if *object* is not a valid key.

Access to a single cardinality role

GetRole *source_object target_role*

Applies to associations whose target class has cardinality 0 or 1(   ).

Returns the key of the object which has the *target_role* in one of the associations of *source_object*, or an empty string if there are no such associations. An error is reported if *source_object* has no association with *target_role* as a role.

Access to a multiple cardinality role

MapRole *source_object   target_role* -filter *fscript   script*

fscript and script represent a sequence of commands.

Applies to associations whose cardinality is greater than or equal to 0(   ).
It scans objects associated with the *source_object* which have *target_role* as a role.

For each object which is the *target_role* in one of the associations of *source_object*, the *fscript* command sequence is evaluated:

• if *fscript* returns a value greater than 0, the *script* sequence is evaluated,

• if *fscript* returns a value equal to 0, the *script* sequence is not evaluated.

If *fscript* is not present, *script* is always evaluated.

If *script* returns a value equal to 0, the MapRole command stops immediately.

At each evaluation, *fscript* and *script* receive as a parameter the identifier of the object to process.

The MapRole command returns the number of times *script* has been evaluated. This number represents the overall number of objects which have *target_role* as a role in one of the associations of *source_object* or, if a filter is specified, it represents the number of

objects that match the filtering condition. An error is reported:

- if *source_object* has no association with, as a role, a target object: *target_role*,

- if *fscript* and *script* end with an uncontrolled value.

A.5 Debugging aid commands

Roles of a class

- Roles Of *object*

Returns the role list for the class of which *object* is an instance.

Attributes of a class

- Attributes Of *object*

Returns the attribute list for the class of which *object* is an instance.

Telelogic Logiscope

Appendix B

Using Logiscope on UNIX

This appendix presents some UNIX specific behaviors and utilities.

B.1   Printer Settings

You can easily add a new printer or change your default printer through the **mwprinter** utility, which launches a printer applet similar to the Windows printers one.

The **Printers** applet includes a default generic postscript printer command mapping that satisfies most printing needs. The PRINTER environment variable should be set to your default printer name.

If you need to add other printer command mappings, you can use the **Add Printer** wizard described below.

Changes you make using the **Printers** applet have an impact on the applications running on the same machine ; they become effective after saving. Applications running on other machines must be restarted for the changes to be effective.

To add a printer using the **Add Printer** wizard:

1. Launch the Visual MainWin Control Panel.
2. Double-click **Printers**. The **MainWin Printers** applet opens.
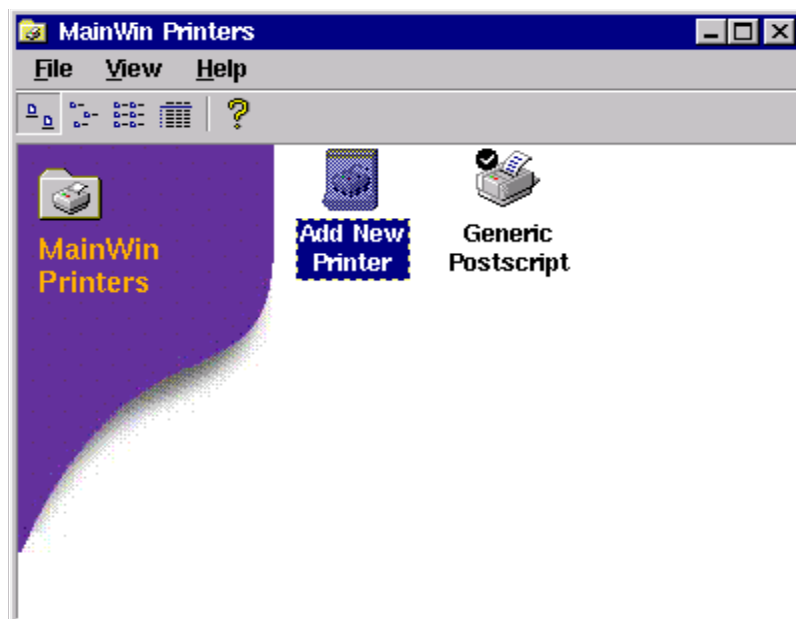
3. Double-click **Add New Printer** in the **MainWin Printers** applet. The **Add Printe**r **Wizard** starts.
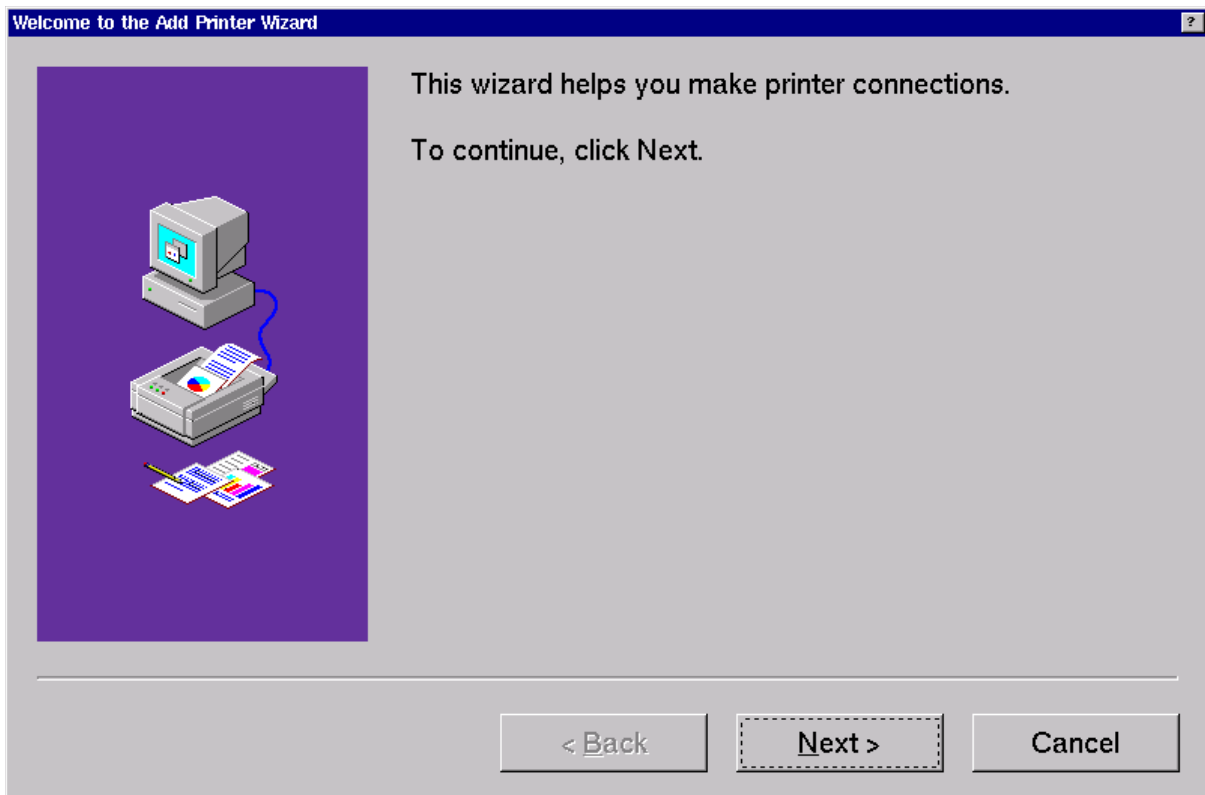
4.   Click **Next**.
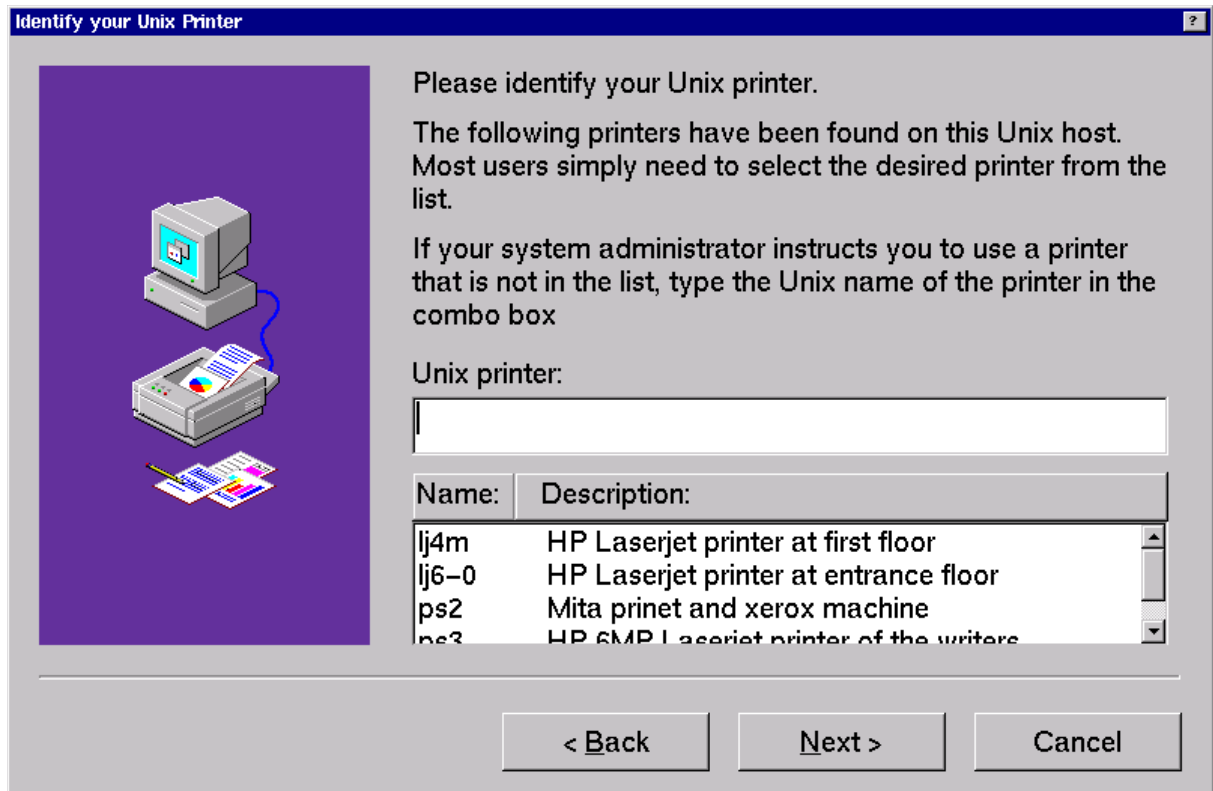
5.   Specify the UNIX printer you want to use.

A list of printers defined on the UNIX host appears at the bottom of the wizard. If the printer you want to add is included in the list, select it. To add a printer that is not in the list, enter the UNIX name of the printer in the UNIX printer text field.

Note: The list of printers is extracted from the /etc/printcap file, which contains the list of all printers on the network to which your UNIX host has access. The printcap file is used on all UNIX platforms as the central location for specifying printers. It is configured and maintained by the system administrator.

6. Click **Next**.
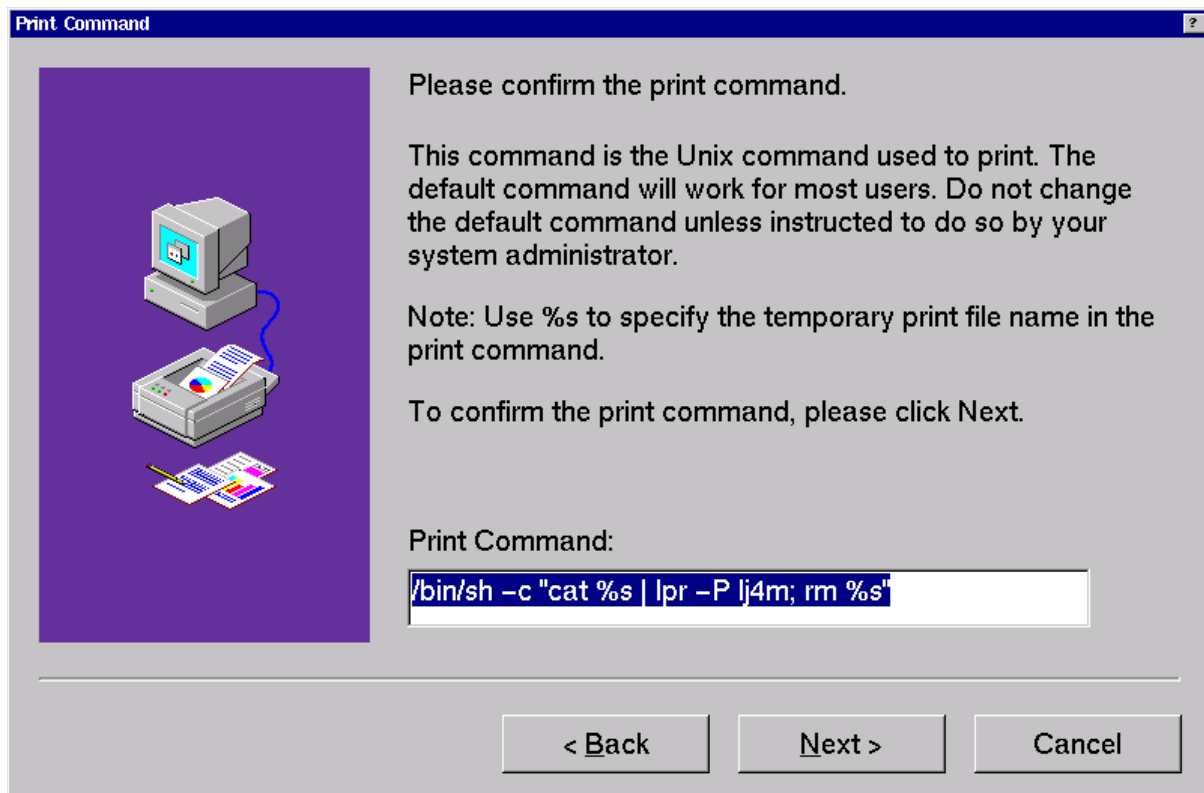
The default print command for the selected printer appears in the **Print Command** text field. Usually, the default command is the one you should use. You should not modify the command unless your system administrator instructs you to do so.

7. Click **Next**.
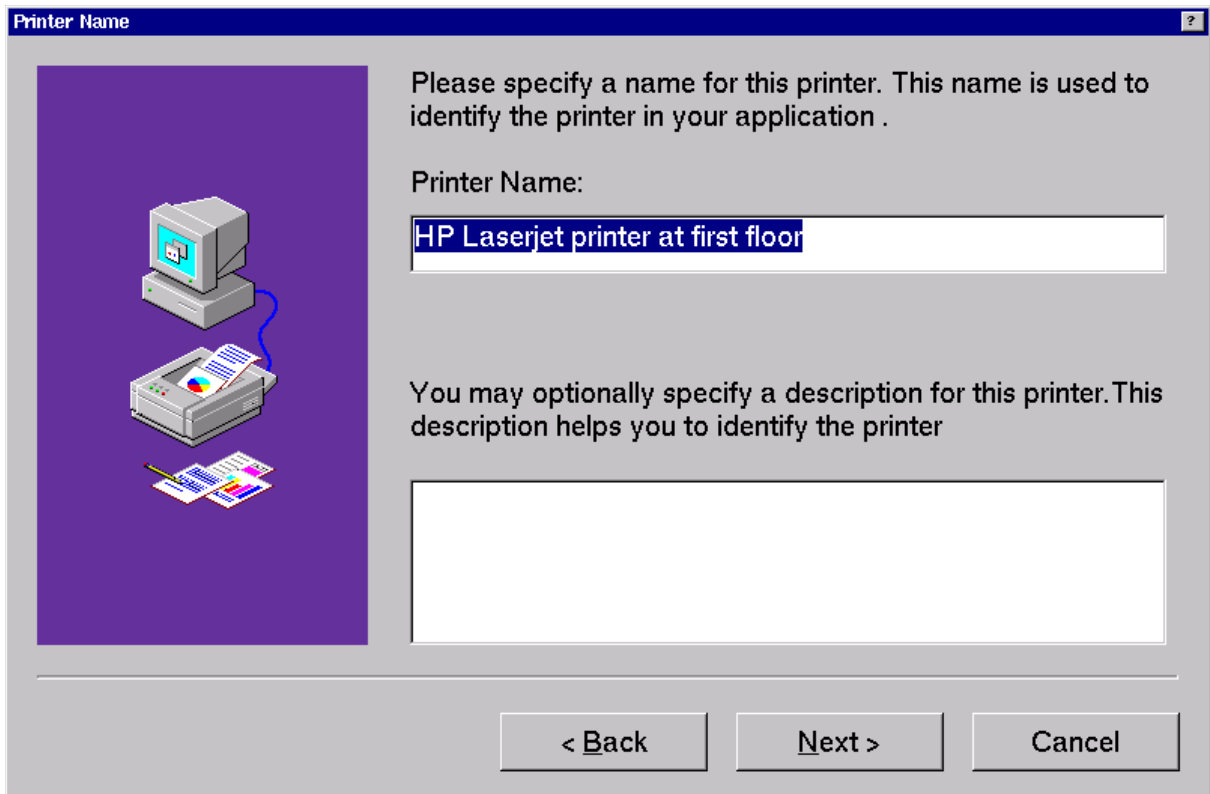
The **Printer Name** is used by the application when displaying available printers in the **Print** dialog. You may modify the default name if you wish.

The printer description has the same functionality as on Microsoft Windows. This field may be left blank.

8. Click **Next**.
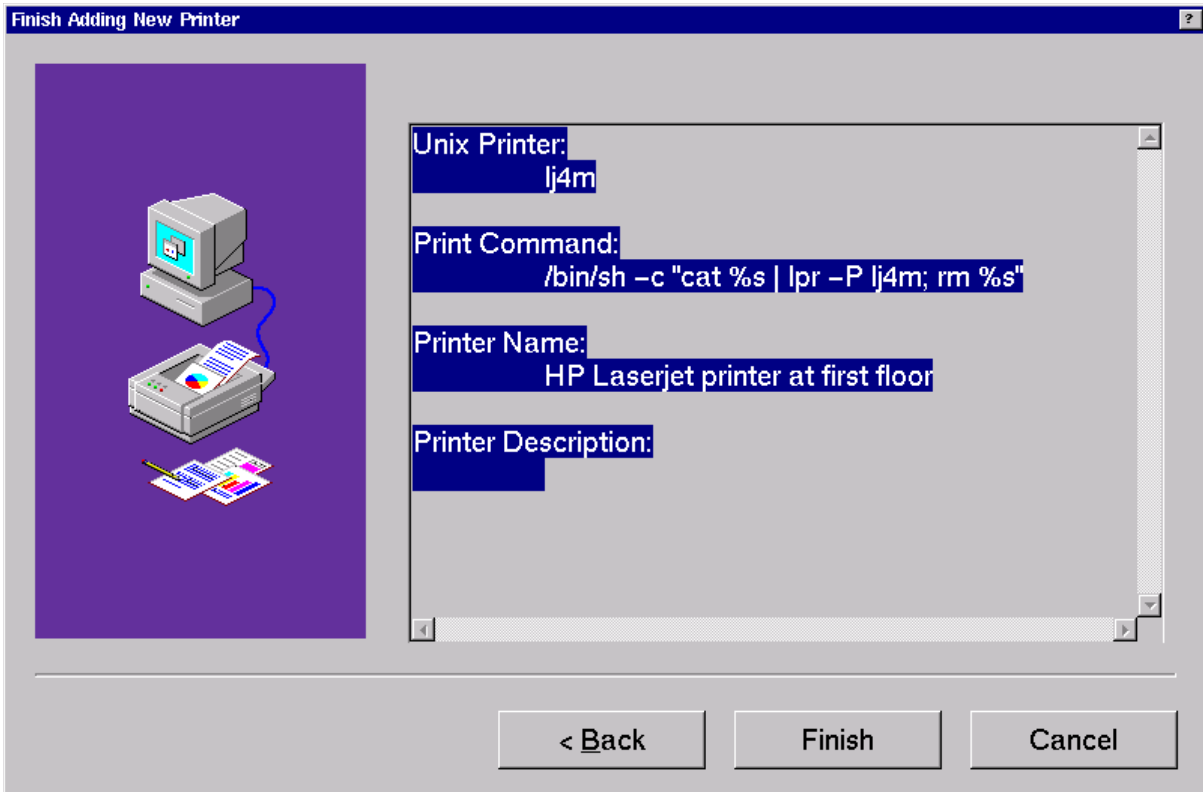
9. Make sure the information you specified in the **Add Printer** wizard is correct. If you are satisfied with the settings, click **Finish**. Otherwise, click **Back** to change settings.

The printer you added now appears in the printers list.
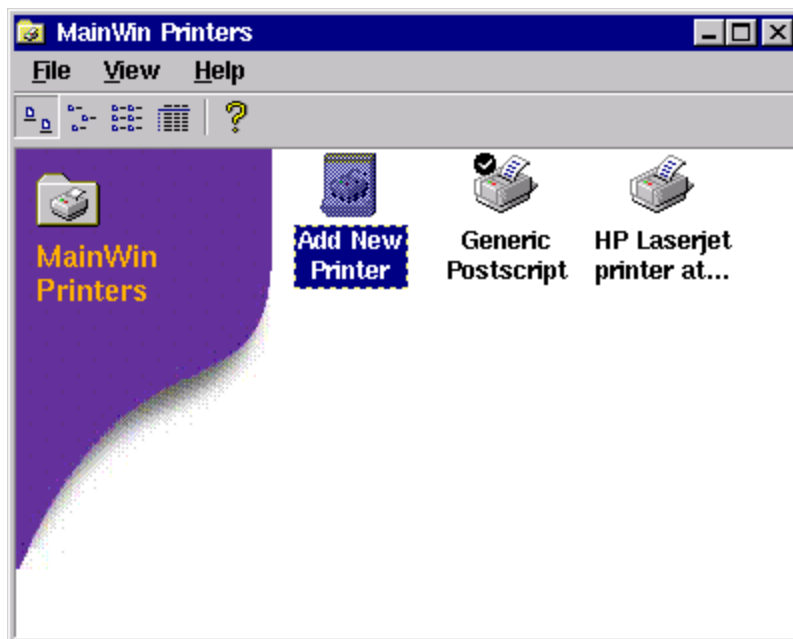
B.2   Docking bar

The floating bar behaves differently in Windows NT and in MainWin.

Behavior on Windows NT	Behavior on MainWin
If you double-click the title bar of a floating toolbar window, the toolbar returns to its original position on the application window, below the menu bar.	Double-clicking the title bar of a floating toolbar window has no effect. For the toolbar to return to its location on the application window, click the system menu in the top left corner of the toolbar window and select **Redock.**
When dragging a floating toolbar window over the toolbar area of the application window, a rectangle appears to indicate the size and location of the toolbar that should be dropped to that position.	When dragging a floating toolbar window over the toolbar area of the application window, no rectangle appears to indicate the size and location of the toolbar on the application window. Instead, the floating toolbar is dragged. When dropped, the toolbar is correctly set in the application window.

B.3   Help

The online help can be accessed outside Studio through the following command:

<logiscope_install>/bin/hh *<logiscope_install>*/bin/lgvcs.chm

Telelogic Logiscope

Appendix C

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send written license inquiries to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send written inquiries to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software
IBM Corporation
1 Rogers Street
Cambridge, Massachusetts 02142
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, Telelogic, Telelogic Synergy, Telelogic Change, Telelogic DOORS, Telelogic Tau, Telelogic DocExpress, Telelogic Rhapsody, Telelogic Statemate, and Telelogic System Architect are trademarks or registered trademarks of International Business Machine Corporation in the United States, other countries, or both, are trademarks of Telelogic, an IBM Company, in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at:

www.ibm.com/legal/copytrade.html.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, FrameMaker, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain jurisdictions.

AIX and Informix are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

HP and HP-UX are registered trademarks of Hewlett-Packard Corporation.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Macrovision and FLEXnet are registered trademarks or trademarks of Macrovision Corporation.

Microsoft, Windows, Windows 2003, Windows XP, Windows Vista and/or other Microsoft products referenced herein are either trademarks or registered trademarks of Microsoft Corporation.

Netscape and Netscape Enterprise Server are registered trademarks of Netscape Communications Corporation in the United States and other countries.

Sun, Sun Microsystems, Solaris, and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Pentium is a trademark of Intel Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product or service names may be trademarks or service marks of others.

Index

Telelogic