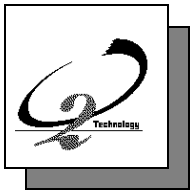


O_2 Tools
User Manual

Release 5.0 - April 1998

Information in this document is subject to change without notice and should not be construed as a commitment by O_2 Technology.

The software described in this document is delivered under a license or nondisclosure agreement.

The software can only be used or copied in accordance with the terms of the agreement. It is against the law to copy this software to magnetic tape, disk, or any other medium for any purpose other than the purchaser's own use.

O_2, O_2Engine API, O_2C, O_2DBAccess, O_2Engine, O_2Graph, O_2Kit, O_2Look, O_2Store, O_2Tools, and O_2Web are registered trademarks of O_2 Technology.

SQL and AIX are registered trademarks of International Business Machines Corporation.

Sun, SunOS, and SOLARIS are registered trademarks of Sun Microsystems, Inc.

X Window System is a registered trademark of the Massachusetts Institute of Technology.

Unix is a registered trademark of Unix System Laboratories, Inc.

HPUX is a registered trademark of Hewlett-Packard Company.

BOSX is a registered trademark of Bull S.A.

IRIX is a registered trademark of Siemens Nixdorf, A.G.

NeXTStep is a registered trademark of the NeXT Computer, Inc.

Purify, Quantify are registered trademarks of Pure Software Inc.

Windows is a registered trademark of Microsoft Corporation.

All other company or product names quoted are trademarks or registered trademarks of their respective trademark holders.

Who should read this manual

This manual presents O_2Tools, a complete graphical programming environment for the design and development of O_2 applications. It describes the browsers and editors available, as well as how to customize O_2Tools screens.

Other documents available are outlined, click below.

See O2 Documentation set.

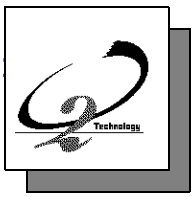


TABLE OF CONTENTS

This manual is divided into the following chapters:

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1 Introduction

Congratulations! You are now a user of O_2Tools - the complete graphical programming environment for the design and development of object-oriented database applications.

O_2Tools has powerful and user-friendly tools enabling you to drastically cut down development time and increase your productivity as a software designer and developer.

This introductory chapter is divided as follows:

1 Introduction

1.1 System Overview

The system architecture of O_2 is illustrated in Figure 1.1.

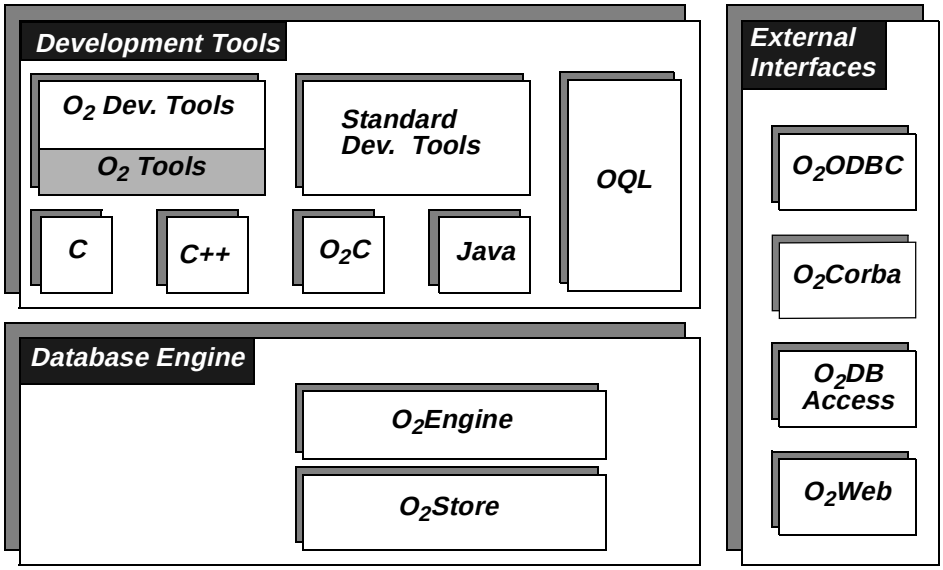


Figure 1.1: O_2 System Architecture

The O_2 system can be viewed as consisting of three components. The *Database Engine* provides all the features of a Database system and an object-oriented system. This engine is accessed with *Development Tools*, such as various programming languages, O_2 development tools and any standard development tool. Numerous *External Interfaces* are provided. All encompassing, O_2 is a versatile, portable, distributed, high-performance dynamic object-oriented database system.

Database Engine:

- O_2Store — The database management system provides low level facilities, through O_2Store API, to access and manage a database: disk volumes, files, records, indices and transactions.
- O_2Engine — The object database engine provides direct control of schemas, classes, objects and transactions, through O_2Engine API. It provides full text indexing and search capabilities with O_2Search and spatial indexing and retrieval capabilities with O_2Spatial. It includes a Notification manager for informing other clients connected to the same O_2 server that an event has occurred, a Version manager for handling multiple object versions and a Replication API for synchronizing multiple copies of an O2 system.

Programming Languages:

O_2 objects may be created and managed using the following programming languages, utilizing all the features available with O_2 (persistence, collection management, transaction management, OQL queries, etc.)

- C — O_2 functions can be invoked by C programs.
- C++ — ODMG compliant C++ binding.
- Java — ODMG compliant Java binding.
- O_2C — A powerful and elegant object-oriented fourth generation language specialized for easy development of object database applications.
- OQL — ODMG standard, easy-to-use SQL-like object query language with special features for dealing with complex O_2 objects and methods.

O_2 Development Tools:

- O_2Graph — Create, modify and edit any type of object graph.
- O_2Look — Design and develop graphical user interfaces, provides interactive manipulation of complex and multimedia objects.
- O_2Kit — Library of predefined classes and methods for faster development of user applications.
- O_2Tools — Complete graphical programming environment to design and develop O_2 database applications.

Standard Development Tools:

All standard programming languages can be used with standard environments (e.g. Visual C++, Sun Sparcworks).

External Interfaces:

- O_2Corba — Create an O_2/ Orbix server to access an O_2 database with CORBA.
- O_2DBAccess — Connect O_2 applications to relational databases on remote hosts and invoke SQL statements.
- O_2ODBC — Connect remote ODBC client applications to O_2 databases.
- O_2Web — Create an O_2 World Wide Web server to access an O_2 database through the internet network.

1.2 Features

Application programming in O_2 consists of defining a schema and writing, editing, compiling and debugging methods and programs.

O_2Tools provides you with graphical displays and direct manipulation of objects representing applications, programs, classes and methods. All application sources are stored in the database giving you the necessary data security, data sharing and concurrency management for the entire development team.

Implemented in O_2C and using O_2Look as its user interface, O_2Tools is itself an O_2 application and can be regarded as the O_2C graphical programming environment.

O_2Tools is aimed at two different types of user

- the software developer who uses O_2Tools to browse, edit and query the database and schema as well as edit, test and debug methods and programs.

- the end-user who uses O_2Tools to browse, query and edit the database and schema.

Integrated tools

O_2Tools provides a set of fully integrated tools essential during the design and the development of object oriented applications.

Graphical browsers enable you to visualize and implement object-oriented concepts such as inheritance, encapsulation and overloading.

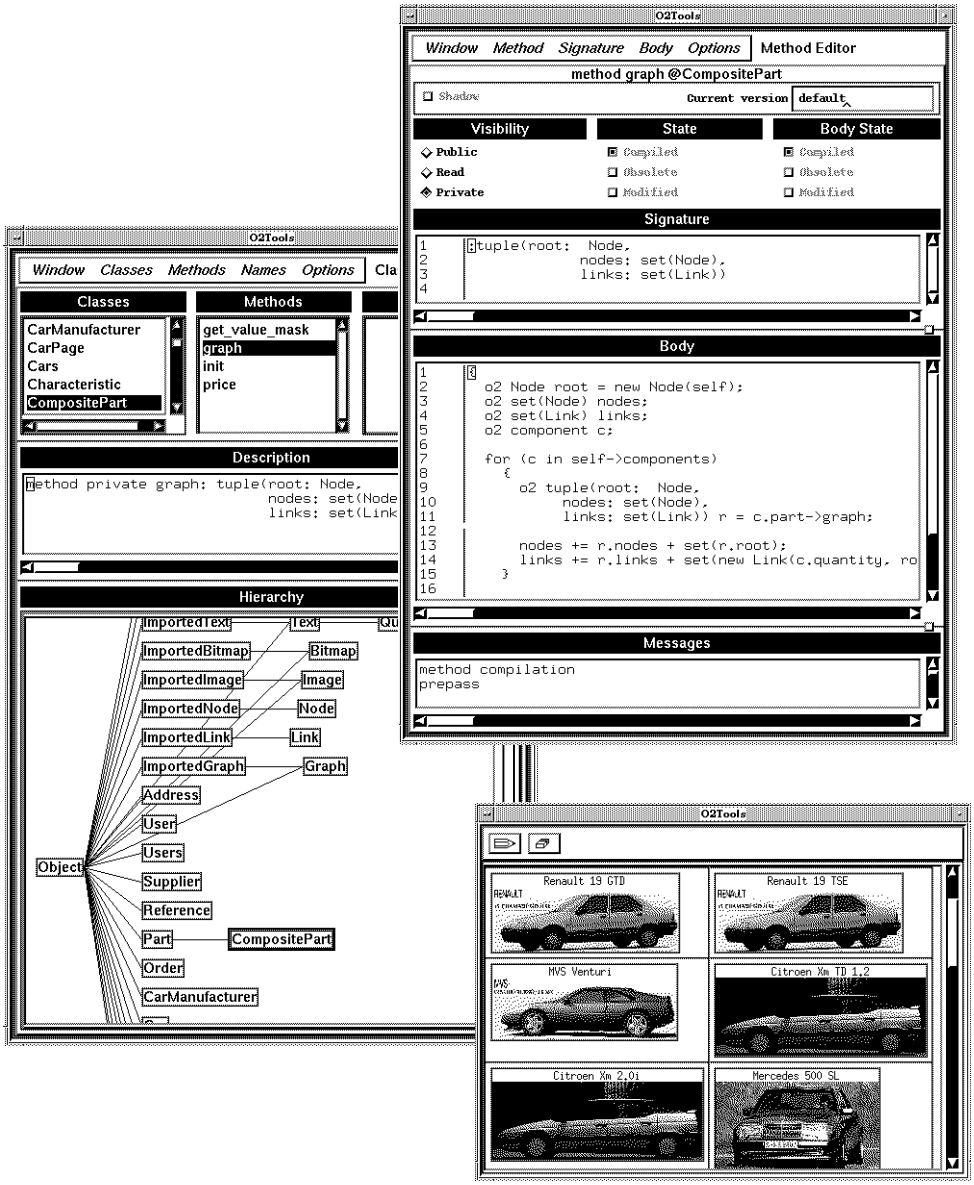
Graphical editors to edit and compile both schema and data.

A full page editor, called O_2Shell, to run alphanumeric commands and O_2 queries.

A cross reference manager and an automatic recompilation tool to write, test, debug and maintain applications in an efficient way and to improve code readability.

Increased productivity

O_2Tools greatly reduces the programming design and development cycle by allowing incrementally designed schemas, running partially written code, incremental compiling and dynamic loading of O_2C code and interactive testing of methods and programs.



Software requirements

To run O_2Tools the following software environment is required

- X windows - release X11 or later.

- O_2Engine

- O_2C

- O_2Look

- 4.5 Mb RAM.

1.3 User Manual overview

This user manual is divided into the following chapters:

- Chapter1 ***- Introduction***

  This is an introductory chapter that outlines the O_2 system and the O_2Tools product architecture and details how to launch O_2Tools in order to start browsing or programming. It also includes other operations such as Help, Lock, Commit, Abort and Alpha.

- Chapter 2 ***- Browsers***

  This chapter is aimed at all types of O_2Tools user and describes how to use all the various browsers available with O_2Tools in order to browse, query and edit the database.

- Chapter 3 ***- Programming environment***

  This chapter is aimed more at software developers wishing to actually program applications using O_2Tools.

  It describes the O_2Tools Programming environment: editing and compiling sources of classes, methods, programs, application variables, functions, persistent types and persistent names. Also detailed are the O_2Shell editor, dependency management, aborting and validating a transaction.

- Chapter 4 ***- How to customize O_2Tools.***

All the displays of O_2Tools can be customized to your specific requirements. This chapter details the graphic resources you need: dashboard, schema, application, class, function, persistent type, persistent name, version browsers and source editors.

How to read this manual

The *O_2Tools User Manual* describes a graphical tool for developing O_2 database applications. You should read this manual in conjunction with the *O_2C Reference Manual* which provides the semantics for the commands presented in this manual. As well, the *O_2C Beginners Guide* provides an overall presentation of object-oriended programming, the O_2 database system and step by step examples of how to develop database applications with O_2.

Important

You should refer to the *O2C Reference Manual* at all times

1.4 Getting Started

To launch O_2Tools type:

```
o2tools -system <system_name> -server <server_name>
```

The O2HOME environment variable is mandatory and must be set before lauching O_2Tools (or any O_2 program). This variable contains the name of the O_2 installation directory.

The options recognized by O_2Tools are the following:

-version : Display the O_2 version number.

-help : Display a help page.

-env : Display the current option values.

-verbose : Enable the verbose mode.

-system <*system_name*> : Indicates the name of the O_2 system to use.

-server <*server_name*> : Indicates the name of the host server to use.

-libs <*lib1:lib2:...:libn*> : Indicates the dynamic libraries used to execute O_2 code.

-*libpath* <*path1:path2:...:pathn*> : Indicates the search paths for libraries specified in -libs.

Options to O_2Tools can also be given in configuration files. These files are named .o2rc and are located in the O_2 distribution directory and in the user's home directories. Information for all users should be placed in the $O2HOME/ .o2rc file, and information specific to a given user should be placed in the user's home directories. Users can also redefine a global option in their personal .o2rc file.

In these files, some options can be prefixed by a system name for which an option value must be given. If an option value is specified alone and also prefixed with a system name, the value without the prefix is used for all systems except the systme that has a specific option value.

The different entries of the configuration file for O_2Tools are:

system-name = sysname

Defines a system name used by default when running O_2Tools.

[sysname.]server = servname

Defines the name of the machine on which the o2 server process is running for the system named sysname.

[sysname.]swapdir = dir

Defines the file directory to be used by O_2Tools for alternate swap files. If not defined, the swap subdirectory of the O_2 installation directory is used.

[sysname.]swapsize = size

size is a positive decimal integer representing the triggering size in kilobytes beyond which O_2 uses alternative swap files instead of the standard Unix swap files. If not defined, the default value is 4000.

[**sysname.**]**libs = lib1:lib2:...:libn**

The values given in **lib1:lib2:...:libn** cause the O_2C compiler to look for external functions in the libraries given. The libraries are sought first in any directories provided in libpath options prior to the standard library directories.

[**sysname.**]**libpath = path1:path2:...:pathn**

The values given in **{path1:path2:...:pathn}** cause the O_2C compiler to look for libraries in the directories given before searching in the standard library directories / lib, / usr/ lib and / usr/ local/ lib for Unix.

+/-[sysname.]verbose

Causes O_2Tools to display warnings and messages.

[sysname.]cachesize = size

Specifies the size of the database buffer for O_2Tools.

The text between brackets [] is optional. If used, the value entered applies for all systems available except those for which a specific value is specified with a prefix.

EXAMPLES

If the .**o2rc** file is :

```
cachesize = 6000

server = mach1

sys1.server = mach1

sys1.cachesize = 5000

+sys1.verbose

sys2.server = mach2

sys3.cachesize = 10000
```

The cache size for all systems is 6 megabytes except for sys1 : 5 Megabytes and sys3 : 10 Megabytes.

By default, the machine running o2server is mach1 except for sys2: mach2.

Options for O_2Tools can also be specified using the O2OPTIONS environment variable. This variable must contain a string with the same syntax of the command line.

For example,

```
(with csh): setenv O2OPTIONS "-system sys1 -server mach3"
```

(with sh): set O2OPTIONS "-system sys1 -server mach3"; export O2OPTIONS

Values with the O2OPTIONS environment variable override values retrieved from .o2rc files.

Values given in the O_2Tools command line override values given with O2OPTIONS variable and .o2rc files.

O_2Tools starts up and displays the dashboard illustrated in Figure 1.2 below.

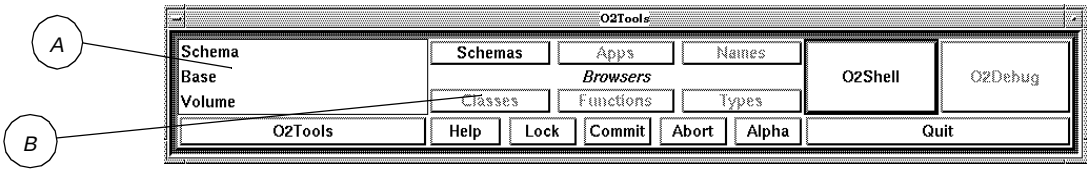


Figure 1.2: O_2Tools dashboard

The dashboard is composed of the following elements:

An information section (*A*) on the left hand side of the dashboard tells you the name of the current working schema, the current base and the current volume. When the dashboard first appears, it is possible that this part is not filled in because no schema and base has been set.

The Browser panel (*B*) in the center of the dashboard has several buttons to trigger the Schema Browser, the Application Browser, the Name Browser, the Class Browser, the Function Browser and the Type Browser.

The O_2Shell button displays the full page graphical editor to carry out alphanumeric commands and O_2 queries.

The O_2Debug button triggers the debugger. (The graphical debugger is not implemented at the time of printing).

The O_2Tools button displays a dialog box shown in Figure 1.3 with which you can save the current configuration of your O_2Tools sessions into a configuration file called.o2toolsrc or set the global options of O_2Tools.

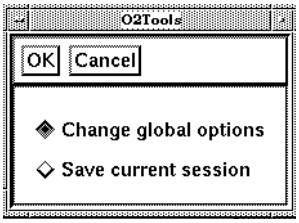


Figure 1.3: O_2Tools dialog box

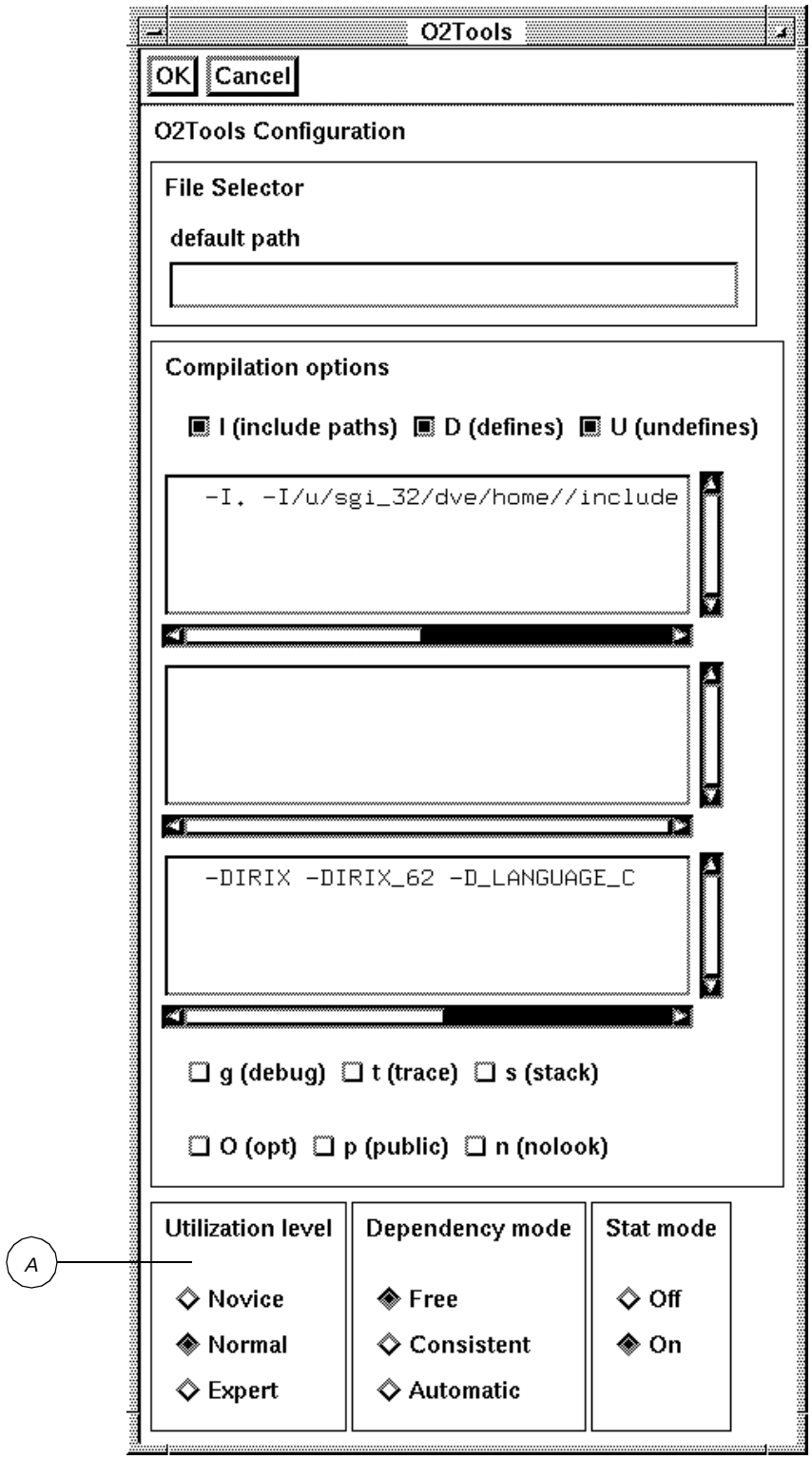


Figure 1.4: Global options

If you click on change global options and then on OK you see the window in Figure 1.4. How to use all these configuration options is described in Section 3.13 but you should know that there are various utilization levels (*A*). Novice indicates confirmation is required each time you carry out dangerous commands such as Delete. Normal means confirmation is only asked for when you use the abort, alpha or quit command. At Expert level none of these confirmations are required.

Important

By default the level is set to Normal but if you really have not used O_2Tools before we strongly recommend that you change the level to Novice until you are completely sure of O_2Tools. All dialog boxes are described in this manual.

At the beginning of each O_2Tools session only the Schema Browser and the O_2Shell can be activated as you must set the schema and base you wish to work on before doing anything else (see Chapter 2). However, this is not the case if you use the configuration file.o2toolsrc to preset your configuration. See Section 4.1.

1.5 Other operations

There a number of other operations you carry out using O_2Tools via the Dashboard: Help, Lock, Commit, Abort and Alpha.

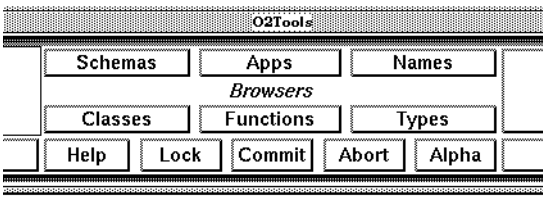


Figure 1.5: Other Dashboard buttons

Help

To obtain information about O_2Tools, click on the Help button. The dialog box shown in (*A*) Figure 1.6 appears.

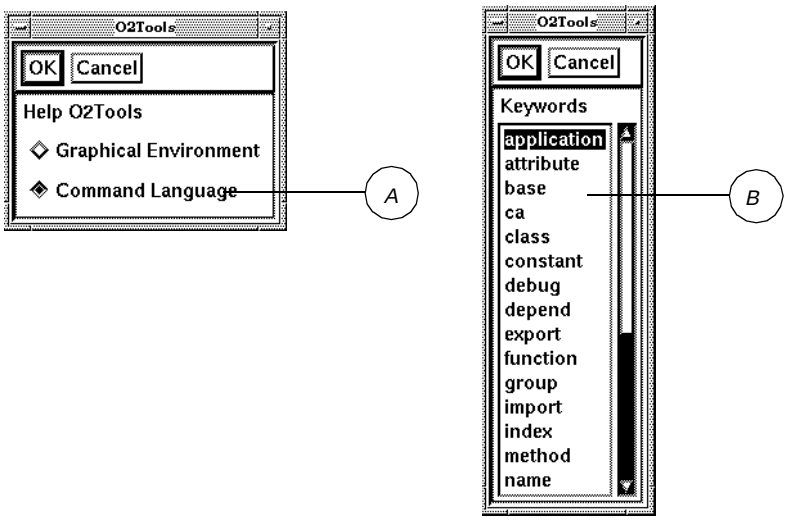


Figure 1.6: Help dialog box

If you now click on Command Language and then on OK, the list of keywords shown in (*B*) Figure 1.6 appears.

If you doubleclick on a name or on OK, information is then displayed in a separate window.

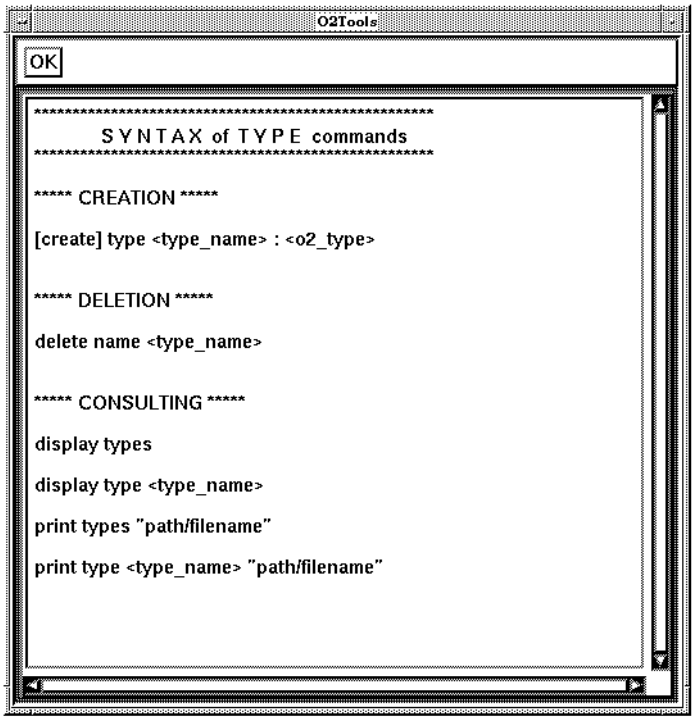


Figure 1.7: Help information

To close this window click on OK. To quit Help click on Cancel.

Note

At the time of printing only the command language help facility is implemented.

Lock

When you click on Lock, O_2Tools is frozen and can no longer be used. To unfreeze O_2Tools, simply click on the Lock button once again and enter your Unix password in the dialog box that appears. You can use O_2Tools once again.

Figure 1.8: Lock password

Commit

To commit or validate a transaction click on the Commit button of the dashboard.

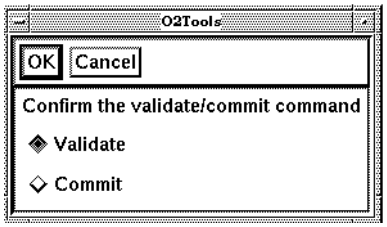


Figure 1.9: Commit or validate

A dialog box appears in which you confirm whether you wish to commit or validate.

Abort

To abort a transaction, click on the Abort button of the dashboard. A dialog box asks for confirmation.

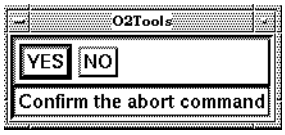


Figure 1.10: Abort dialog box

Note

When you click on Commit or Abort, all the other windows on display disappear while the O_2 restart mechanism is triggered.

Alpha

When you click on Alpha, a dialog box appears in which you confirm that you want to leave O_2Tools and return to the alphanumeric top-level prompt.

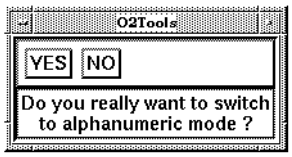


Figure 1.11: Alpha dialog box

To return to O_2Tools type the command **`toolsgraphic`**.

1.6 To exit O_2

To exit O_2, click on the Quit button (*A*) of the dashboard.

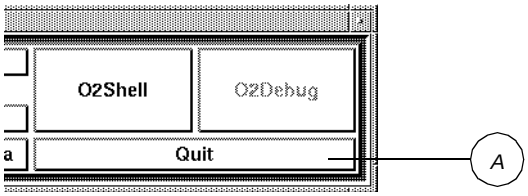


Figure 1.12: Quit button

A dialog box then appears asking for confirmation.

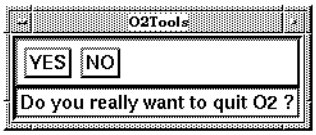


Figure 1.13: Quit confirmation box

1.7 Multiuser Development

When multiple users are developing the same schema simultaneously, it is recommended to use the command
`catalog transaction on`
from within the o2shell editor. By setting catalog transaction on, O_2 will automatically minimize the conflicts for accessing the schema. Every action which modifies the catalogue is carried out as a single transaction, for example create schema, import schema and create base, thereby minimizing the amount of time the schema is locked.

1 Introduction

2 Browsers

This chapter introduces the browsers available with O_2Tools which can be used by both the end user as well as the software developer in order to browse, query and modify the schema and database.

It is divided into the following sections:

- Introducing the O2 Tools browser
- The Schema Browser- browsing schemas and bases
- The Class Browser - browsing classes and methods
- The Application Browser - browsing applications and programs
- The Function Browser - browsing functions
- The Persistent Type Browser - browsing persistent types
- The Persistent Name Browser - browsing persistent names

Note

This chapter explains what you see in O_2Tools if you have the Global Utilization option set to Novice i.e. all the possible confirmation dialog boxes are described. Refer to Section 1.4 for details.

2 Browsers

2.1 Introducing the O_2 Tools browser

You access the browsers of O_2Tools via the dashboard. When the O_2Tools dashboard is displayed at the beginning of a session only the Schemas button can be selected (unless you have preset the O_2Tools configuration using the O_2Tools button).

Important

You must first select the schema and/ or base you want to browse. This is described in Section 2.2.

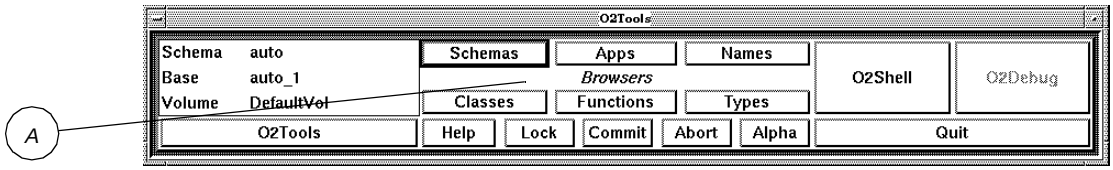


Figure 2.14: O_2Tools dashboard

Once you have set the schema and base all the other browser buttons can be selected and you can begin to browse the database. To display any browser, simply click on the corresponding button on the dashboard (*A*) of the browser you wish to open:

- Schemas
- Classes
- Apps
- Functions
- Types
- Names

The browser window now appears.

Common features

When you click on a browser button:

- the browser is displayed
- if already displayed, the browser comes to the front of all other windows
- if iconified in X windows, the browser window is displayed and comes to the front of all other windows.

Each browser has a menu bar and title, one or more lists from which elements can be selected, one or two optional windows and a

documentation/ error messages window. This can be seen in the Schema Browser in Section 2.2.

- ***Window and Options menus***

  The menu bar entries Window and Options appear in every browser:

  - Window

  This menu has one entry, Close which allows you to leave the browser at any time.

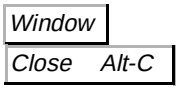


Figure 2.15: Window menu

  - Options

    This menu contains different options with which you can Toggle the display of each window on and off enabling you to concentrate on a particular part of the browser. The Options menu also contains a Clear Documentation item that allows you to clear the browser Error messages/ Documentation window.

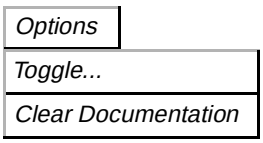


Figure 2.16: Options menu

- ***Accelerators***

  The browser menus have various accelerators:

  Accelerators are marked on the menu to the right of the menu item. When you type Alt-C, the Window menu item Closeis selected and the browser is closed without the actual Window menu, Figure 2.15, ever being displayed.

- ***Edit***

  You edit, create, delete or rename a particular source using the corresponding browser. To edit a source, you access and display the source editor in one of the following ways:

  - chose the Edit option in the corresponding source menu
  - using the left mouse button, double-click on the source name in the browser list

- drag and drop the source name onto an already displayed source editor. See Section 3.1 describing the common features of Source editors.

2.2 The Schema Browser

With the Schema Browser you can visualize the definitions of the schemas and bases available to you before setting a schema and base to be browsed. You can also easily create and delete your schemas and bases.

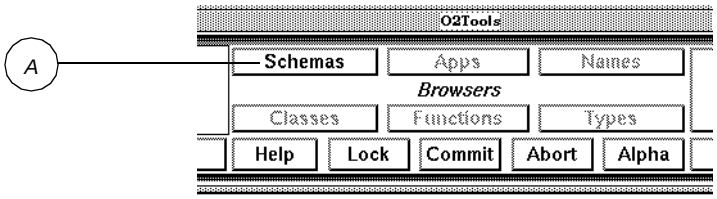


Figure 2.17: Schema button

To display the Schema Browser click on the Schemas button (*A*) on the O_2Tools dashboard. The following window appears:

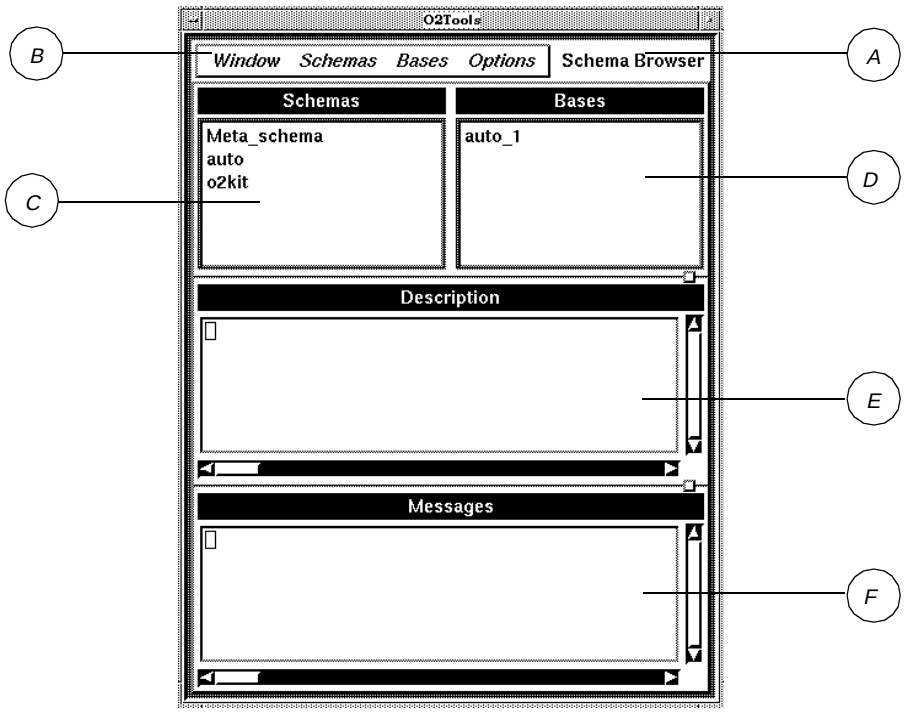


Figure 2.18: Schema Browser Components:

A: Browser title	*B: Menu bar*
C: Schema list	*D: Base list*
E: Description window	*F: Error messages window*

As in Figure 2.18, when you display the Schema Browser for the first time no schema or base is selected. The Schema list shows all the schemas and the Base list all the bases available to you.

Manipulating Schemas

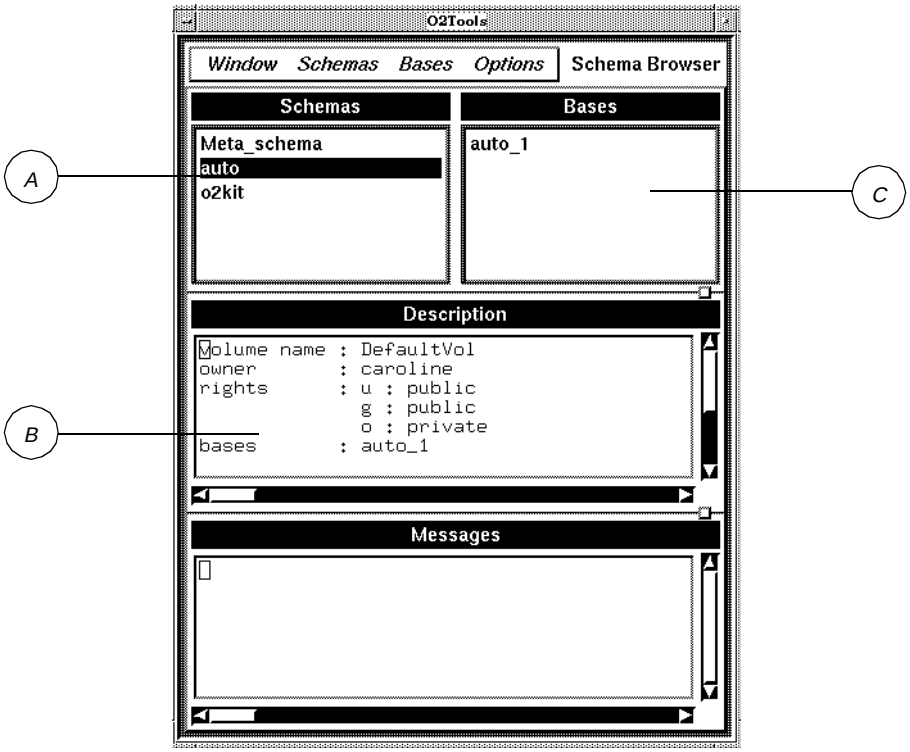


Figure 2.19: Description of the schema "auto"

To obtain information about a particular schema, simply click on the schema name in the Schema list (*A*).

All the bases attached to the schema are displayed in the Bases list (*C*).

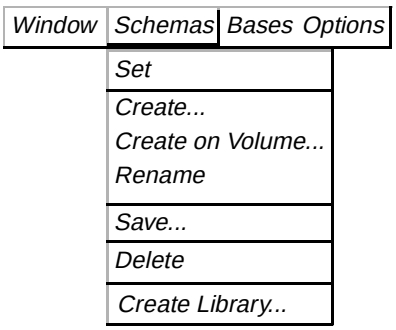


Figure 2.20: The Schemas menu

With the Schemas menu, shown in Figure 2.20, you set the schema you want to browse as well as create and delete schemas. The rest of this section details these menu items.

- ***Setting a schema***

  To browse a particular schema you must firstly set it as the current working schema. Select the name of the schema you want to browse in the Schema list and then click on Set in the Schemas menu.

Note

The dashboard now shows your schema as the current working schema. All the other browser buttons on the dashboard can now be activated and you can start browsing.

If you already have a current working schema with browsers and/ or sources editors on display and you want to set another schema as the current working schema, a dialog box appears with which you confirm your decision to change schemas.

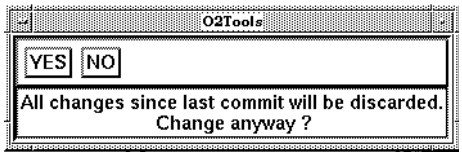


Figure 2.21: Set Schema confirmation

All the displayed browsers and source editors immediately disappear and the new schema is currently set.

- ***Creating a schema***

  To create a schema on the Default volume, simply select Create in the Schemas menu and enter the name of the new schema in the dialog box that appears and click on OK.

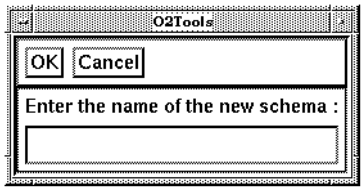


Figure 2.22: Create schema dialog box

The new schema name now appears in the Schema list and is set as the current working schema in the browser and on the dashboard.

- ***Creating a schema on a volume***

  To create a schema on a particular volume, simply select Create on Volume. The dialog box to enter the new schema name now appears as above. When you have entered the new name and clicked on OK, the list of volumes available is then displayed in a separate window.

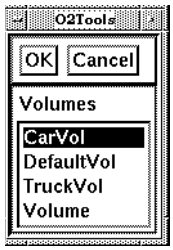


Figure 2.23: List of available volumes

Click on the volume name you want to use and click on OK.

- ***Renaming a schema***

  To rename a schema, select Rename from the Schemas menu and you see the following box:

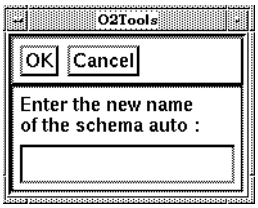


Figure 2.24: Rename confirmation box

Enter the new name and click on Ok. The schema is renamed and appears in the Schema list.

- ***Saving a schema***

  Save enables you to save a schema in a Unix file.

  When you choose this item, a dialog box pops up for the path name.

  You can either give a file name, saving the entire schema in this file, or you can give a directory name, thereby generating different files

containing the various components of classes, applications, functions, persistent types and persistent names.

Please refer to the *O_2C Reference Manual* for more details.

- ***Deleting a schema***

  To delete a schema, select Delete. A dialog appears in which you confirm the deletion.

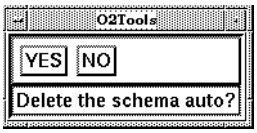


Figure 2.25: Delete confirmation

The schema name is now removed from the Schema list.

Note

You cannot delete a schema that either has been set during the current transaction or that has bases associated to it. You must set another schema and commit the transaction in order to then delete the schema.

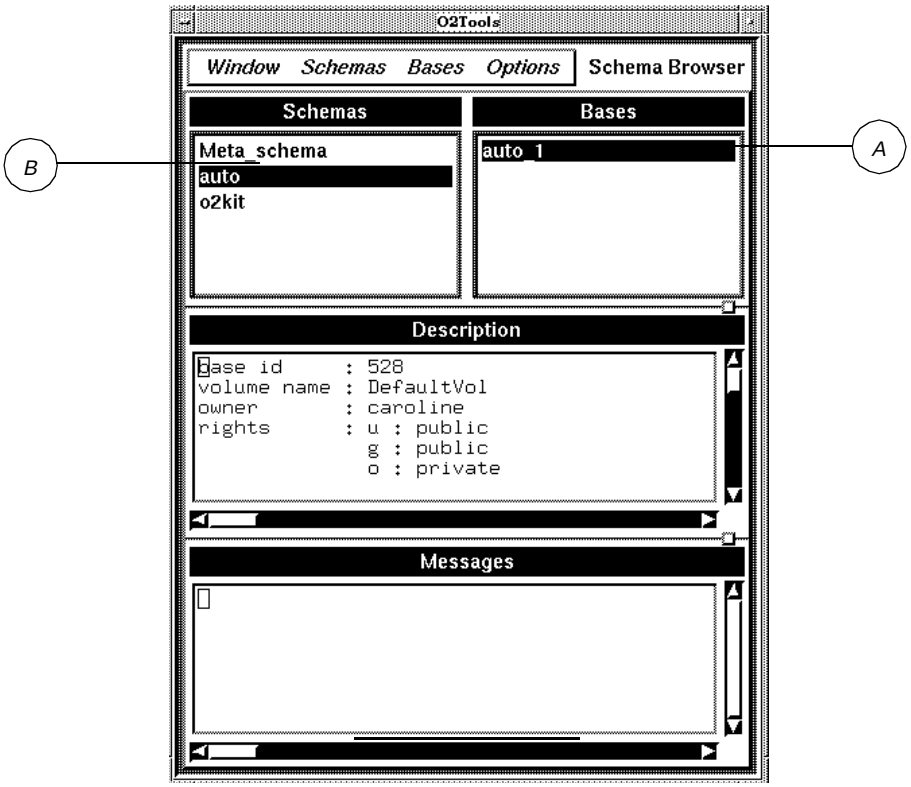
- ***Creating a library***

  To create a library for a specified schema, simply select Create Library in the Schemas menu and enter the name of the library in the dialog box that appears and click on OK.

Manipulating Bases

To obtain information about a particular base, simply click on the base name in the Bases list (*A*) as in Figure 2.26 below. The associated schema is selected (*B*).

Figure 2.26: Selecting a base



With the Bases menu, shown in Figure 2.27, you can look at all the bases available to you and set the base you want to browse.

You can also create and delete bases. This section now details these menu items.

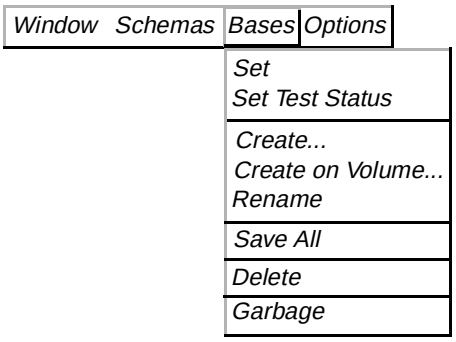


Figure 2.27: The Bases menu

- ***Setting the TEST Status***

  To set the TEST status of a particular base, simply select the base name from the Bases list and click on Set Test Status in the Bases menu. Refer to the O_2 System Administration Guide for further details.

- ***Setting a base***

  To set a particular base to be the current working base, simply select the base name from the Bases list and click on Set in the Bases menu. This base becomes the current working base and its associated schema becomes the current working schema.

Note

The dashboard now displays your newly set base name and associated schema. All browser buttons on the dashboard can now be activated and you can begin browsing the selected base.

If you already have a current working base with browsers and/ or sources editors on display and you want to set another base as the current working base, a dialog box appears with which you confirm your decision to change bases.

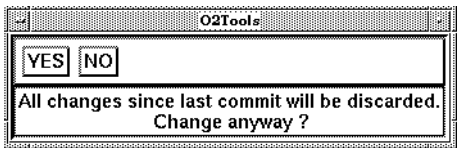


Figure 2.28: Set base confirmation

All the displayed browsers and source editors immediately disappear and the new base is currently set.

- ***Creating a base***

  Select Create from the Bases menu. Enter the name of the new base in the dialog box that appears and click on OK.

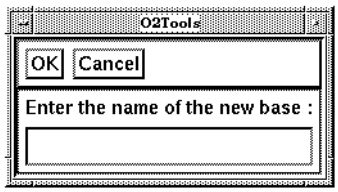


Figure 2.29: Create base dialog box

The created base now appears in the Bases list and becomes the current working base on the Default Volume in the browser and on the dashboard.

- ***Creating a base on a volume***

  To create a base on a particular volume, simply select Create on Volume. Enter the name of the new base in the dialog box that appears as above. The list of volumes available is then displayed in a separate window.

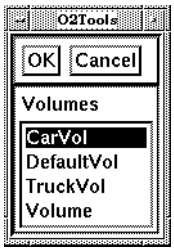


Figure 2.30: List of available volumes

Click on the volume name you want to use and click on OK.

- ***Renaming a base***

  To rename a base, select Rename. You see the following box:

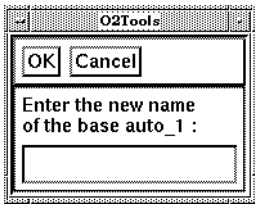


Figure 2.31: Rename a base

Enter the new name and click on OK. The base is renamed and the new base name appears in the Base list.

- ***Available bases***

  From the Bases menu, select Show All. All the bases available to you are now displayed in the Bases list.

- ***Deleting a base***

Click on the name of the base you wish to delete from the Bases list and select Delete. A dialog box appears in which you confirm the deletion.

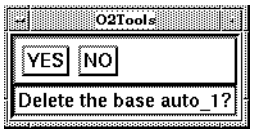


Figure 2.32: Delete base confirmation

The base name is deleted from the Bases list and no base is selected. The description window is refreshed with the schema definition.‘

Note

You cannot delete a base that has been set during the current transaction. You must set another base and commit the transaction in order to then delete the base.

- ***Garbage collecting a base***

  To use the garbage collector, simply select the base name from the Bases list and click on Garbage in the Bases menu.

2.3 The Class Browser

You use the Class Browser to visualize the definitions of classes, methods and named objects as well as the associated class hierarchy and documentation. You access the database via the persistent roots of the schema to display and modify data, classes and methods.

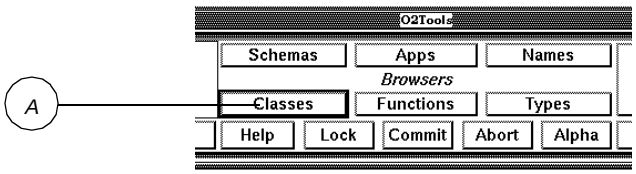


Figure 2.33: Classes button

To display the Class Browser, click on the Classes button (*A*) on the O_2Tools dashboard as in Figure 2.33. (This is only possible if you have first selected a schema). The Class Browser is then displayed with the class list showing all the classes available with the current working schema. No class is selected.

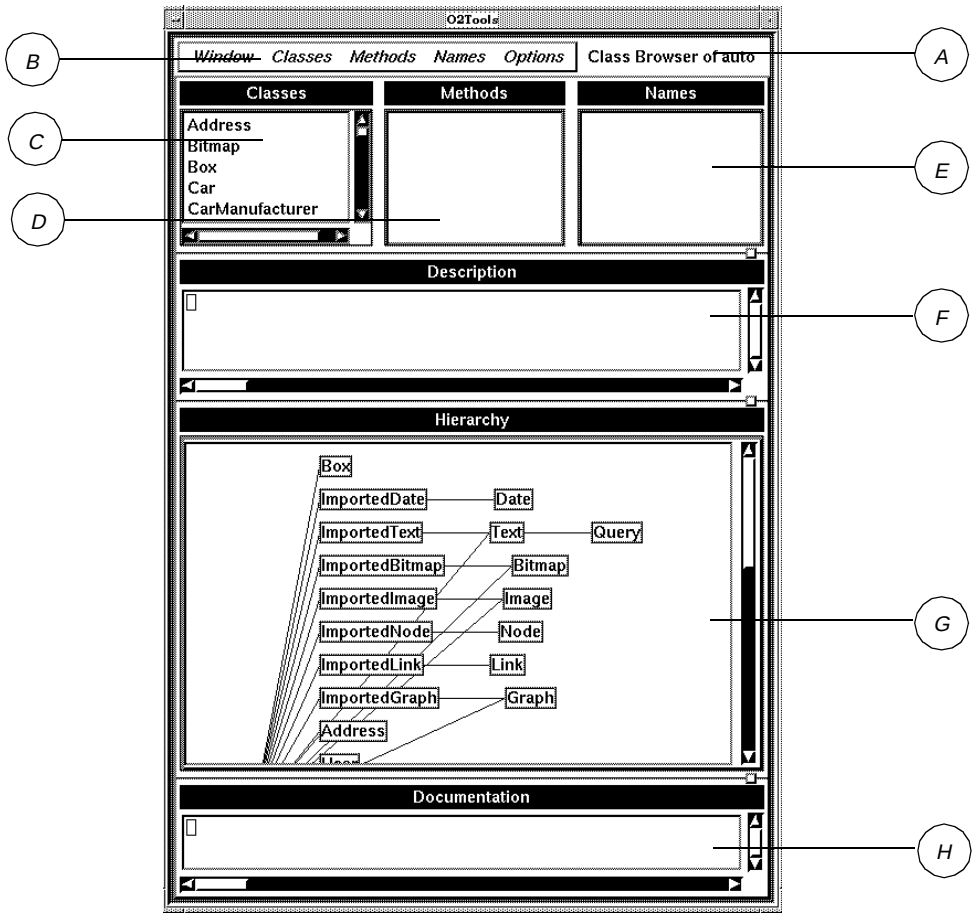


Figure 2.34: Class Browser components:

A: Browser title	*B: Menu bar*	*C: Class list*
D: Method list	*E: Named object list*	*F: Description window*
G: Class hierarchy	*H: Documentation*	

Manipulating Classes

To obtain information about a particular class, click on the class name in the Class list, as in (*A*) in Figure 2.35.

The local methods (*B*) and the named objects (*C*) belonging to the selected class are then displayed in the Class Browser.

Figure 2.35: Selecting a class

In the Hierarchy window (*E*), the node corresponding to the selected class is highlighted in the graph. The documentation of the selected class, if any exists, is displayed in the documentation window (*F*).

In the description window (*D*), information about the class is displayed.

For example, the alphanumeric display of the class with the complete type of the class (i.e. local and inherited attributes) is given, as in Figure 2.36 below.

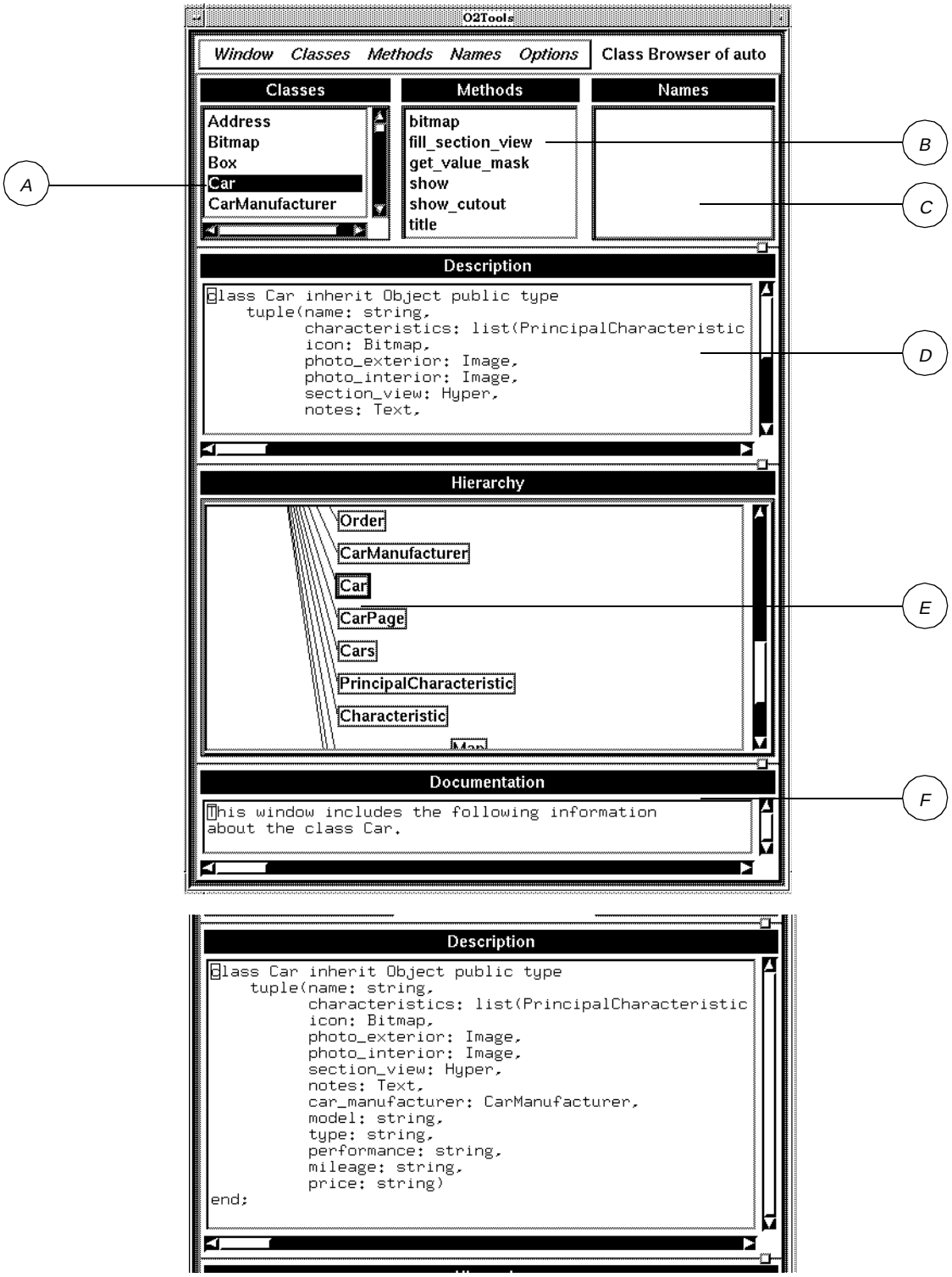


Figure 2.36: Class description

Note

If you click directly on a graph node with the left mouse button, you obtain exactly the same information.

With the Classes menu, shown in Figure 2.37, you can create and delete classes and inheritance links as well as rename classes. You can also display the class source editor.

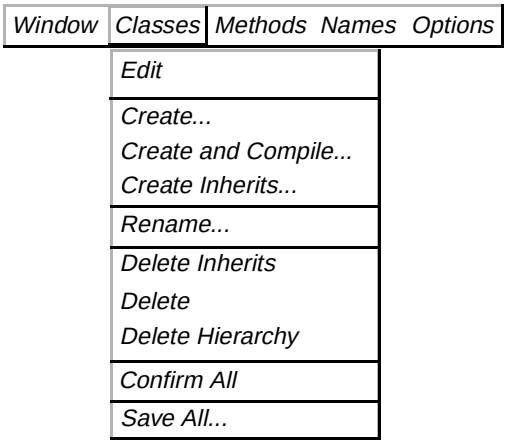


Figure 2.37: Classes menu

When you click on any node in the Class Hierarchy window, using the right mouse button, you obtain a pop-up menu as shown in Figure 2.38 below that has the same items as the Classes menu (except Delete Inherits, Confirm All and Save All).

This node does not need to be selected as shown in Figure 2.38. This is simply another faster way of accessing the various options available to you.

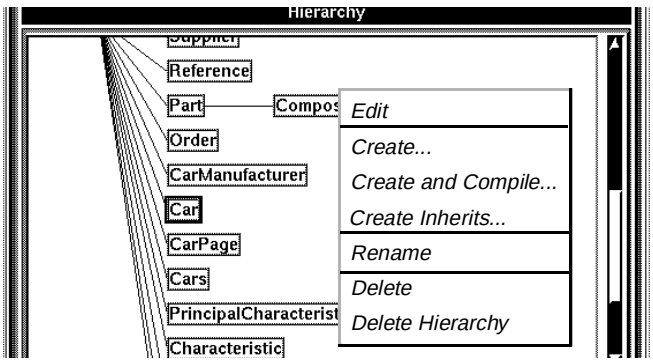


Figure 2.38: the class hierarchy pop-up menu

- ***Displaying a class source editor***

  To display the source editor of a class you can either

  - select the class in the Class list and click on Edit in the Classes menu, or
  - double click with the left mouse button on the class name in the Class list, or
  - click on the class name using the middle mouse button and drag and drop the name onto an already displayed class editor, see Section 3.1, or
  - click on the relevant node in the Hierarchy graph using the right mouse button and select Edit in the pop-up menu.

  A class source editor is then displayed as in Figure 2.39. For more information on how to use a class source editor, see Section 3.5.

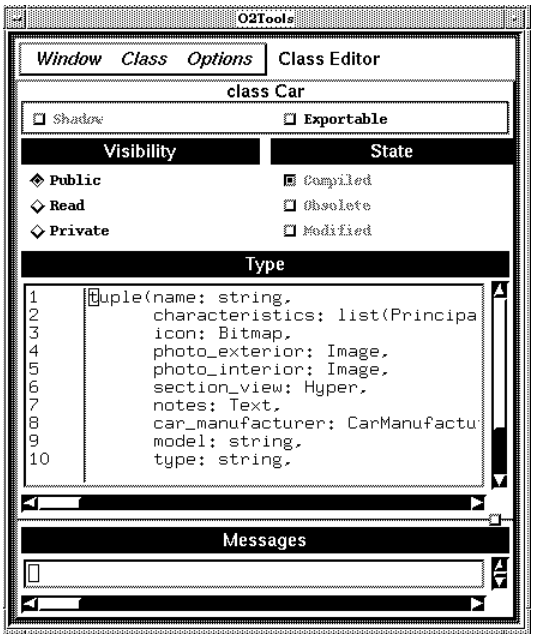


Figure 2.39: A Class source editor

- ***Creating a class***

  When you create a new class it is a subclass of the currently selected class.

  Select Create in the Classes menu, and enter the name of the new class in the dialog box that appears and click on OK.

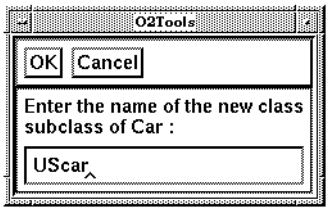


Figure 2.40: Creating a new class

The newly created class name appears in the Class list and a node appears in the hierarchy graph.

The created class becomes the currently selected class and its default definition is displayed in the description window prefixed with a comment **`/* uncompiled class */`**.

The source of the new class is displayed so that you can compile the new class immediately.

Warning !

The source of the class is created but the class is not yet compiled. Please refer to Section 3.5 for more details on how to use and compile a class.

- ***Creating and compiling a class***

  You can also create and compile a class directly without going through the Class source editor. Chose Create and Compile from the Class menu. Enter the name of the new class in the dialog box and click on OK. The created class inherits the visibility and type of its super-class.

- ***Creating an inheritance link***

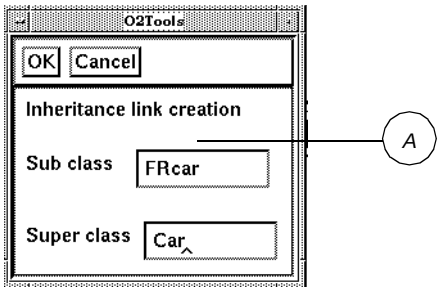
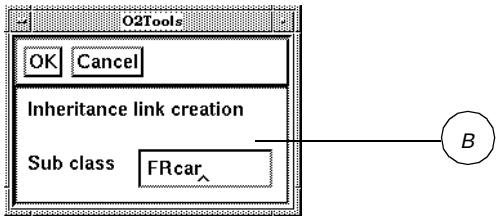


Figure 2.41: Create inherits dialog box

You can create an inheritance link in several ways.

- If no class is selected in the Class list and you select Create Inherits, the dialog box in Figure 2.41 (*A*) appears asking for the sub class and the super class.



Create Inherits dialog box

- If you have selected a class in the Class list and the you select Create Inherits, the dialog box in Figure (*B*) appears asking for the sub class. Enter the class name(s) and click on OK. The created inheritance link is then shown in the description window and in the graph.

- You can also create an inheritance link by clicking on the graph node of the super class using Control-Shift and the right mouse button. A ghost of the link appears which you can then drag the cursor across the graph. You then drop the link by releasing the mouse button when you reach another node (the subclass). In Figure 2.42 the subclass is FRcar and the super class Car.

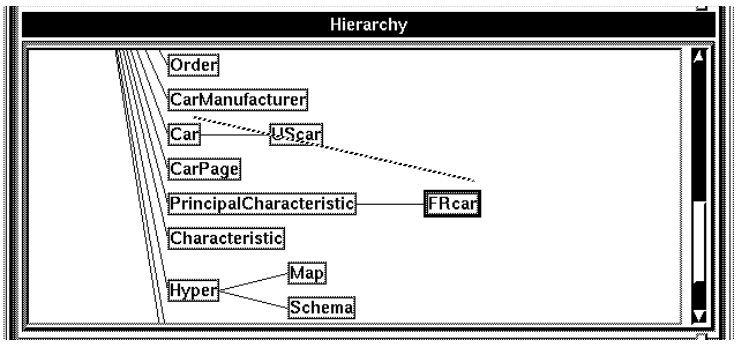


Figure 2.42: Interactive graphs to create inheritance links

- ***Renaming a class***

To rename a particular class, select Rename and enter the new name in the dialog box that appears and click on OK. The class is renamed and the new name appears in the Class list.

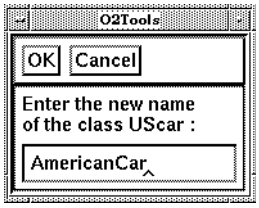


Figure 2.43: Class rename dialog box

If the source editor of the renamed class is displayed, it is refreshed with the new name.

- ***Deleting a class***

  Select the name of the class you wish to delete from the Class list or on the graph and select Delete. A dialog box appears in which you confirm the deletion.

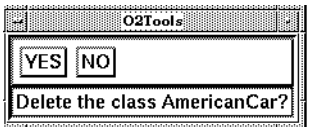


Figure 2.44: Delete class confirmation

The class name is then deleted from the Class list and no class is currently selected. If the source editor of the deleted class is displayed, it is unmapped as are all the other mapped method and named object source editors associated to the deleted class.

- ***Deleting an inheritance link***

  Select the link you wish to delete on the actual graph, and select Delete Inherits. A dialog box appears in which you confirm the deletion.

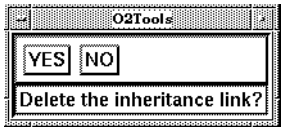


Figure 2.45: Delete Inherits confirmation

The description window and the graph are refreshed immediately to include your modifications.

Note

This item is not available in the Hierarchy pop-up menu

- ***Deleting a hierarchy***

  This item enables you to delete an entire section of the class hierarchy. Select the node at which you wish to start deleting the hierarchy on the actual graph and select Delete Hierarchy. A dialog box appears in which you confirm the deletion.

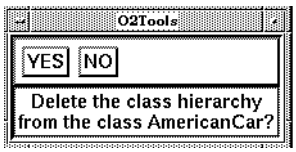


Figure 2.46: Delete hierarchy

All the nodes from and including the node selected are deleted from the hierarchy. The Description window and Class list are refreshed immediately.

- ***Confirm classes***

  To confirm all the classes, select Confirm All from the Class menu.

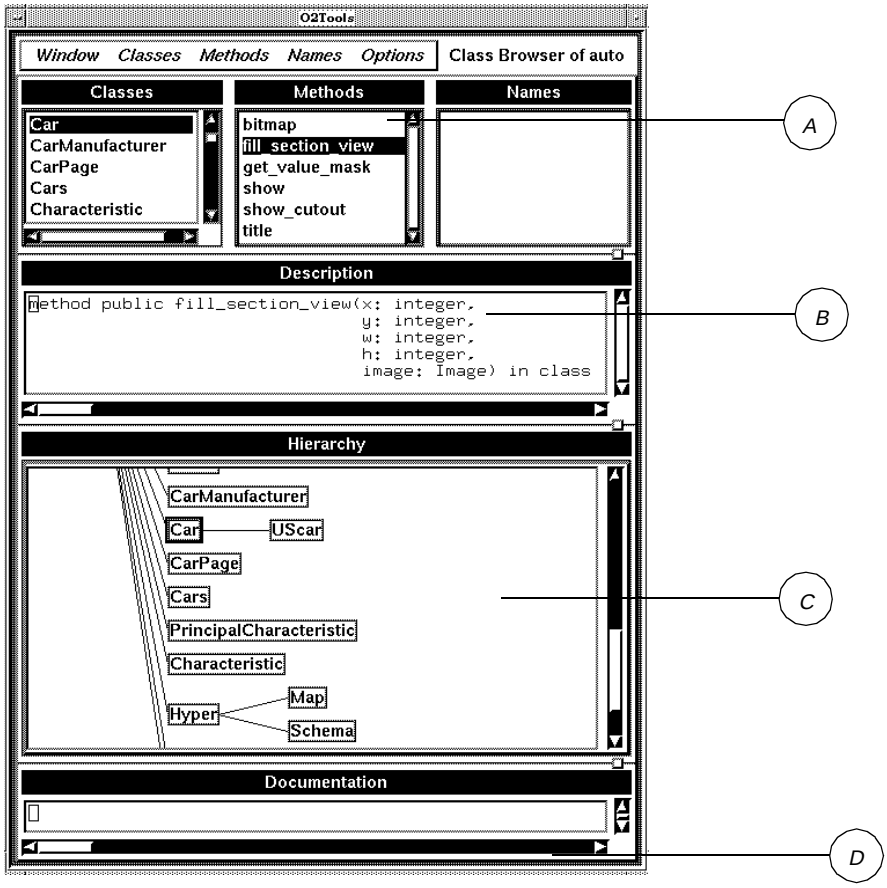
- ***Saving classes***

  To save all the classes, select Save All from the Class menu.

Manipulating Methods

For information about a particular method, click on its name in the Method list (*A*) shown in Figure 2.47. Any documentation appears in the Documentation window (*D*) and the method signature in the description window (*B*).

Figure 2.47: Method description



With the Methods menu, in Figure 2.48, you create, delete and rename methods as well as display method source editors and inherited methods. This section now describes these items.

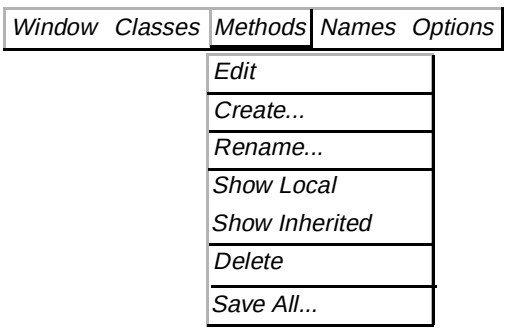


Figure 2.48: Methods menu

- ***Displaying a method source editor***
  To display the method source editor you can either:

- select the class in the Method list and click on Edit in the Methods menu, or
- double click with the left mouse button on the method name in the Method list, or
- click on the method name using the middle mouse button and drag and drop the name onto an already displayed method editor, see Section 3.1.

A method source editor is then displayed as in Figure 2.49 below, containing the method signature and body. For more details on editing and compiling methods, refer to Section 3.6.

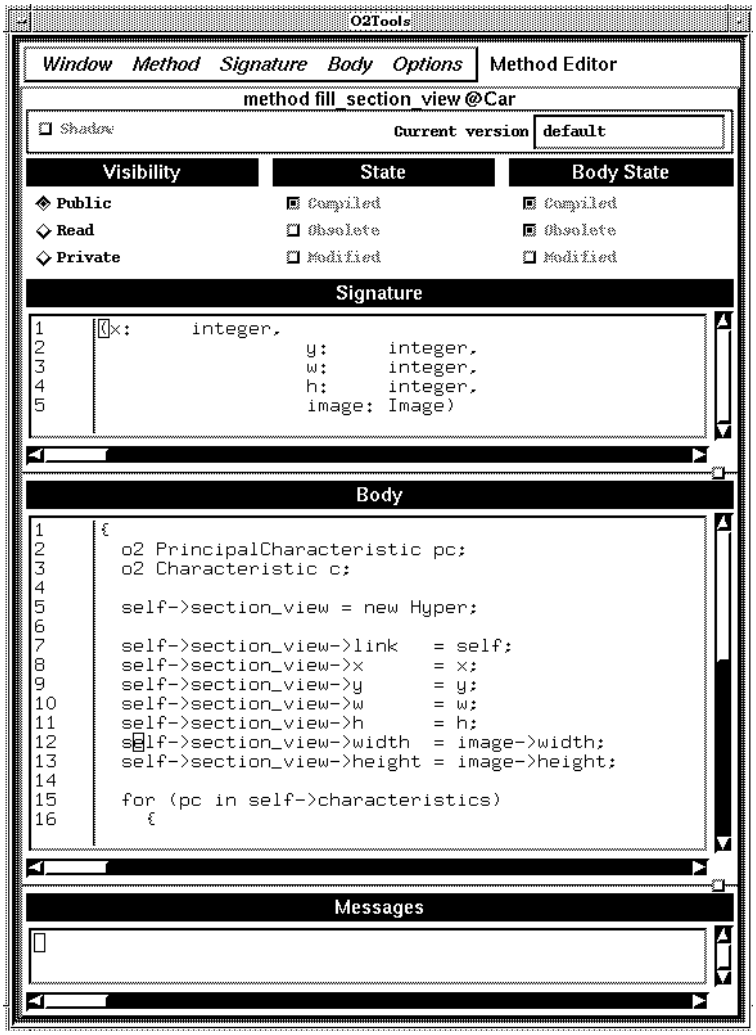


Figure 2.49: Method source editor

- ***Creating a method***

To create a method, you must firstly select in the Class list, the class to which the method will belong.

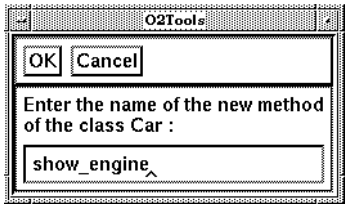


Figure 2.50: Create method dialog box

Now select Create and enter the name in the box displayed.

Click on OK. The new method name now appears in the Method list (*A*) and is the currently selected method.

Its default signature is displayed in the description window but is prefixed with the comment **`/* uncompiled method */`** as in Figure 2.51.

The method source editor is also displayed.

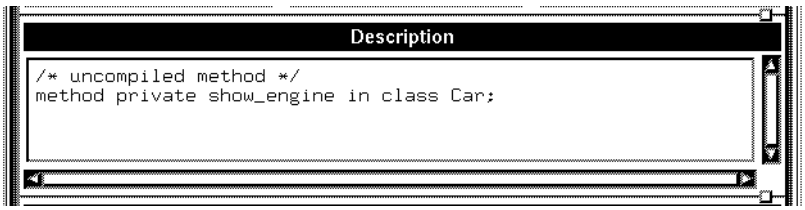


Figure 2.51: Uncompiled method

Warning !

The method is created but not compiled. For details about compiling methods, see Section 3.6.

- ***Renaming a method***

To rename a method, select Rename and enter the new name in the dialog box that appears. Click on OK.

The new name is now displayed in the Method list and in the method source editor.

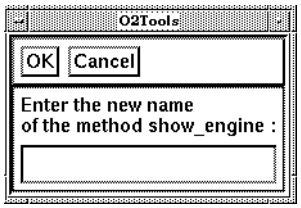


Figure 2.52: Method rename dialog box

- ***Inherited methods***

  With the class name selected in the Class list, select Show Inherited. All the inherited methods are now shown in the Class Browser.

  To see the local methods again, select Show Local.

- ***Deleting a method***

  Select the name of the method you wish to delete and click on Delete from the Methods menu. A dialog box appears in which you confirm the deletion.

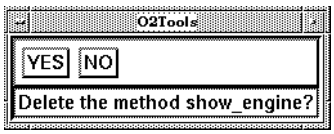


Figure 2.53: Delete method dialog box

The method name no longer appears in the Method list and no method is currently selected.

The description window is refreshed to contain the definition of the currently selected class. If the method source editor is displayed it is unmapped immediately.

- ***Saving methods***

  To save all the methods, select Save All from the Methods menu.

Manipulating Named objects

For information about a named object, simply click on its name in the Name list (*A*). The description of the named object now appears in the description window (*B*).

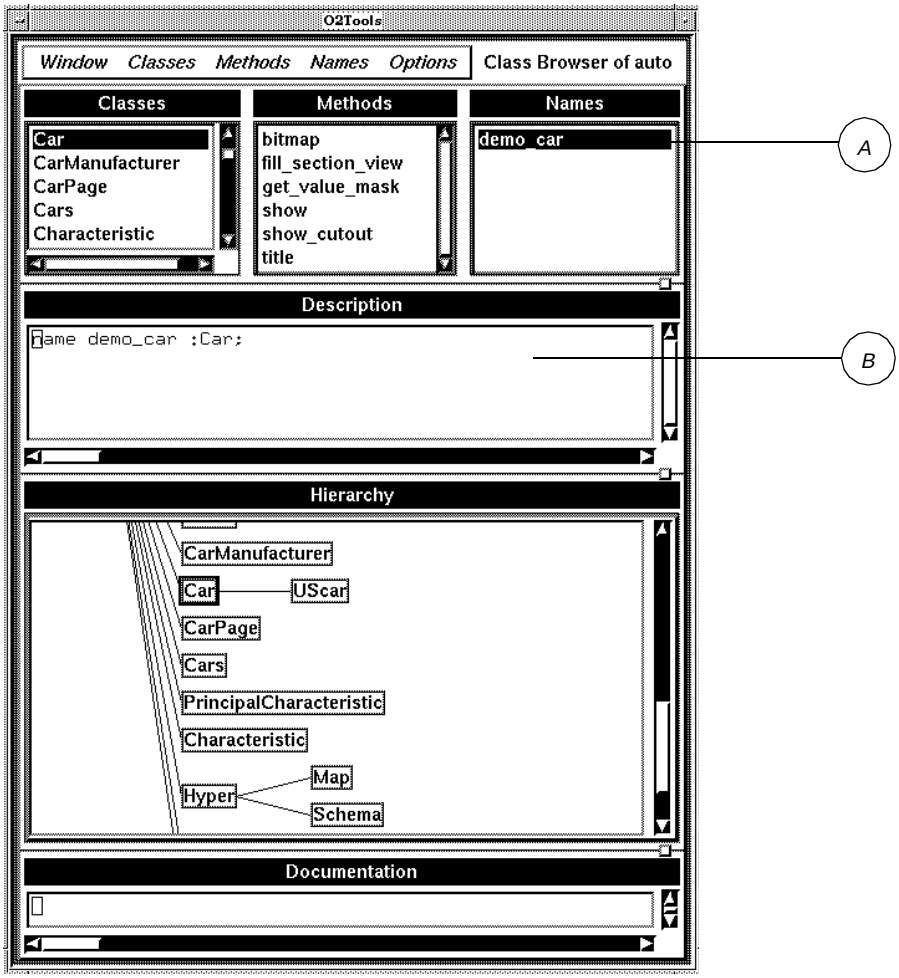


Figure 2.54: Named object

With the Names menu, shown in Figure 2.57, you can create, delete and rename named objects as well as display the value of a name.

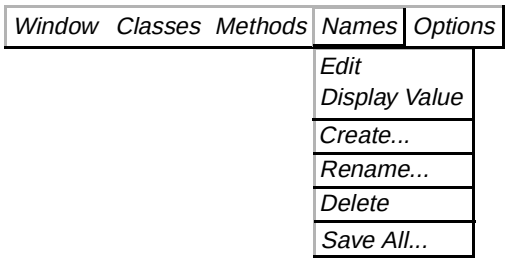


Figure 2.55: Names menu

This section now details these items.

- ***Displaying a named object editor***

  To display the named object source editor you can either

  - select the name in the Name list and click on Edit in the Names menu, or
  - double click with the left mouse button on the named object name in the Name list, or
  - click on the named object name using the middle mouse button and drag and drop the name onto an already displayed named object editor, see Section 3.11

  A named object source editor is then displayed as in below.

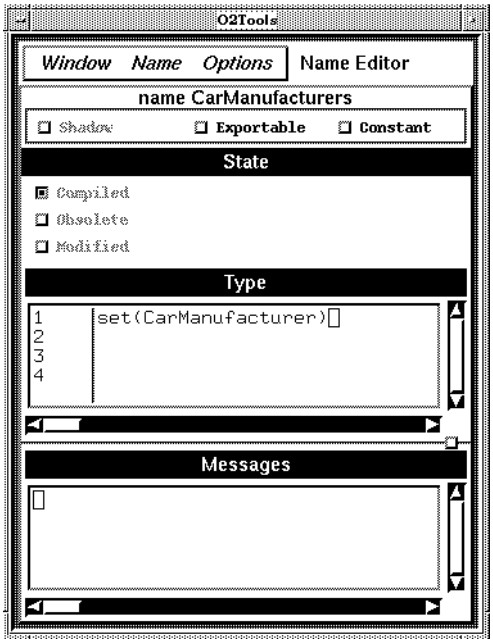


Figure 2.56: Name source editor

- ***Displaying the value of a name***

  Select the name in the Class Browser Name list and then select Display Value. The value of the named object is displayed separately as in Figure 2.57.

Figure 2.57: Value of a named object

- ***Creating a named object***

  You must first select the class to which the named object will belong. Then simply click on Create in the Names menu, enter the name of the new named object in the dialog box that appears and click on OK.

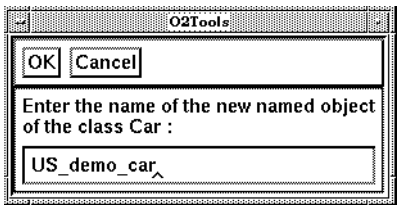


Figure 2.58: Create named object dialog box

The new named object now appears in the Name list. If the Persistent Name Browser (see Section 2.7) is displayed, it is immediately refreshed with the new name.

- ***Renaming a named object***

  Select the named object to be renamed and select Rename. Enter the new name in the dialog box that is displayed and click on OK.

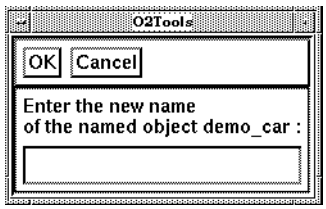


Figure 2.59: Rename dialog box

The newly named object now appears in the Name list and the Persistent Name Browser, if displayed, is refreshed.

- ***Deleting a named object***

  Select the named object you wish to delete and select Delete. A dialog box appears in which you confirm the deletion.

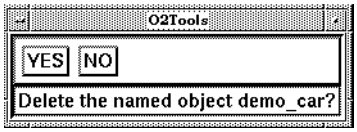


Figure 2.60: Delete named object confirmation

The name no longer appears in the Name list and no named object is currently selected. The Persistent Name Browser is also refreshed if it is displayed.

- ***Saving named objects***

  To save all the named objects, select Save All from the Names menu.

2.4 The Application Browser

With the Application Browser, you can create applications, programs, application variables as well as display their definitions and associated documentation and dynamically test your applications.

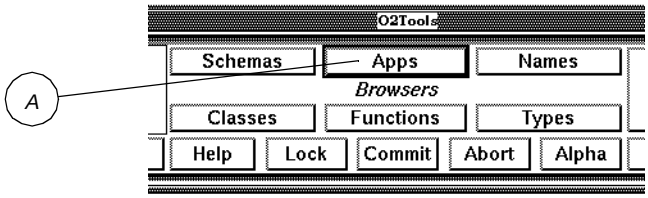


Figure 2.61: Apps button

With the schema selected and set up as in Section 2.2, simply click on the Apps button (*A*) on the dashboard in order to display the Application Browser, shown in Figure 2.62 below.

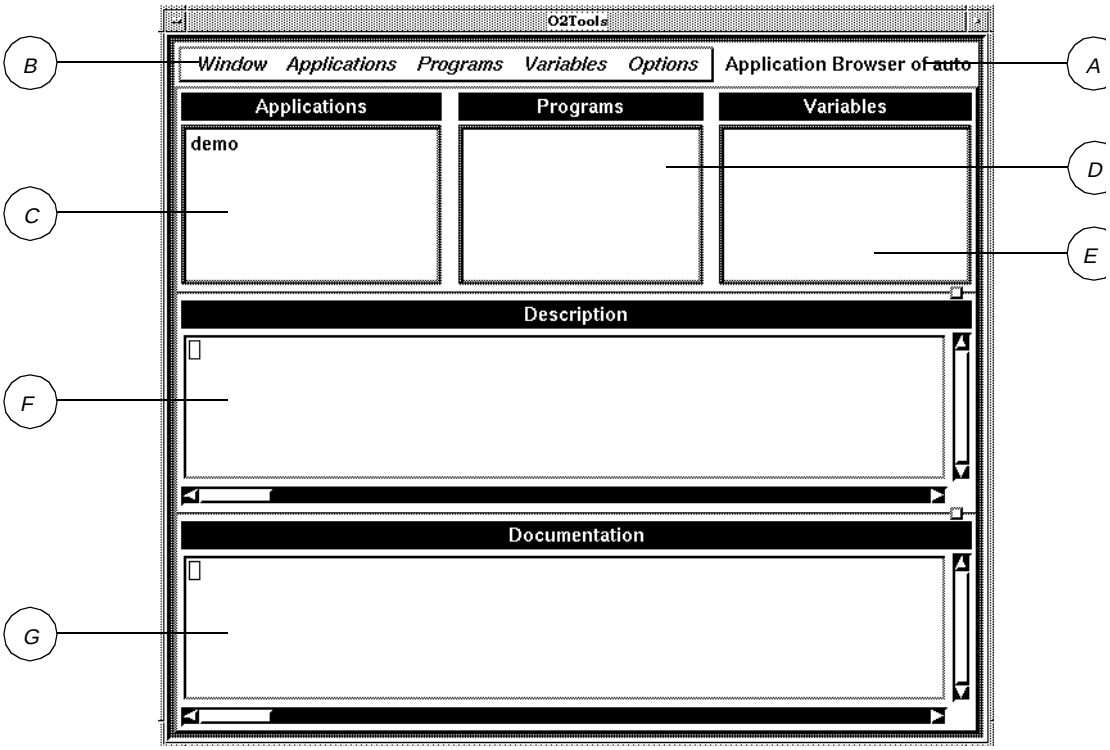


Figure 2.62: Application Browser components:

A: Browser title *B: Menu bar*
C: Application list *D: Program list*
E: Application variable list *F: Description*
G: Documentation/error messages

When you display the Application Browser for the first time, all applications available to you with the current schema are listed in the Application list (*C*). No application is selected.

Manipulating Applications

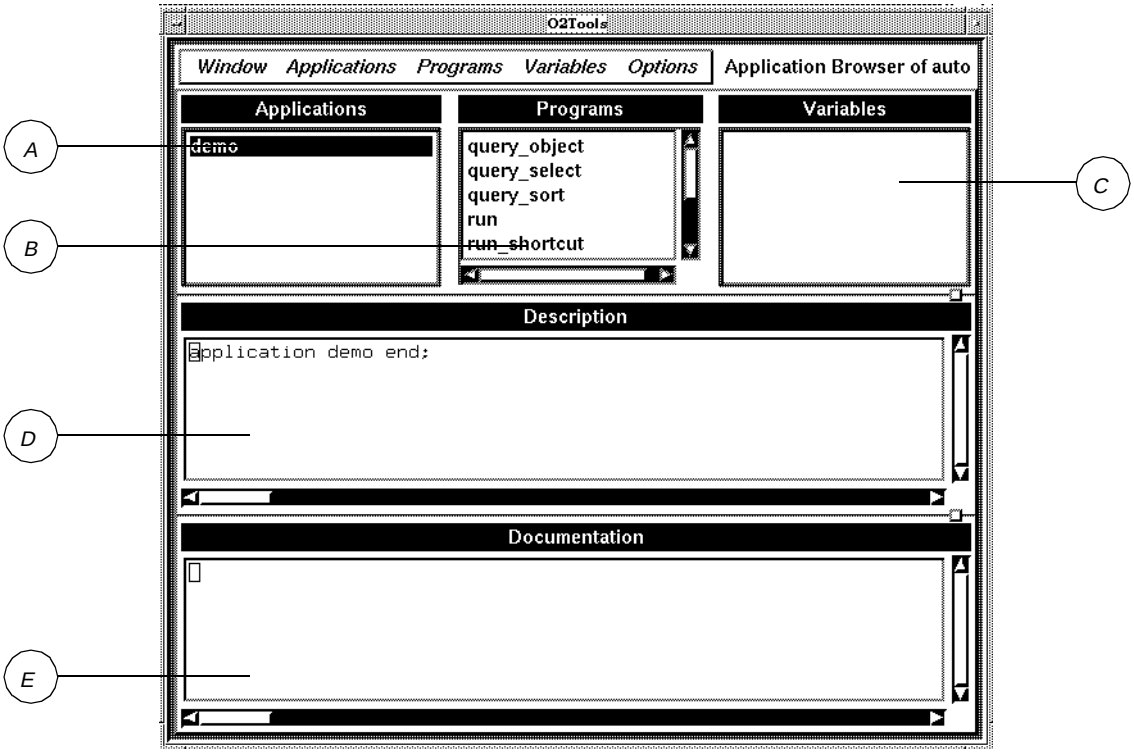


Figure 2.63: Selected application

If you want information on a specific application, simply click on its name in the Application list (*A*) as in Figure 2.63.

You now see all the programs (*B*) and application variables (*C*) belonging to that application.

The description of the application is displayed in the Description window (*D*). Any associated documentation also appears (*E*).

With the Applications menu, shown in Figure 2.64, you can create, delete and rename applications.

This section describes each of these items.

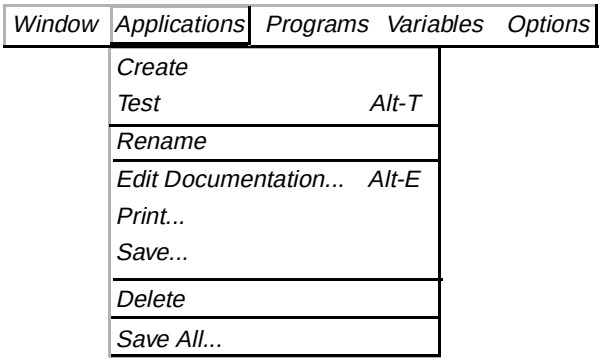


Figure 2.64: Applications menu

- ***Creating an application***

  Select Create from the Applications menu and enter the name of the new application in the dialog box that is displayed.

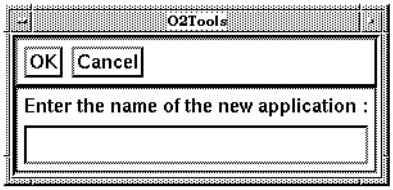


Figure 2.65: Creating an application

Click on OK to confirm the new name. The new application appears in the Application list.

Warning !

An application is automatically compiled

- ***Testing an application***

  To test an application display the Applications pull-down menu from the Application Browser and select Test.

Note

When the application is running O_2Tools is inhibited. It is available again when the application test finishes.

- ***Renaming an application***

  Select Rename from the Applications menu and enter the name of the application to be renamed in the dialog box that is displayed.

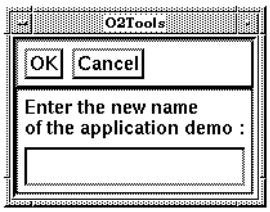


Figure 2.66: Renaming an application

Click on OK to confirm the new name. The new name appears in the Application list. Any related program source editors are also refreshed immediately with the new name.

- ***Printing an application***

  This item prints the application into an external file as with the alphanumeric command print.

  For example:

  `print application demo “path_name”`

  When you click on Print a dialog box appears asking you to choose the file you want to use. Select the file you wish to use and click on OK.

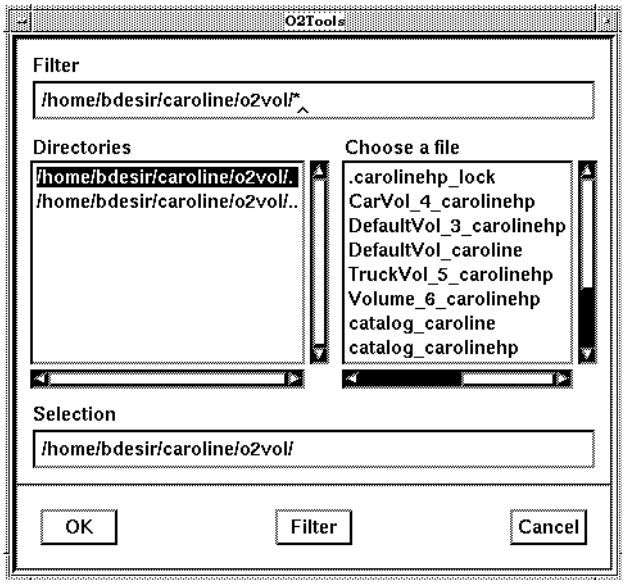


Figure 2.67: Print dialog box

- ***Saving an application***

  To save a paticular application click on its name in the Application list and select Save from the Applications menu.

- ***Deleting an application***

  Click on the name of the application to be deleted and Select Delete. A dialog box now appears asking you to confirm the deletion.

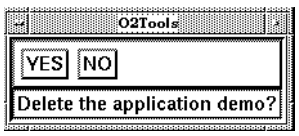


Figure 2.68: Confirm Delete application

The deleted application name disappears from the Application list.

Any related source editors of programs or application variables are immediately unmapped if they are displayed.

- ***Saving all applications***

  To save all the applications, select Save All from the Applications menu.

Manipulating programs

To get information about a particular program, click on the program name in the Program list (*A*) in the Application Browser, as shown in Figure 2.69.

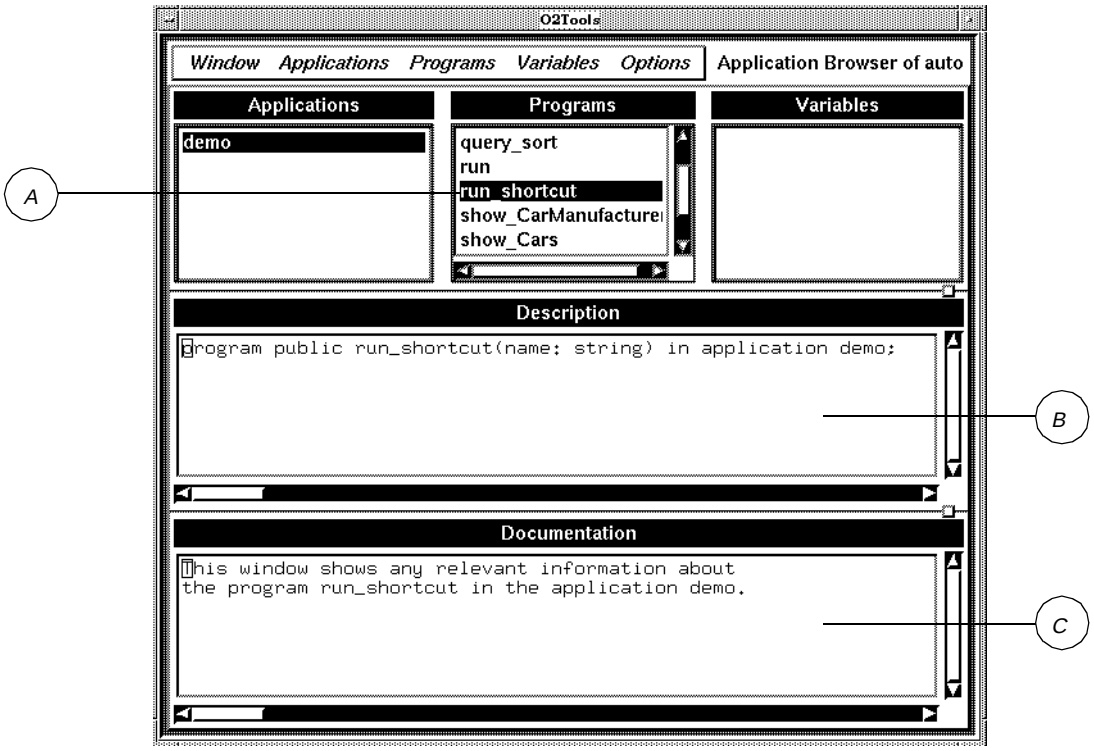


Figure 2.69: Selected Program

The program's signature appears in the description window (*B*) and any documentation in the Documentation window (*C*).

With the Programs menu, shown in Figure 2.70, you can create, delete and rename programs as well as display program source editors.

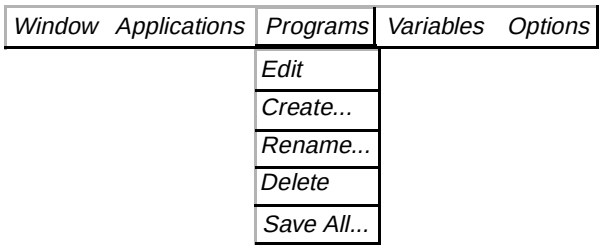


Figure 2.70: Programs menu

This section now outlines these items.

- ***Displaying a program source editor***

  To display the program source editor you can either
  - select the program in the Program list and click on Edit in the Programs menu, or
  - double click with the left mouse button on the program name in the Program list, or
  - click to select the program name using the middle mouse button and drag and drop it onto an already displayed program editor.

  The program source editor appears as in Figure 2.71. To use program source editor, refer to Section 3.7.

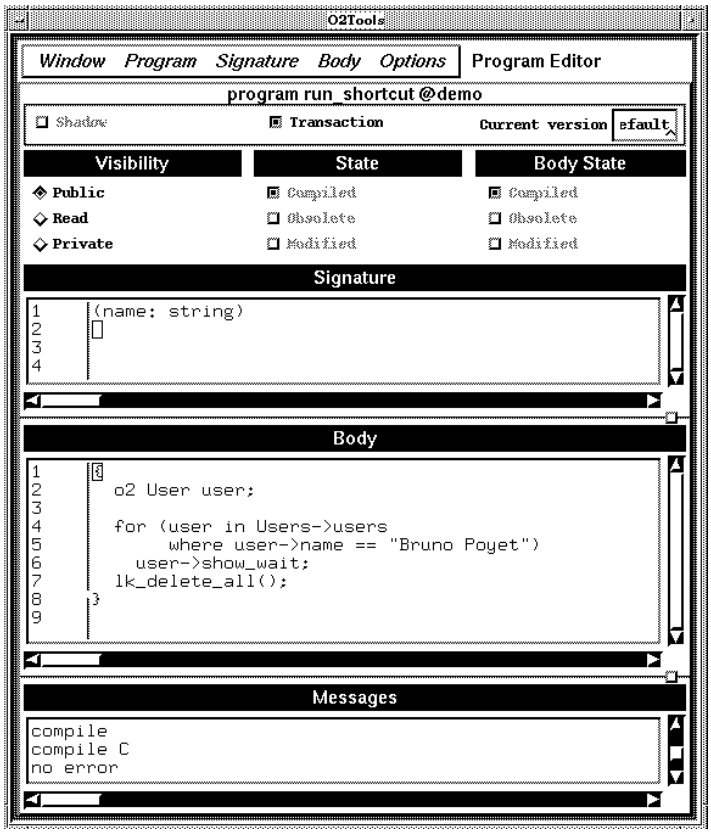


Figure 2.71: Program source editor

- ***Creating a program***

  The program you create belongs to the currently selected application. Select Create from the Programs menu and enter the name of the new program in the dialog box shown in Figure 2.72.

  Click on OK.

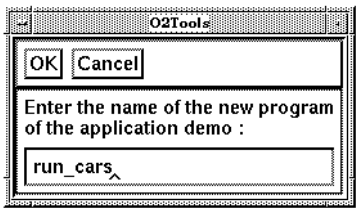


Figure 2.72: Create program dialog box

The created program now appears in the Program list and is currently selected.

Its signature is displayed in the description window prefixed with the comment **`/* uncompiled program */`**.

Figure 2.73: Uncompiled program

The source editor of the new program is displayed in separate window.

Warning !

The program is created but not compiled. To compile a program refer to Section 3.7.

- ***Renaming a program***

  Click on the program name you wish to rename and select Rename. Enter the new name in the dialog box that appears and click on OK to confirm your decision.

  The new name now appears in the Program list and the source editor is refreshed.

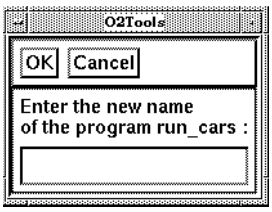


Figure 2.74: Rename program dialog box

- ***Deleting a program***

  Click on the name of the program you wish to delete and select Delete from the Programs menu. A dialog box now appears in which you confirm the deletion.

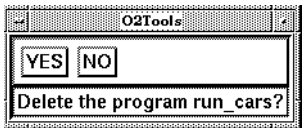


Figure 2.75: Confirm deletion dialog box

The Program name no longer appears in the Program list and the description window is refreshed.

If the program source editor is displayed it is unmapped immediately.

- ***Saving all programs***

  To save all the programs, select Save All from the Programs menu.

Manipulating Application variables

To obtain information about an application variable click on the variable name in the Application variable list in the Application Browser as in (*A*), Figure 2.76.

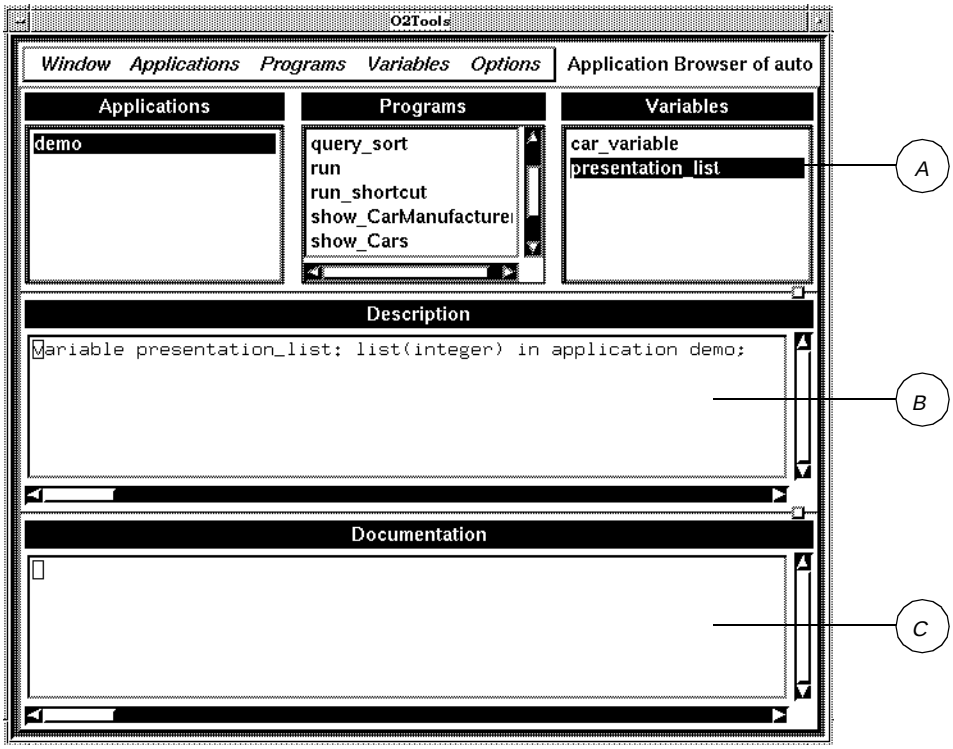


Figure 2.76: Selected Variable

The description is displayed in the description window (*B*) and any associated documentation in the Documentation window (*C*).

The Variables menu, shown in Figure 2.77, allows you to create, delete and rename variables as well as display their source editors.

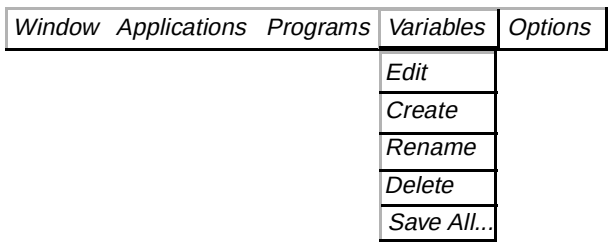


Figure 2.77: Variables menu

This section now details these items.

- ***Displaying an application variable source editor***

  To display an application variable source editor you can either

- select the variable in the Variable list and click on Edit in the Variables menu, or
- double click with the left mouse button on the variable name in the Variable list, or
- click on the variable name using the middle mouse button and drag and drop it onto an already displayed variable editor, see Section 3.1.

A variable source editor is then displayed as in Figure 2.78. For more information on how to use a variable source editor, see Section 3.8.

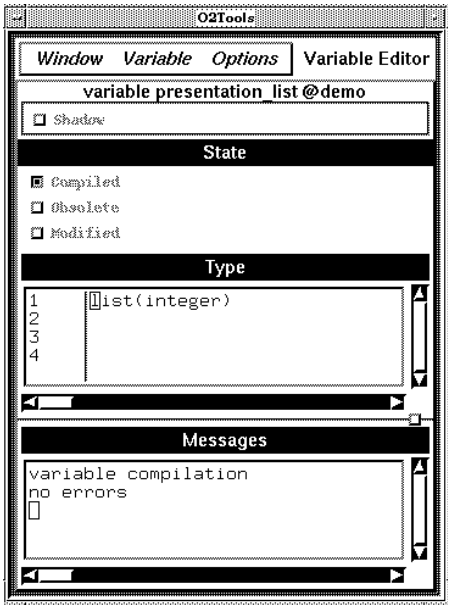


Figure 2.78: Variable source editor

- ***Creating an application variable***

  The variable you create belongs to the currently selected application. Select Create from the Variables menu and enter the name of the new variable in the dialog box. Click on OK.

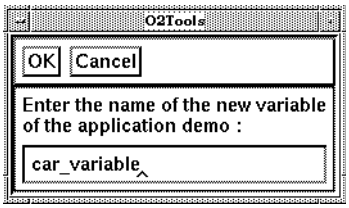


Figure 2.79: Create application variable dialog box

The created variable now appears in the Variable list and is currently selected.

The variable description is displayed in the description window prefixed with the comment **`/* uncompiled variable */`**.

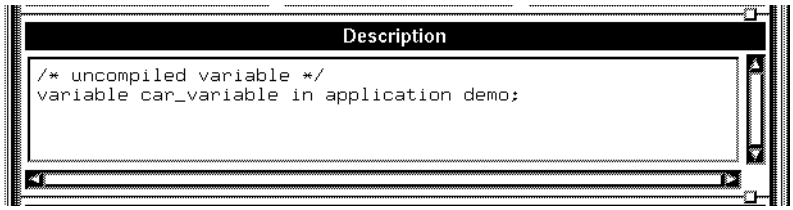


Figure 2.80: Uncompiled Variable

The source editor of the new variable is displayed separately.

Warning !

The variable is created but not compiled. For details on compiling variables see Section 3.8.

- ***Renaming an application variable***

  Click on the application variable name you wish to rename and select Rename. Enter the new name in the dialog box that appears and click on OK to confirm your decision.

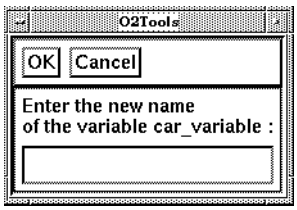


Figure 2.81: Rename variable dialog box

The new name is now displayed in the Variable list and the source editor refreshed.

- ***Deleting an application variable***

  Click on the name of the application variable you wish to delete and select Delete from the Applications menu. Confirm the deletion in the dialog box that appears.

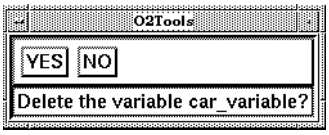


Figure 2.82: Confirm delete command

The application variable name no longer appears in the list and the description window is refreshed.

The displayed application variable source editor, if displayed, is unmapped immediately.

- ***Saving all variables***

  To save all the variables, select Save All from the Variables menu.

2.5 The Function Browser

The Function Browser lets you visualize function definitions and associated documentation. Functions are created and displayed using the Function Browser.

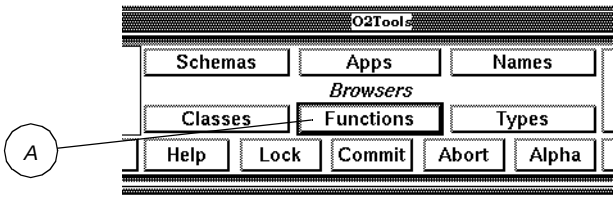


Figure 2.83: Function button

Having previously selected the current working schema, simply click on the Functions button (*A*) on dashboard shown in Figure 2.83 in order to display the Function Browser.

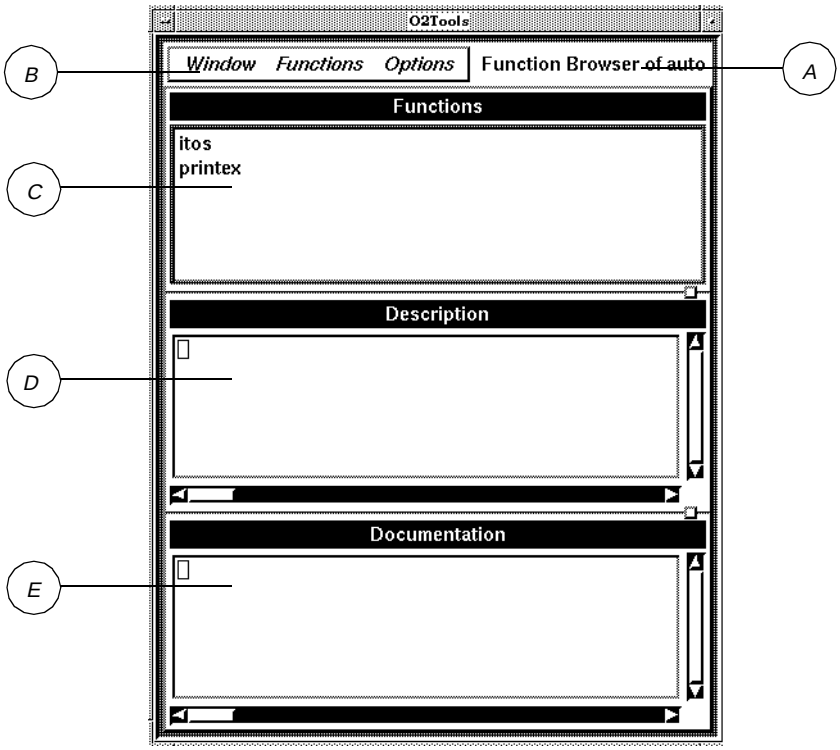


Figure 2.84: Function Browser components

A: Browser title *B: Menu bar*

C: Function list *D: Description with function signature*

E: Documentation/error messages

When you first display the Function Browser, the Function list (C) shows all the available functions of the current working schema. No function is selected.

Manipulating Functions

To obtain information about a function, simply click on its name in the Function list as in (*A*), Figure 2.85.

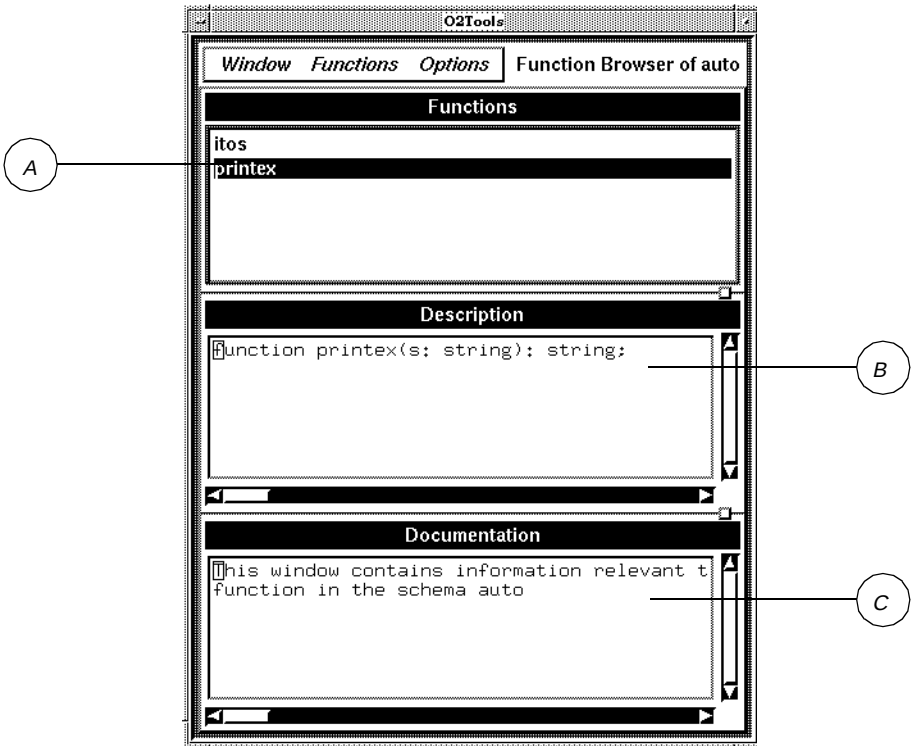


Figure 2.85: Selected function

The signature of the selected function is displayed in the description window (*B*). Any associated documentation is also displayed in the Documentation window (*C*).

With the Function menu, shown above in Figure 2.86, you can display the function source editor as well as create, delete and rename functions.

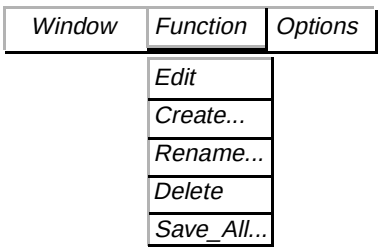


Figure 2.86: Function menu

The rest of this section details these items.

- ***Displaying the function source editor***

  To display a function source editor, as in Figure 2.87, you can either

  - select the function in the Function list and click on Edit in the Function menu, or
  - double click with the left mouse button on the function name in the Function list, or
  - click on the function name using the middle mouse button and drag and drop it onto an already displayed function source editor, see Section 3.1.

  See Section 3.9 for compilation details.

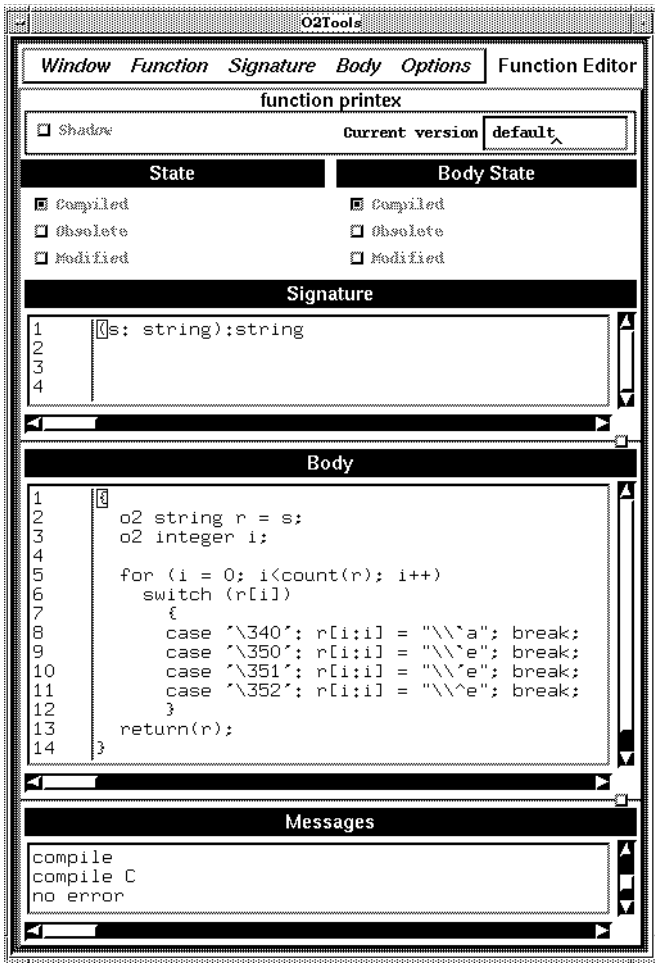


Figure 2.87: Function source editor

- ***Creating a function***

  Select Create from the Function menu and enter the name of the new function in the dialog box that is displayed.

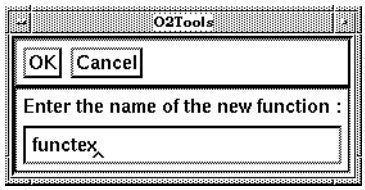


Figure 2.88: Create function dialog box

The newly created function is now listed in the browser and is the currently selected.

The function default signature is displayed in the description window prefixed with the comment **`/* uncompiled function */`** as in Figure 2.89.

The new function source editor is displayed separately.

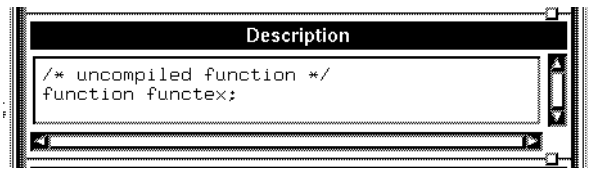


Figure 2.89: Uncompiled Function

- ***Save all functions***

  To save all the functions, select Save All from the Function menu.

Warning !

The function source is created but not compiled. For more information, on compiling functions refer to Section 3.9.

- ***Renaming a function.***

  To rename a function, click on the function name and select Rename from the Function menu. Enter the new name in the dialog box that appears and click on OK.

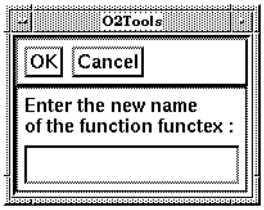


Figure 2.90: Renaming a function

The function is renamed and the new name appears in the Function list.

If the source editor of the renamed function is displayed, it is immediately refreshed with the new name.

- ***Deleting a function***

  Select the function name you wish to delete and select Delete. A dialog box appears in which you confirm the deletion of the function.

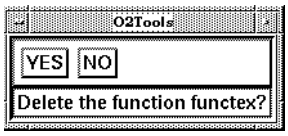


Figure 2.91: Confirm deletion

The function name is deleted from the list and no function is selected. The function source editor, if displayed, is unmapped.

2.6 The Persistent Type Browser

With the Persistent type Browser, you visualize persistent type definitions and associated documentation. Persistent types are created and displayed using the Persistent type Browser.

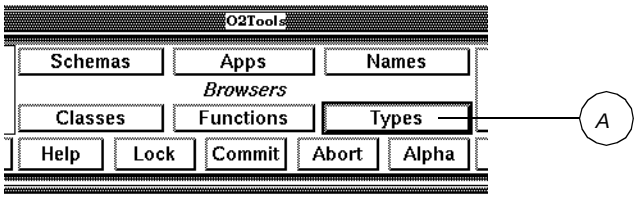


Figure 2.92: Persistent Type button

To display the Persistent Type Browser click on the Types button (*A*) on the dashboard, shown in Figure 2.92. The following type of window appears.

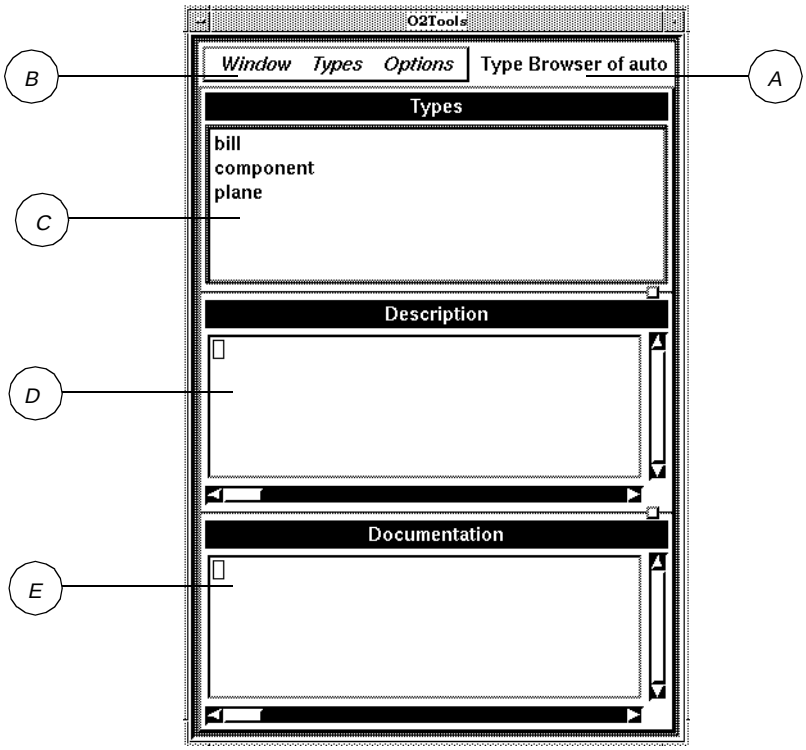


Figure 2.93: Persistent Type Browser components

A: Browser title *B: Menu bar*

C: Persistent Type list *D: Description*

E: Documentation/error messages

When you first display the Persistent Type Browser, the Persistent type list (*C*) shows all the available persistent types of the current working schema. No persistent type is selected.

Manipulating Persistent types

To obtain information about a persistent type, simply click on its name in the Persistent type list as in (*A*) Figure 2.94.

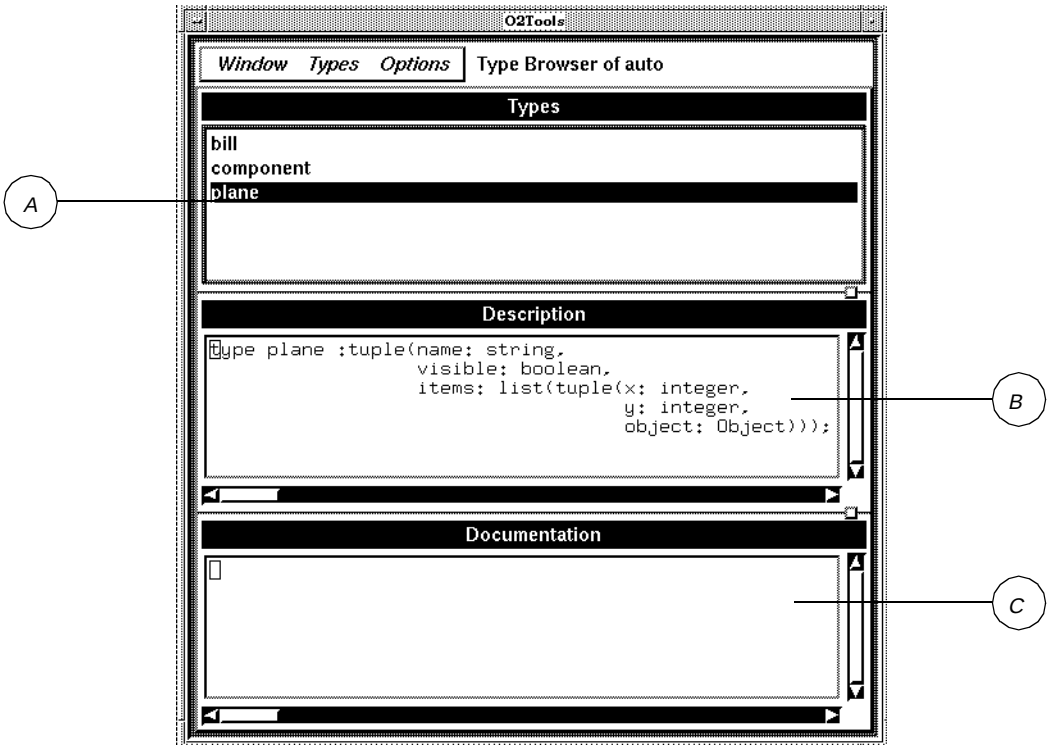


Figure 2.94: Selected Type

The description of the selected persistent type is displayed in the description window (*B*). Any associated documentation is also displayed (*C*).

With the Type menu, shown in Figure 2.95, you can display the persistent type source editor as well as create, and delete persistent types.

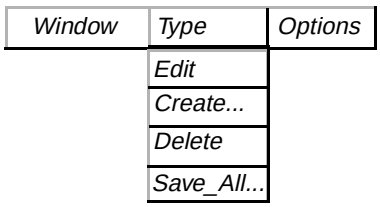


Figure 2.95: Type menu

This section now details these items.

- ***Displaying the persistent type source editor***

  To display a persistent type source editor, as in Figure 2.96, you can either

  - select the persistent type in the Type list and click on Edit in the Type menu, or
  - double click with the left mouse button on the persistent type name in the Type list, or
  - click on the persistent type using the middle mouse button and drag and drop it onto an already displayed persistent type source editor, see Section 3.1.

  See Section 3.10 for compilation details.

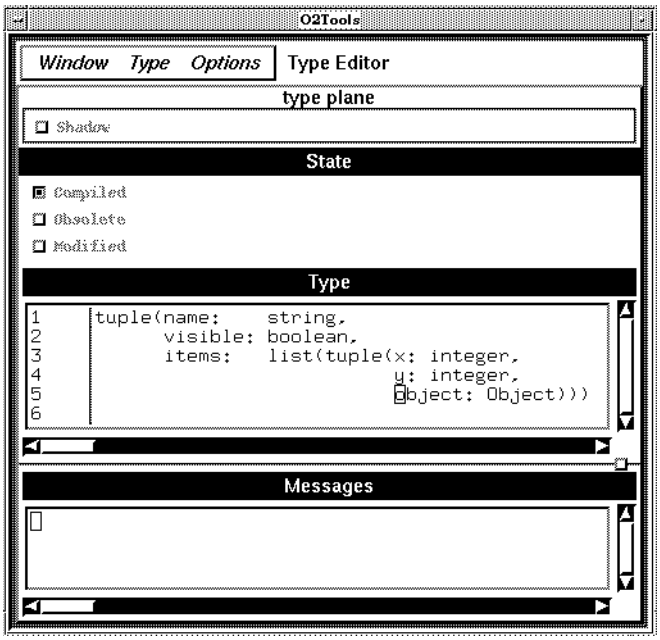


Figure 2.96: Type source editor

- ***Creating a persistent type***

  Select Create from the Type menu and enter the name of the new persistent type in the dialog box that is displayed.

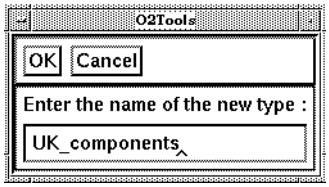


Figure 2.97: Create persistent type dialog box

The newly created persistent type is now listed in the browser and is the currently selected. Its default definition is displayed in the description window prefixed with the comment **`/* uncompiled type */`** as in Figure 2.98 below.

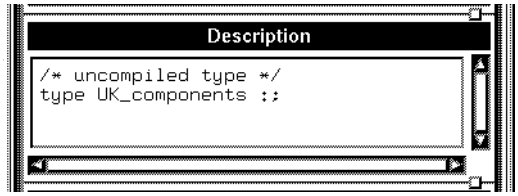


Figure 2.98: Uncompiled Type

The new persistent type source editor is displayed separately.

Note

The persistent type source is created but not compiled. For more information on compiling sources refer to Section 3.10.

- ***Deleting a persistent type***

  Select the persistent type name you wish to delete and select Delete. A dialog box then asks for confirmation.

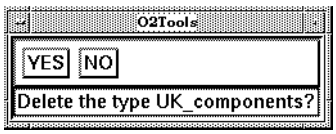


Figure 2.99: Confirm deletion

The persistent type name is deleted from the list and no persistent type is selected. The persistent type source editor, if displayed, is unmapped.

- ***Saving all types***

  To save all the types, select Save All from the Type menu.

2.7 The Persistent Name Browser

With the Persistent Name Browser, you visualize persistent name definitions and any associated documentation as well as create, delete and rename persistent names, display persistent name source editors and access the database in order to display and modify data.

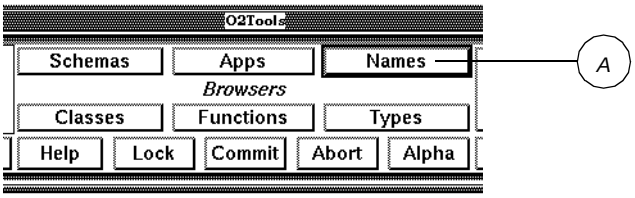


Figure 2.100: Persistent Name button

To display the Persistent Name Browser click on the Names button (*A*) on the dashboard shown in Figure 2.100. The following browser now appears.

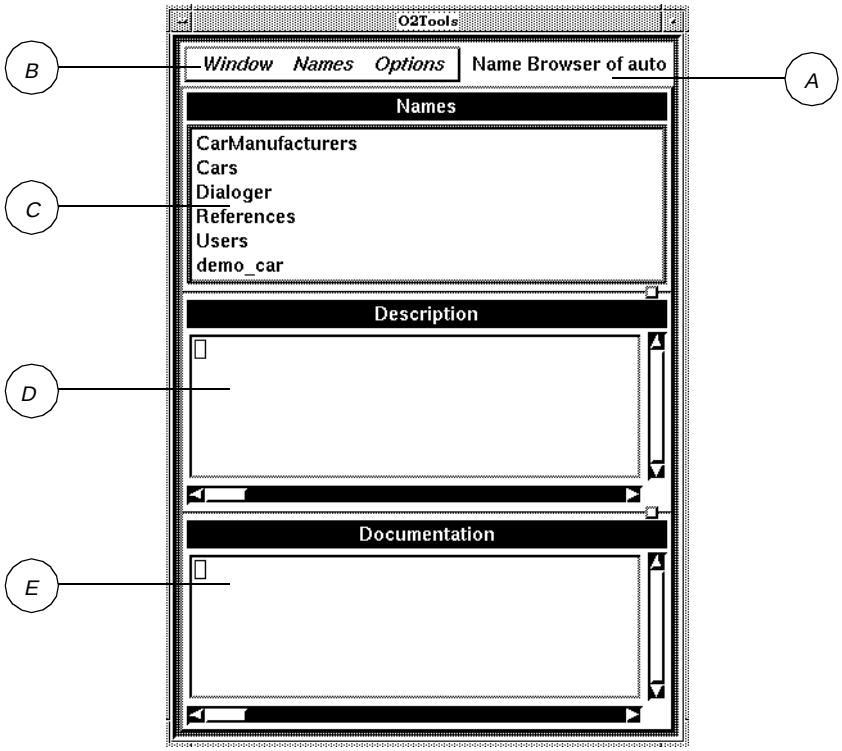


Figure 2.101: Persistent Name Browser components

A: Browser title *B: Menu bar*

C: Persistent Name list *D: Description*

E: Documentation/error messages

When you first display the Persistent Name Browser, the persistent name list (C) shows all the available persistent names of the current working schema. No persistent name is selected.

Manipulating Persistent names

To get information about a persistent name, click on the name that interests you in the Persistent Name list, shown in Figure 2.102 (*A*).

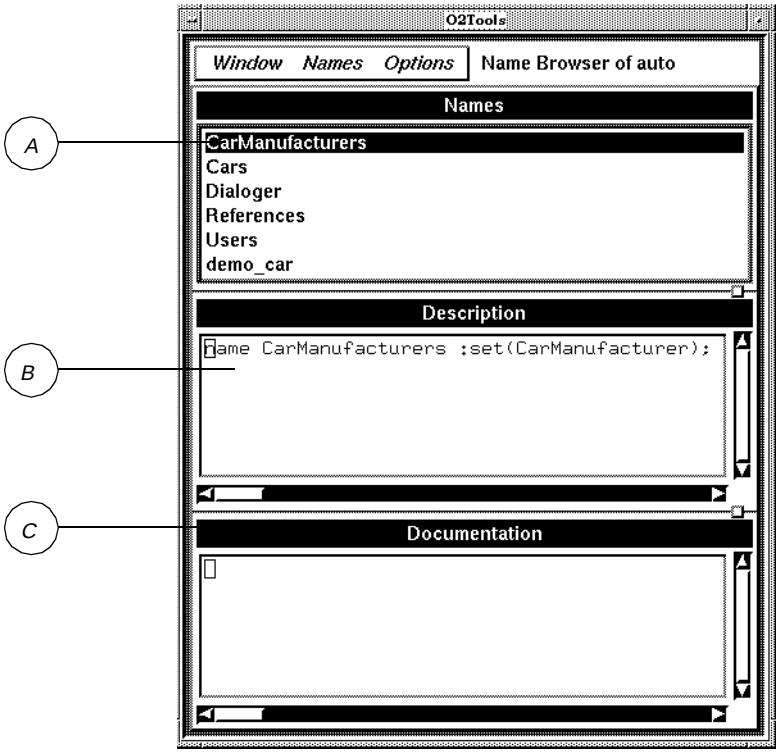


Figure 2.102: Selected persistent name

The description of the selected persistent name is displayed in the description window (*B*).

Any associated documentation is displayed in the Documentation window (*C*).

With the Names menu, shown in Figure 2.103, you can display the persistent name source editor as well as create, rename and delete persistent names.

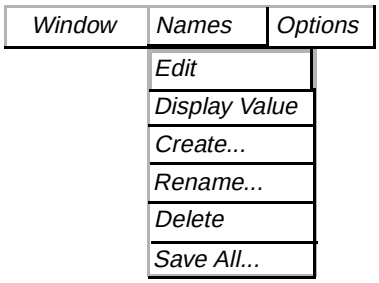


Figure 2.103: Name menu

The rest of this section now details these items.

- ***Displaying a persistent name source editor***

  To display a persistent name source editor, as in Figure 2.104, you can either

  - select the persistent type in the Name list and click on Edit in the Name menu, or
  - double click with the left mouse button on the persistent name in the Name list, or
  - click on the persistent name using the middle mouse button and drag and drop it onto an already displayed persistent name source editor, see Section 3.1.

  To use the persistent name source editor, see Section 3.11.

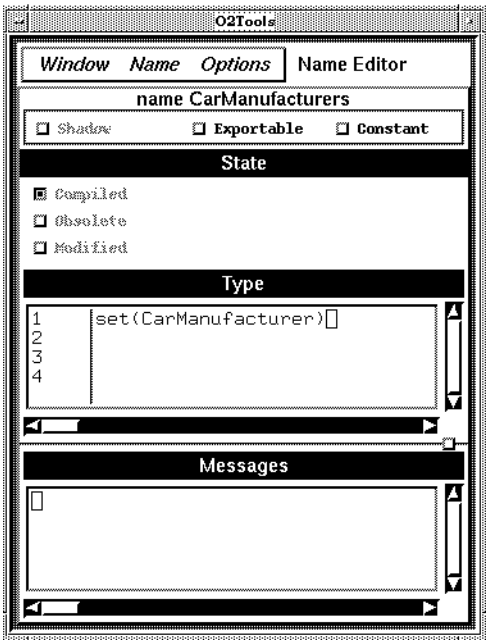


Figure 2.104: Name source editor

- ***Displaying the value of a name***

  To display the value of a name select the name and click on Display value. The value of the name now appears in a separate window. For example, the value of the name CarManufacturers is as follows:

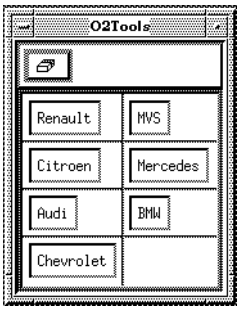


Figure 2.105: Name value

- ***Creating a persistent name***

  Select Create from the Name menu and enter the name of the new persistent name in the dialog box that is displayed.

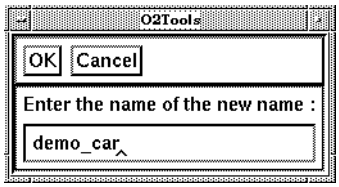


Figure 2.106: Create persistent name dialog box

The newly created persistent name is now listed in the browser and is the currently selected. Its default definition is displayed in the description window prefixed with the comment: **`/*uncompiled name*/`** as in Figure 2.107. The new persistent name source editor is displayed separately.

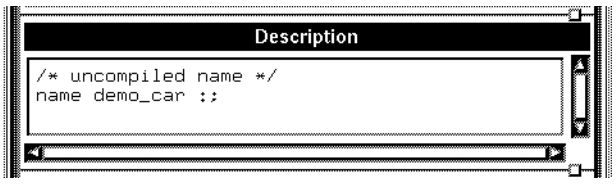


Figure 2.107: Uncompiled Name

Warning !

The persistent name source is created but not compiled. For more information on compiling sources, refer to Section 3.11.

- ***Renaming persistent name***

  To rename a persistent name, click on the persistent name and select Rename from the Name menu.

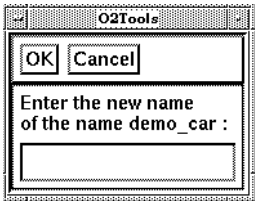


Figure 2.108: Rename persistent name dialog box

Enter the new name in the dialog box that appears and click on OK. The persistent name is renamed and the new name appears in the Names list.

If the source editor of the renamed persistent name is displayed, it is immediately refreshed with the new name.

- ***Deleting a persistent name***

  Select the persistent name you wish to delete and select Delete. A dialog box appears in which you confirm the deletion.

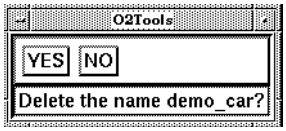


Figure 2.109: Confirm deletion

The persistent name is deleted from the list and no persistent name is selected.

The persistent name source editor, if displayed, is unmapped.

- ***Saving all names***

  To save all the persistent names, select Save All from the Names menu.

2 Browsers

3 Programming Environment

This chapter is aimed at the experienced O_2 programmer. You are accustomed to using O_2 and are familiar with the vocabulary used in the documentation. If this is not the case, you should refer to the *O_2C Reference Manual* at all times.

This chapter on the O_2Tools programming environment is divided into the following sections:

3 Programming Environment

Note

This chapter explains what you see in O_2Tools if you have the Global Utilization option set to Novice i.e. all the possible confirmation dialog boxes are described. Refer to Section 1.4 for details.

3.1 Source editors - overview

In the O_2Tools programming environment you edit, compile, test and debug different sources.

As seen in Chapter 2, Class and Method Source Editors are called up from the Class Browser; Program and Application Variable Source Editors from the Application Browser, and the Function, Persistent type and Persistent Name Source Editors from their respective browsers.

In all the source editors you edit and compile any modifications and write any relevant and associated documentation you want to appear in the browsers.

Dependencies and body versions of methods, programs and functions are managed in their source editors.

Common features

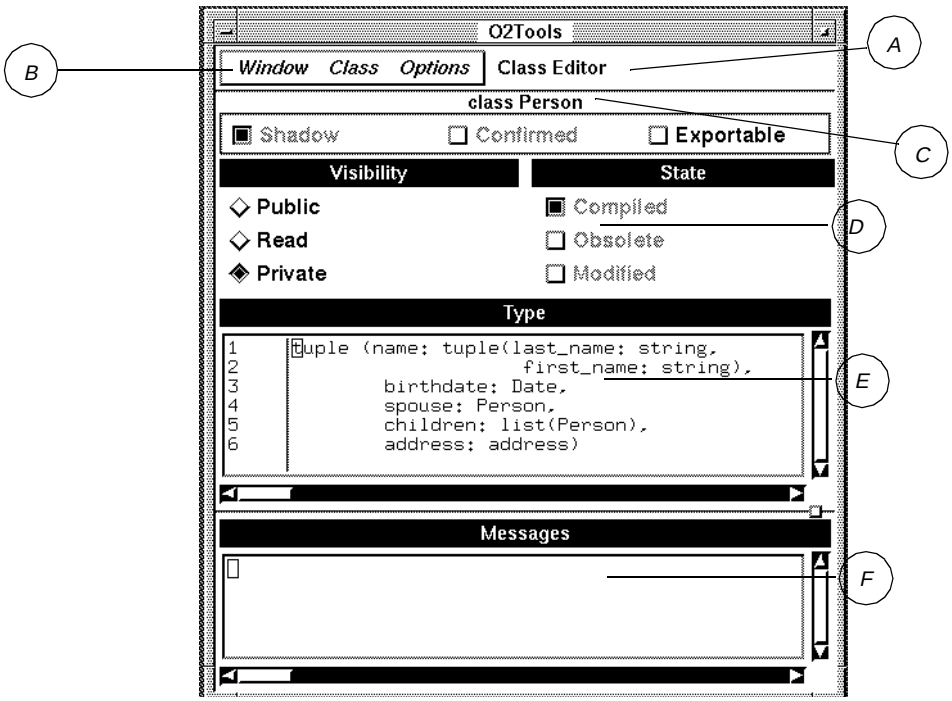


Figure 3.110: A typical O_2Tools source

A: Type of Source editor
B: Menu bar
C: Sensitive title area
D: Current state
E: Modifiable text window
F: Messages window

All source editor display their type (*A*) and current state (*D*). There is a menu bar (*B*), a “sensitive” title area (*C*) used in the drag and drop facility to display editors and an error messages window (*F*).

The modifiable text window (*E*) is called the type window in the Class, Variable, Name and Type source editors, and is divided up into a Signature and a Body window in the Method, Function and Program source editors. In these windows you can use emacs like commands to help you program. These features are discussed in detail in the following sections.

Note

Two stars ** before the title as in Figure 3.111 (*A*) means that the source has been interactively modified but not saved.

The stars disappear when the source is stored or when the source is compiled. A compilation implicitly stores the source.

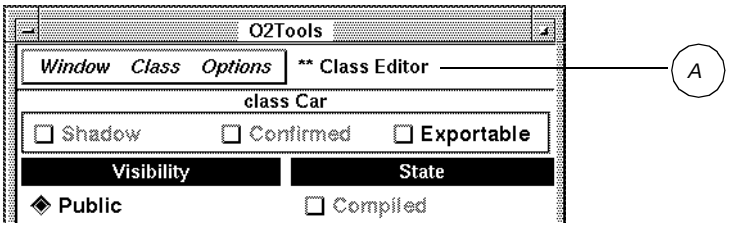


Figure 3.111: Modified but unsaved editor

The rest of this section explains in detail common features of source editors:

- Common menus
- Common source menu items
- Manipulating text
- Drag and drop display facility
- Current state of source editor
- Error messages window
- Entering source documentation

Common menus

The menu bar entries Window and Options appear in every O_2Tools source editor.

- ***Window menu***

The Window menu has two entries as in Figure 3.112.

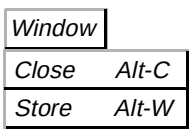


Figure 3.112: Window menu

The Close item allows you to leave the source editor at any time as with the browser Window menu described in Section 2.1.

Use the Store item if you want to save any modifications you have carried out but you do not want to compile them immediately. Please

note that anymodification is saved in the database after the transaction has been validated

Note

The Source editor menus have various keyboard accelerators using the Alt key and various other letters.

- ***Options menu***

  The Options menu contains Toggle options. The display of each window can be toggled on and off enabling you to concentrate on a particular part of the source editor.

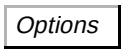


Figure 3.113: Options menu

- ***Accelerators***

  As with browsers (see Section 2.1), there are various keyboard accelerators.

  Accelerators are marked to the right of the menu item and select and invoke the associated function without displaying the menu, e.g. Alt-C in Figure 3.112 for the item Close.

Common source menu items

Each source menu: class, method, program, variable, function, type and name always includes the items shown in Figure 3.114 below.

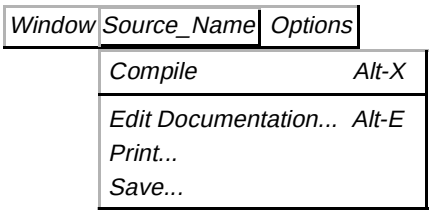


Figure 3.114: Common items

- ***Compile***

  This item enables you to compile any modifications. When you carry out any modification you must edit the source and then recompile. This item is described in detail for each source editor in the sections that follow.

- ***Print***

  Use this item to print the description of a source into a file.When you select Print, a dialog box appears in which you choose the file you want the result to be stored in. This item is the same as the alphanumeric command **`print`**.

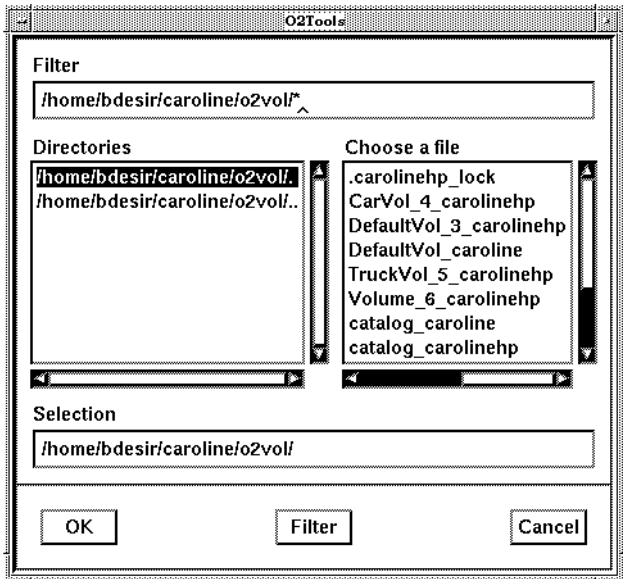


Figure 3.115: Dialog box

- ***Save***

  To save the source in a Unix file, select the item Save. You now see the same window as in Figure 3.115 above.

- ***Entering source documentation***

  Use this item to enter any relevant information about the source you are handling.

  If you select Edit Documentation in the menu of the source editor you are working in, a text editor is displayed as in Figure 3.116.

  Enter the text you want and save by clicking on Store in the Window menu. You now see the text appear in the corresponding browser.

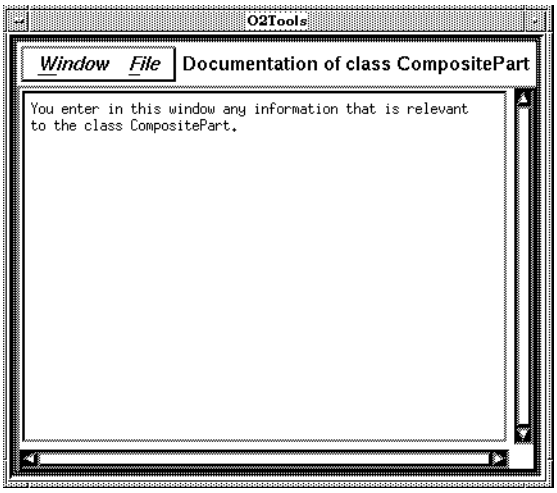


Figure 3.116: Entering documentation

You can also load a file by selecting Open or Append from the File menu or save the contents into a file by clicking on Write As.

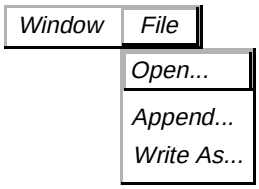


Figure 3.117: Documentation Editor File menu

When you click on Open, Append or Write As the dialog box shown in Figure 3.118 appears.

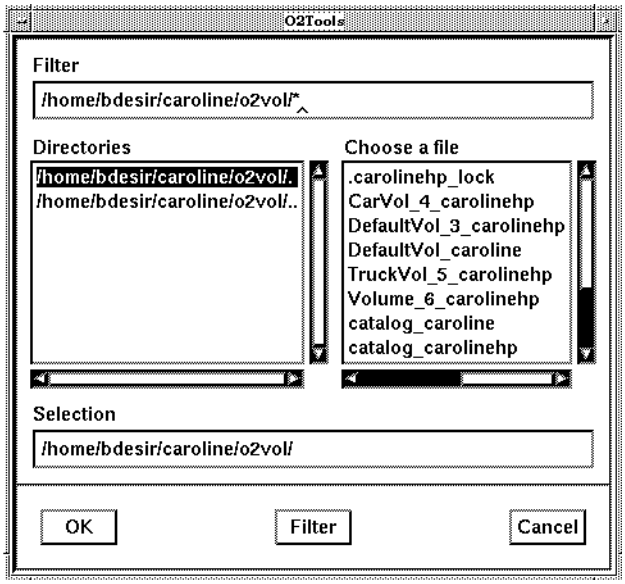


Figure 3.118: Dialog box

When you select the Open item in the File menu, shown in Figure 3.117, you can load an ASCII file into the documentation editor.

Simply enter the name of the file in the Selection box and click on OK. The file contents are now displayed in the documentation editor.

With the Append item, you add on the contents of an ASCII file to what is already in the documentation editor.

Enter the file name in the Selection box and click on OK. The file contents are immediately added onto the contents of the documentation editor.

Use the Write As item to save the contents of the documentation editor in an external file.

Enter the name of the file in which you want to save the documentation and click on OK.

Drag and drop display facility

In order to limit the number of windows you have on display, O_2Tools has a Drag and drop facility that enables you to display different sources using the same source editor.

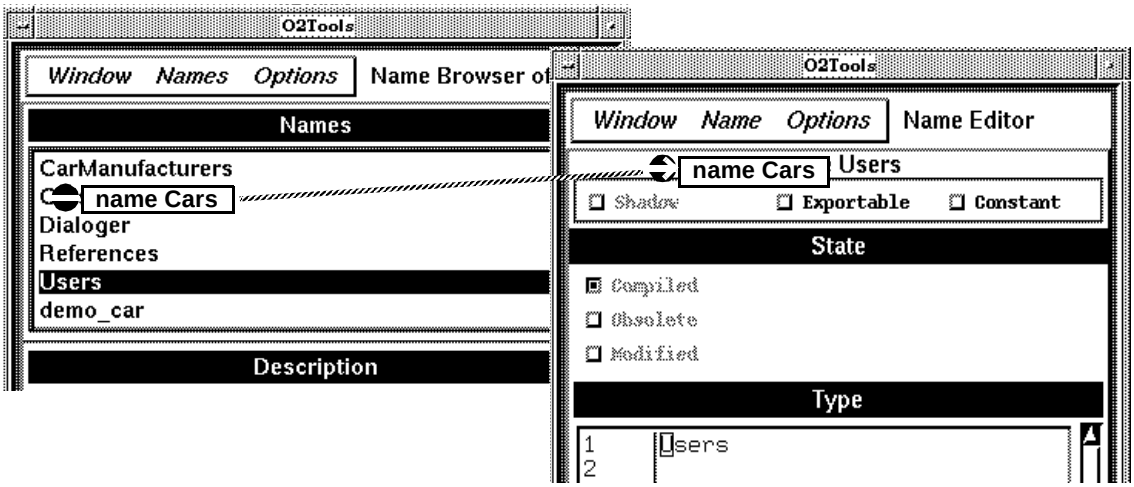


Figure 3.119: Drag and drop facility

If you want to change the source on display, in the browser simply click using the middle mouse button on the name of the source you want to display and keep the button pressed.

A box, showing the source name and a no entry sign, appears as in Figure 3.119. You can now drag this box to the source editor.

Place the box over the sensitive source name area of the editor until the sign changes to an arrow and release the button.

The editor now displays the source you have chose and you can begin to work on it.

Note

You can only use this facility if the source editor type corresponds to that of the browser, i.e a name browser with a name source editor.

Current Source Editor State

The state column of the source editor enables you to see at all times the current state of the source editor.

It has three flags that cannot be modified: Compiled, Obsolete and Modified.

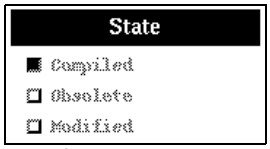


Figure 3.120: Source state column

When the source saved is different from the compiled source, the Modified flag is highlighted.

When you successfully compile a source, the Compiled flag is highlighted and the flag modified is turned off as in Figure 3.120.

When the schema is modified and the source can no longer be compiled the Obsolete flag is highlighted.

Messages

If an error occurs, the error message is displayed in the message window.

Compilation messages are displayed in the messages window of each source editor.

You see whether the source has been compiled correctly or if any errors have occurred. In Figure 3.121 there is no error shown.

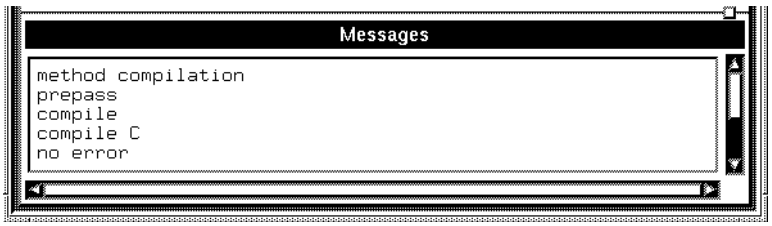


Figure 3.121: Messages window

3.2 Body - overview

All the characteristics described above exist for all the source editors. However, the Method, Program and Function editors are different from the other Class, Type, Variable and Name editors in that they have a signature and a body window as shown in Figure 3.122, instead of just a type window (see Figure 3.110).

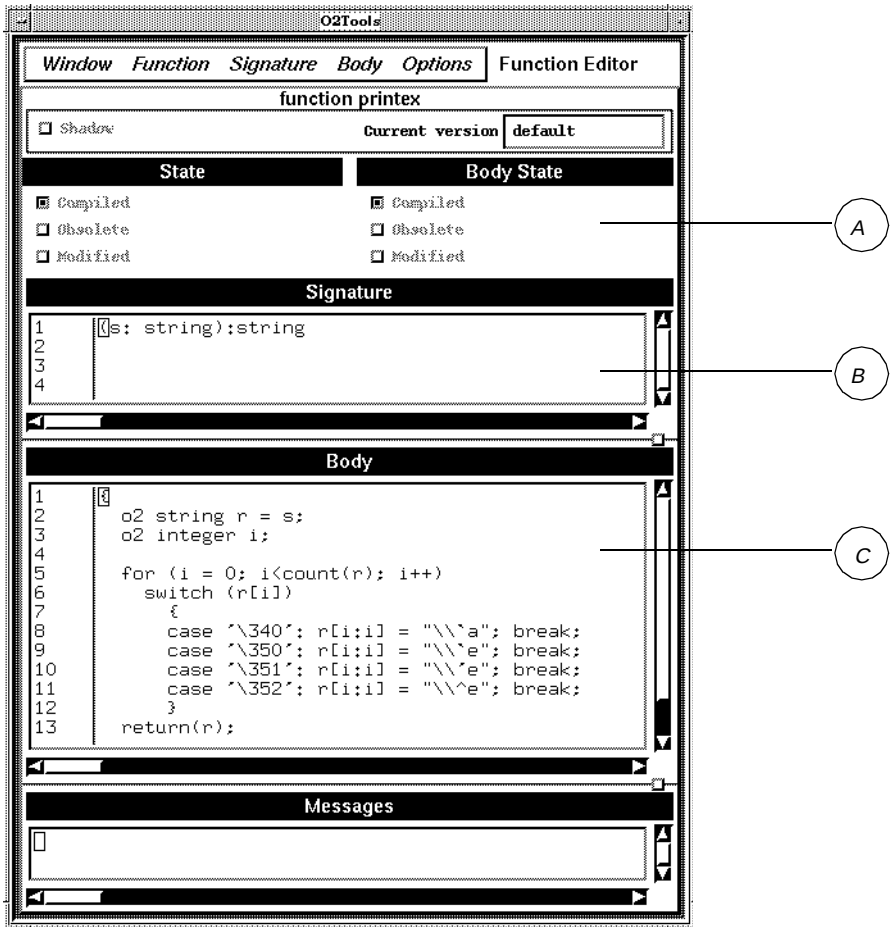


Figure 3.122: Typical editor with body and signature

A: Body State *B: Signature window*
C: Body window

This section now details the following:

- Body State
- Common menu items
- Body compilation error messages
- Body options

- Body versions

Body State

The Body state column only exists in the Method, Program and Function Source Editors.

When you edit and compile a method, program or a function you can compile just the body or just the signature or both at the same time. This is explained fully in each of the sections about these particular sources.

The Body state column displays at all times the current state of the source body.

As with the state column, there are three flags that cannot be modified: Compiled, Obsolete and Modified.

The combination of these flags have the following meanings:

- ***compiled: true, obsolete: false, modified: false***

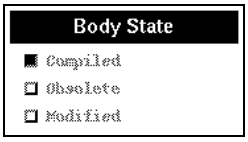
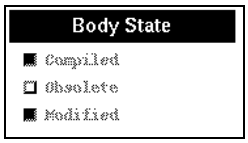


Figure 3.123: Body state column

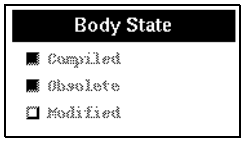
The source body is compiled and the source corresponds to the binary.

- ***compiled: true, obsolete: false, modified:true***



The source body is compiled but you have interactively modified and stored the source and it no longer corresponds to the binary.

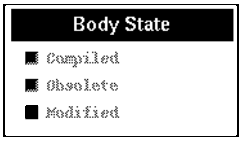
- ***compiled: true, obsolete: true, modified: false***



After schema modification, the source no longer corresponds to the binary and can not be compiled.

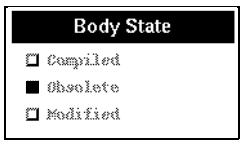
This occurs when you work with the dependence option “free”.

- ***compiled: true, obsolete: true, modified: true***



The source body is compiled but after schema modification is no longer compilable and you have also interactively modified and stored the source.

- ***compiled: false, obsolete: true, modified: false***

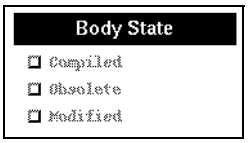


The source body is not compiled but after schema modification is no longer compilable.

You encounter this when you work with the dependence option “consistent”.

The binary is automatically uncompiled after the schema modification.

- ***compiled: false, obsolete: false, modified: false***



The source body is not yet compiled and you have not yet interactively modified the source.

Note

You visualize dependencies and launch automatic recompilation using the O_2 Shell. See Section 3.12.

For more information about dependencies see the Section on O_2C Compiler options in the *O_2C Reference Manual*.

Common menu items

All Body menus are as in Figure 3.124.

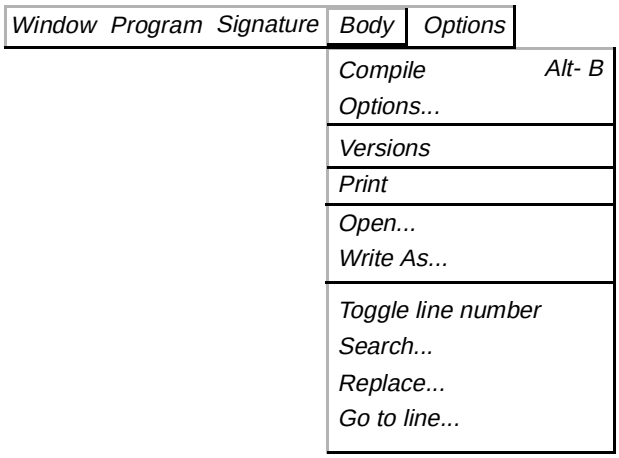


Figure 3.124: Body menu

Body compilation error messages

As with every source editor, the Compile item enables you to compile any modifications you have carried out in the body of your method, program or function. This item is described in detail for each source editor in the sections that follow.

When you compile the body of a method, program or function (refer to Sections 3.6, 3.7, and 3.9), O_2Tools lists the compilation errors in the Messages window of the Source editor in question.

This error list, shown in (*A*) in Figure 3.125, is in fact a selectable list. When you click on the error message that you are interested in, the cursor in the body window shows the location of the error as in (*B*).

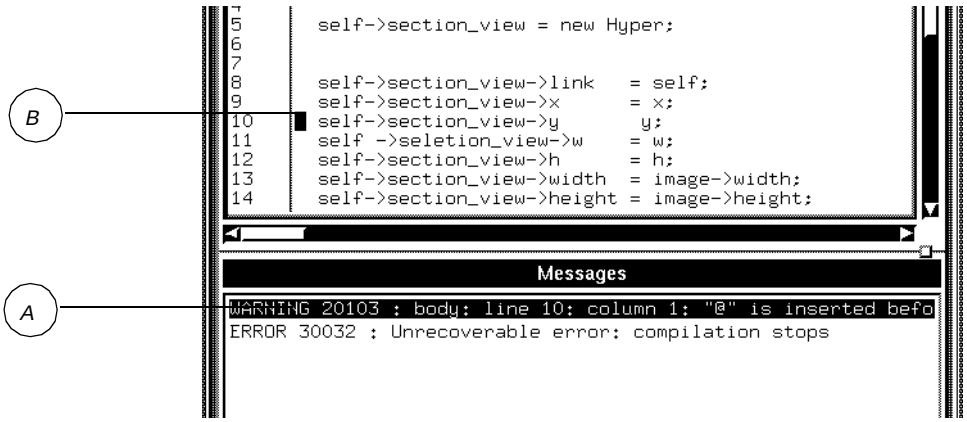


Figure 3.125: Selectable error list

Body Compilation options

When you compile the body of a method, program or function (refer to Sections 3.6, 3.7, and 3.9), you can also define local compilation options for the method, program or function body. Select Options from the Body pull-down menu. The dialog box shown below in Figure 3.126 appears :

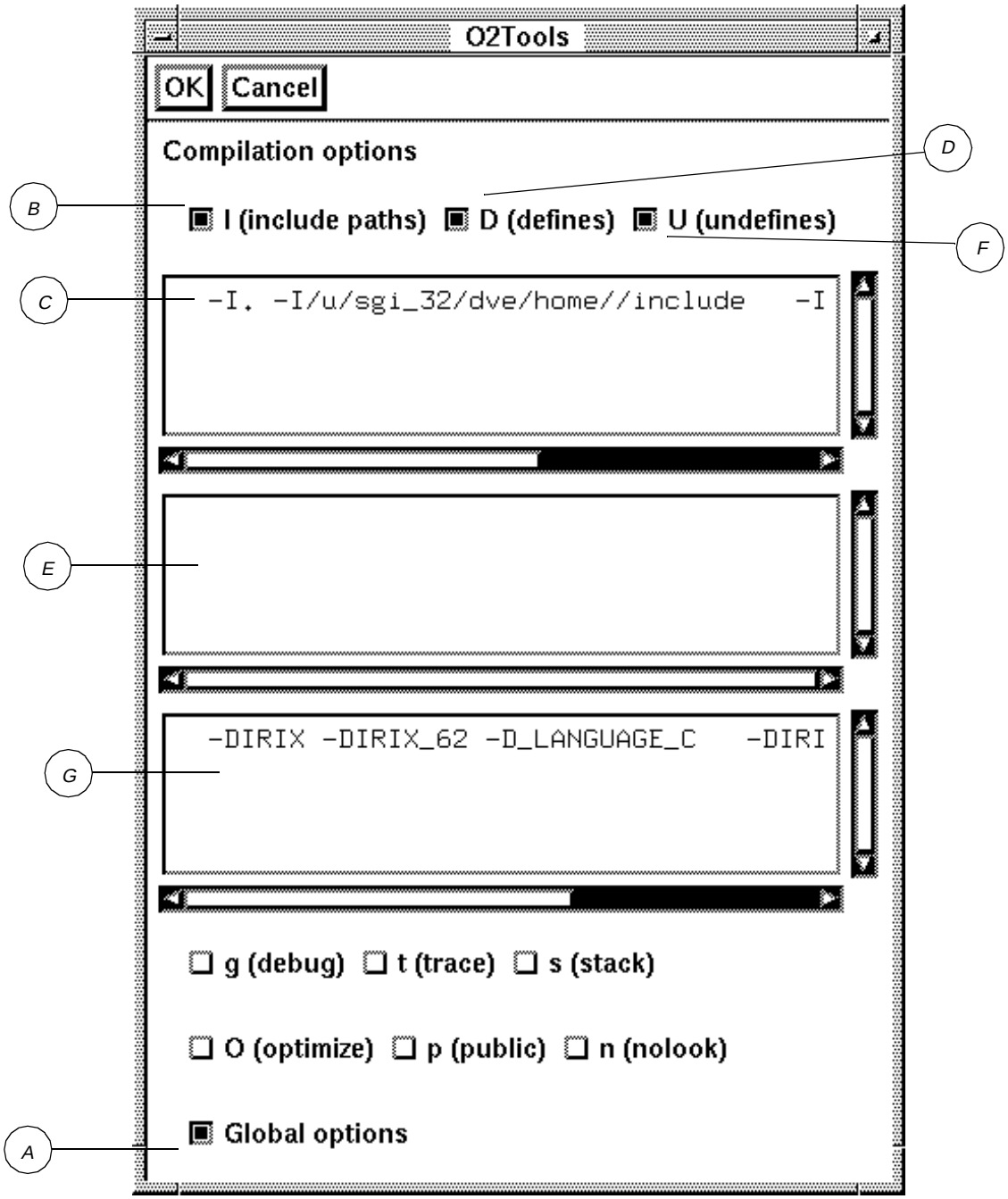


Figure 3.126: Compilation options

If no local options already exist, the dialog box is initialized to the Global options button (*A*). Global options are detailed in Section 3.13.

To create local options, you must first deselect Global options.

To enter include paths, click on the I (include path) button (*B*) and enter one or more paths in the window. Each path must be prefixed by -I as shown in Figure 3.126 (*C*).

To enter #define directives, click on the D (defines) button (*D*) and enter the names, each preceded by -D in the window (*E*).

To enter #undefine directives, click on the U (undefines) button (*F*) and enter the names, each preceded by -U in the window (*G*).

If you want to add any of the set options: g (debug), t (trace), s (stack), O (optimize), p (public), or n (nolook) simply click on the relevant button(s). For details on these compilation options, refer to the *O_2C Reference Manual* on Data Definition.

To delete the local options and return to using the global options, click on the Global options button and then on OK.

Important

Local options persist from one O_2Tools session to another.

Versions of the source body

You can create, delete and compile different versions of the source body using the version editor. From the Body menu select Versions.

A version editor is displayed separately as shown in Figure 3.127 below.

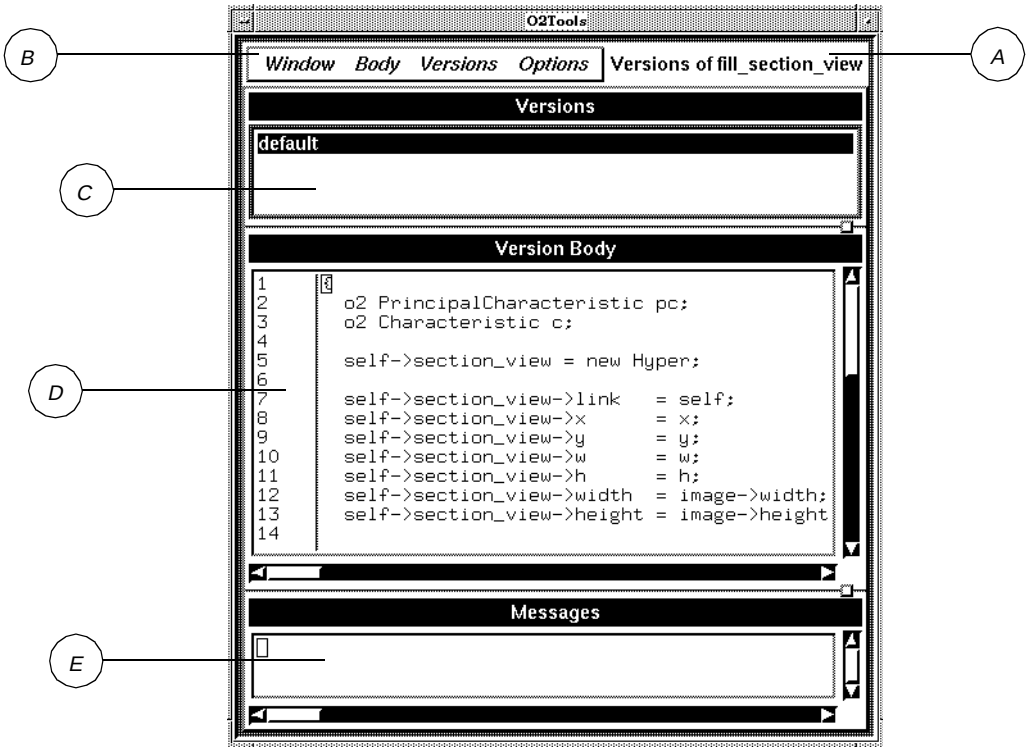


Figure 3.127: Version editor components:

A: Version editor title containing method name

B: Menu bar

C: List of versions available

D: Body window

E: Messages window

All the available versions of the source body are displayed in the version list (*C*). You edit or modify the version of the source body in the body window (*D*).

This section now describes how to create, delete and compile different versions of the source body using this version editor.

- ***Creating a version***

To create a new version of a source body, pull down the Versions menu, shown in Figure 3.128 and select Create.

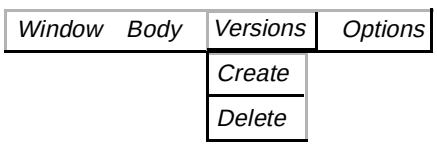


Figure 3.128: Versions menu

Enter the name of the new version in the dialog box that appears and click on OK.

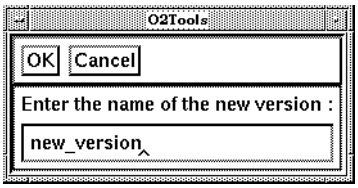


Figure 3.129: Create version dialog box

The version is created and the version editor body window is refreshed. The new version name now appears in the version list. You now can enter in the body window the body of the new version.

- ***Deleting a version***

To delete a version, select the version name from the version list and select the item Delete from the Versions pull-down menu. The version is deleted and the name no longer appears in the version list.

Warning !

The current compiled version cannot be deleted.

- ***Compiling a version***

To compile a version, select the version name in the version list and select Compile from the version editor Body pull-down menu shown in Figure 3.130. The version is compiled and any error message is displayed in the version editor Messages window.

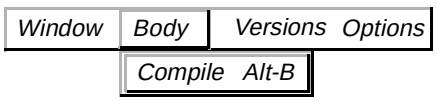


Figure 3.130: Body menu of version editor

Printing a body

Use the Print item in the Body menu to “print” just the body in an external file as with the alphanumeric command print. For example:

```
print method body fill_section_view in class Car “path_file”
```

When you select Print, the dialog box, shown in Figure 3.131, appears in which you choose the file you want the result to be stored.

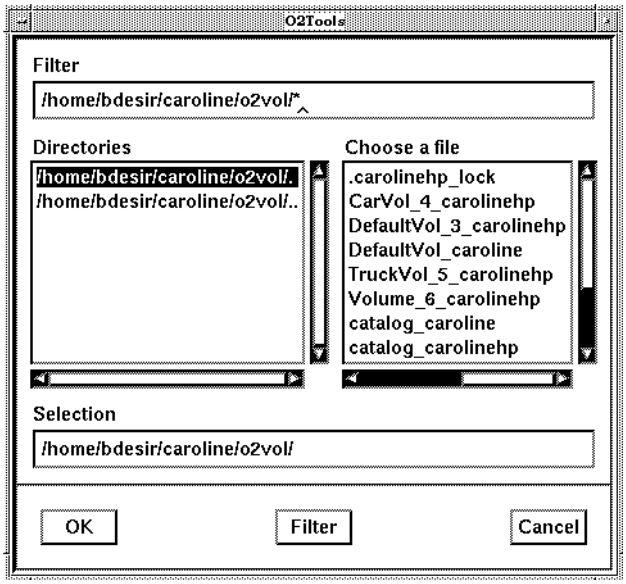


Figure 3.131: Print dialog box

Opening a body

Open allows you to modify the Body using the contents of an external file. When you click on Open, the same dialog box shown in Figure 3.131, appears in which you select the file you want to use. Click on OK to confirm your choice and the contents of the Body window now contain the contents of the file you have just chosen.

Saving a body

Write As enables you to save the Body contents in an external file. Click on Write As and the dialog box in Figure 3.131 is displayed. Enter the file name and click on OK. The Body contents are now saved in that file.

Text Editing

All the text editing facilities: Toggle, Search, Replace and Go To are described in Section 3.4.

3.3 Signature - overview

The Signature menu is composed of the following items:

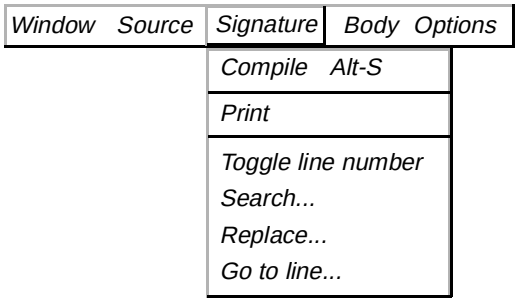


Figure 3.132: Typical Signature menu

Compile

Use this item to compile any modifications carried out in the Signature window. For any modification, you must edit the source and then recompile. This item is detailed for each separate source signature.

Print

Use this item to print the source signature description into a file as you would print a body. When you select Print, a dialog box appears in which you choose the file in which you want the result to be stored.

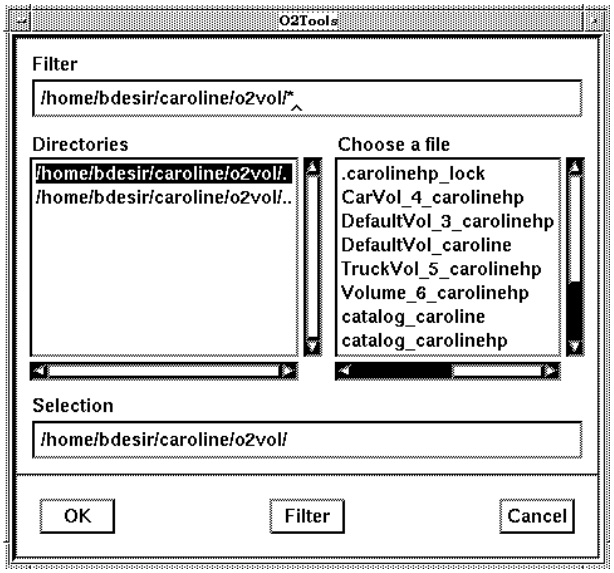


Figure 3.133: Print dialog box

Text Editing

All the text editing facilities: Toggle, Search, Replace and Go To are described in Section 3.4.

3.4 Manipulating text

In all the signature and body menus, and in the Class, Variable, Name and Type editors you see the following menu items. These items are part of the text editor available with O_2Tools which greatly simplifies entering code.

.

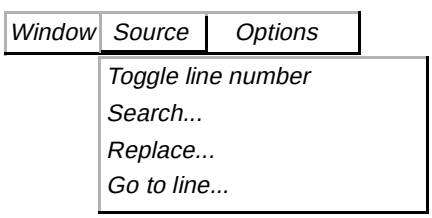


Figure 3.134: Text items

- ***Toggle line number***

  Remove or display the lines at the side of the text in the type, body or signature windows.

- ***Search***

  Search for a particular word in the text window. The box shown in Figure 3.135 appears.

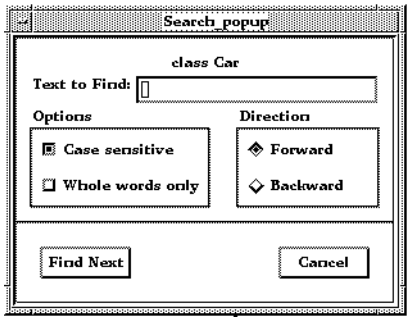


Figure 3.135: Search dialog box

  Type in the word you are looking for and click on the Find Next button. The word is then selected in the text window. You can stipulate the direction in which you want to search by clicking on the Forward or Backward direction button.

  You can also specify whether you want the search to take the letter case into account and whether to search for the whole word. If this last option is not selected the text editor highlights the word if it is part of another word.

  When you have finished click on Cancel.

- ***Replace***

  Replace is similar to Search except that it enables you to replace the selected word. If you select the Replace item, the dialog box, shown in Figure 3.136, appears.

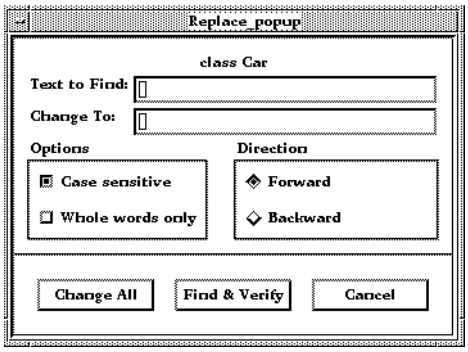


Figure 3.136: Replace dialog box

Type in the word or string of words you want find in the Text to Find section, and then type in the replacement word or words in the Change To section. Select the options you want to specify as for the Search item described above. If you want to change all the words in the text window click on the Change All button.

However if you want to check that you really want to change a particular occurrence of the selected word, click on the Find and Verify button. The text editor stops each time it changes a word so that you can verify that the change is correct.

When you have finished click on Cancel.

- ***Go to line***

  To go to a particular line in the text, select the Go to Line item and type the line number in the box that appears.

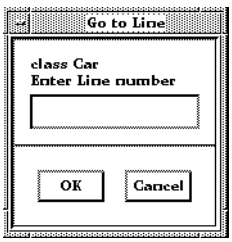


Figure 3.137: Go to line dialog box

Click on OK. The cursor in the text window is placed at the correct line.

- ***Text Editor commands***

  In Table 1 below are listed all the Text Editor commands available to you.

Table 1. Alphabetical list of Text Editor commands

Command	Function	Description
^B	backward-char	move cursor back one character
^P	backward-line	move cursor back one line
Esc v	backward-page	move up one screen
Esc b	backward-word	move cursor back one word
Esc <	beginning-of-buffer	move to buffer start
^A	beginning-of-line	move to current line start
Esc c	capitalize-word	capitalize first character of current word
Esc k	copy-end-line	copy current line end to the kill stack
Esc w	copy-region	copy marked region to kill-stack
^D	delete-char	delete one character
^K	delete-end-line	delete from cursor to end of line
^R	delete-line	delete one line and push onto kill stack
Esc d	delete-word	remove n words in front of cursor
Esc >	end-of-line	move to buffer end
^E	end-of line	move to current line end
^X^X	exchange-point-mark	exchange positions of cursor and mark
Btn1Mov	extend-adjust	select text in direction of cursor
Btn1Up	extend-end	highlighted text becomes first selection
^F	forward-char	move cursor forward one character
^N	forward-line	move cursor down one line
^V	forward page	move down 1 screen
Esc f	forward-word	move forward one word
Esc |	lower-case-word	change to lower case
^M	new-line	insert line feed character at cursor point
Esc o	open-line	open one empty line above cursor point
^O	open-space	open one empty line after cursor
^C	repeat-command	repeat last command
^H	eraser-char	delete preceding character
Esc H	eraser-word	delete preceding word
^X^N	scroll-down	scroll down one line
^X^B	scroll-left	scroll left one column
^X^F	scroll-right	scroll right one column
^X^P	scroll-up	scroll up one line
Btn3Mov	scroll-window	scroll vertically keeping current cursor position
Btn1Dwn(3)	select line	select line under cursor
Btn1Dwn(2)	select word	select word under cursor
Btn1Dwn	selection-start	set mark and begin text selection
^<Space>	set-mark	set mark at current cursor point
Btn2Dwn	stuff	copy text form x selection to cursor point
^I	tabulation	insert n tab character
^T	twiddle chars	swap two characters preceeding cursor
Esc u	upper-case-word	change to upper case
^W	wipe-region	delete the region
^Y	yank	insert the nth killed iten at the cursor point.

3.5 The Class Source Editor

This section details the Class Source Editor and describes how to edit and compile a class and how to compile a class with renaming of inherited properties. Display the Class Source Editor from the Class Browser as explained in Section 2.3. You see the source editor appear as in Figure 3.138.

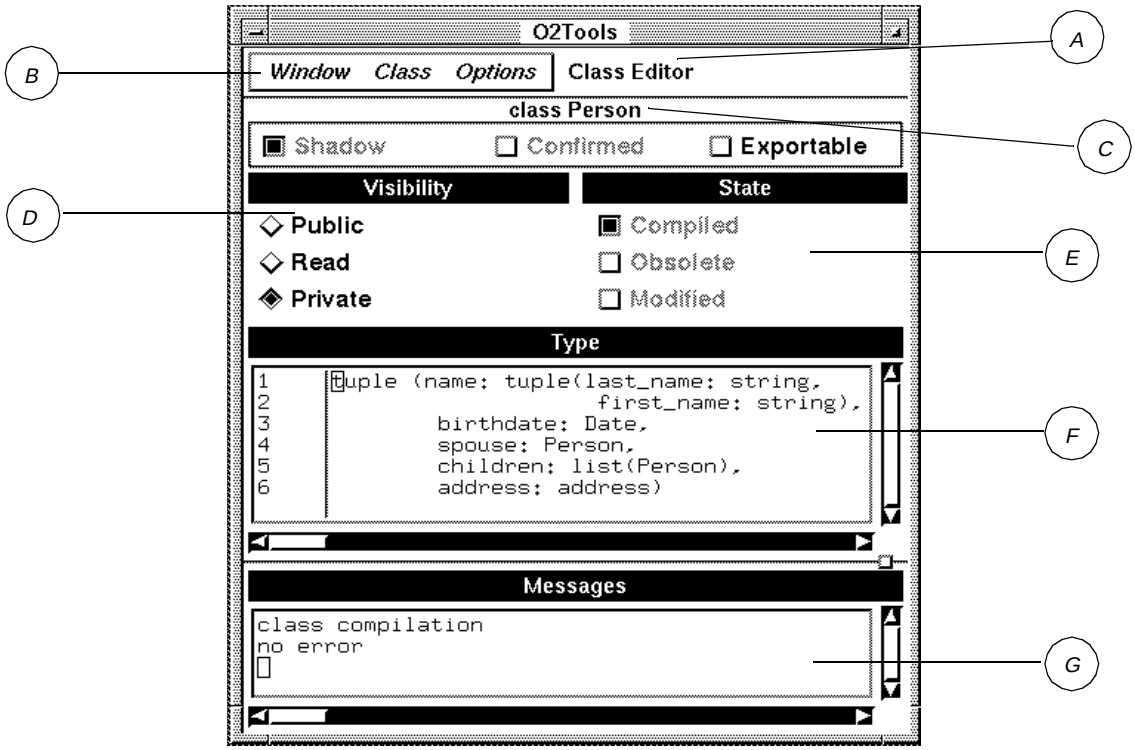


Figure 3.138: Class Source Editor components:

A: Source editor type *B: Menu bar*
C: Sensitive source name *D: Visibility part*
E: state column *F: Type window*
G: Messages window

The visibility part (*D*) is made up of three modifiable toggle buttons: Private, Read and Public.

Use these buttons to set or modify the visibility of the class. You edit or modify the type of the class in the type window (*F*). If the Shadow flag is highlighted, this means that the class is partially defined (i.e. a non-existent class appears in the type definition).

Select the Exportable button so that the class can be exported to other schemas.

You use the Class menu, shown in Figure 3.139, to edit and compile a class, compile a class with renaming, enter any documentation relevant to the class and print to file the class source.

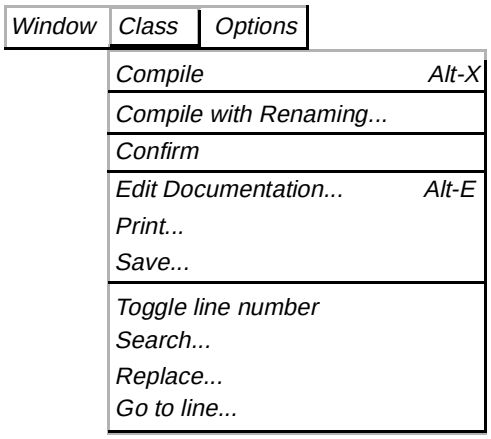


Figure 3.139: Class source editor menu

Editing and compiling a class

To edit and compile a class simply enter the class type in the type window, marked (*F*) in Figure 3.138. Display the Class source menu shown in Figure 3.139 and select Compile. If no error occurs you see the message, marked (*G*) in Figure 3.138.

Note

A class can also be compiled using Create and Compile in the Class Browser menu. See Section 2.3.

Compiling a class with renaming

To compile a class and rename its inherited properties, select Compile with renaming. The following dialog box appears:

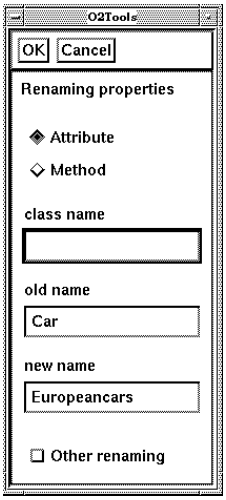


Figure 3.140: Compile with renaming dialog box

Choose whether the inherited property to be renamed is a method or attribute. Fill in the property name, while the class from which the property is inherited is optional.

If you want to rename more than one property click on the other renaming button.

Click on OK to confirm.

Confirm a class

Use the Confirm item to confirm the changes carried out on the class before accessing the base. Trying to access a base before doing confirm will produce an error.

Entering Documentation

Use the Edit Documentation item to enter and record any information relevant to the current class. This option is common to all source editors and is described in detail in Section 3.1.

Printing a class

Use the Print item to "print" the class description to an external file as with the alphanumeric command print. For example:

```
print class Car "path_file"
```

Refer to Section 3.1 for further details.

Saving a class

Use the Save item to save a class source in a given Unix file. Refer to Section 3.1 for further details.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

3.6 The Method Source Editor

This section describes the Method Source Editor and details how to edit and compile the method signature and body, how to test a method and how to manage the versions and dependencies of method bodies. The Method Source Editor is displayed from the Class Browser as explained in Section 2.3.

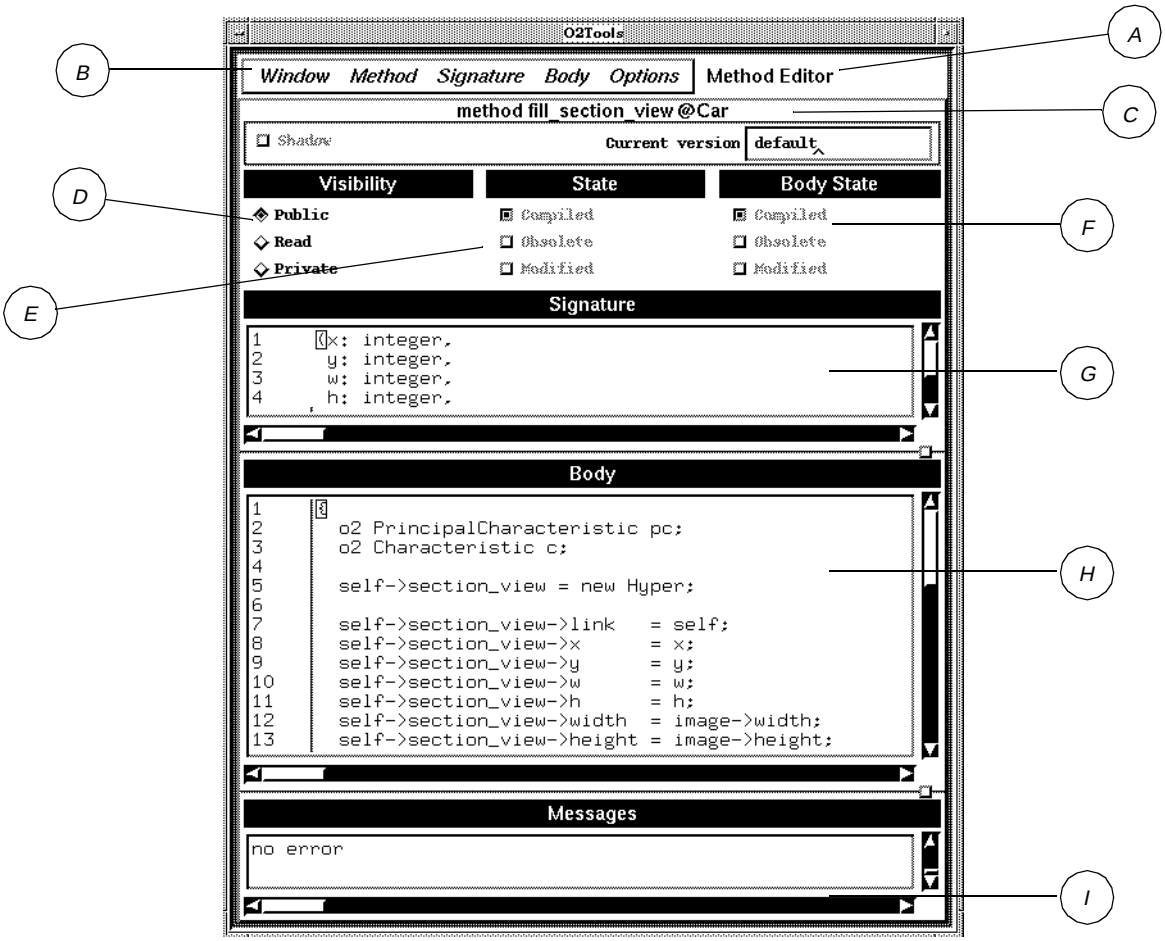


Figure 3.141: Method Source Editor components:

A: Source editor type	*B: Menu bar*	*C: Sensitive source name*
D: Visibility part	*E: Current state*	*F: Body state*
G: Method signature	*H: Method body*	*I: Messages window*

The visibility part (*C*) has three modifiable toggle buttons: Private, Read and Public. Use these buttons to set or modify the visibility of the method. Both the signature state column (*E*) and body state column (*F*) have three flags that cannot be modified: Compiled, Obsolete, Modified as described in Section 3.1. You edit and modify the method signature in the signature window (*G*) and the method body in the body window (*H*).

If the Shadow flag is highlighted, this means that a non-existent class appears in the method signature.

The Current version gives the name of the current version of the method. This section now describes how to use the Method menu, the Signature menu and the Body menu.

The Method menu is made up of the following items shown in Figure 3.142.

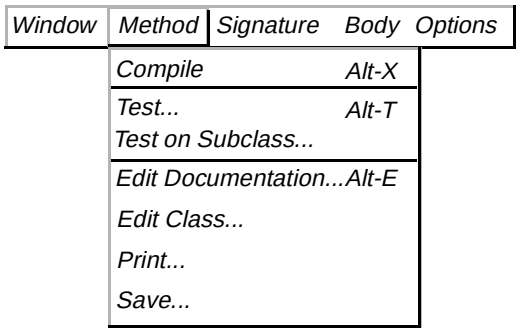


Figure 3.142: Method source method menu

Editing and Compiling a method

To compile a method, you can compile the method signature and the method body separately or at the same time. How to compile a method signature or a method body is described in the following sections.

You can also compile the signature and the body of the method together.

After entering the method signature and body, simply select Compile from the Method pull-down menu, shown in Figure 3.142.

Testing a method

To test the compiled method, select Test from the Method menu shown in Figure 3.142.

An instance of the receiver class is automatically created and displayed on the screen as shown in (A) in Figure 3.143.

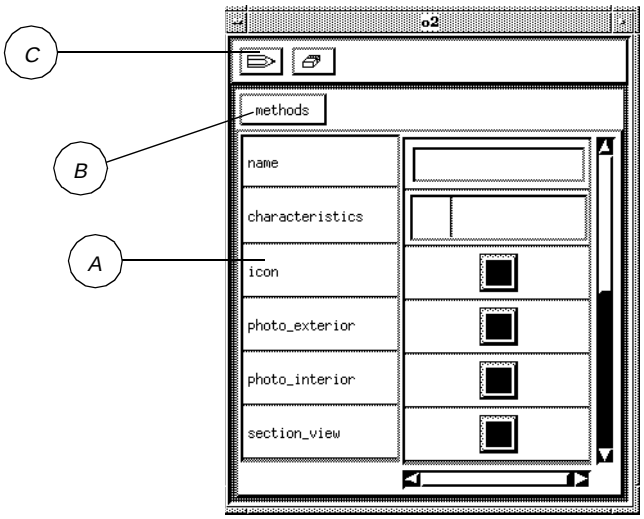


Figure 3.143: Instance of the receiver class

You can edit this class instance and trigger the method to be tested from its menu bar (*B*). You save any modifications by clicking on the pencil button (*C*).

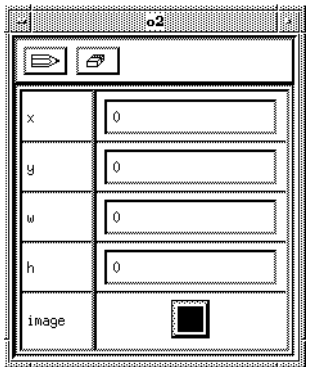


Figure 3.144: Method parameters

If the method requires certain parameters they are asked for interactively in a separate dialog box as shown in Figure 3.144.

You cancel the test at any time by clicking on the eraser button.

Note

During the test, the programming environment is inhibited. It is available again when the test has finished.

Testing a method on a subclass

To test a method on a particular subclass, simply select the item Test on Subclass in the Method menu in Figure 3.142.

The list of available subclasses is displayed in a separate window. Select a subclass and click on OK.

Figure 3.145: Available subclasses

The test is then the same as the simple method test described above with an instance of the receiver class automatically created and displayed as in Figure 3.143. And as with the simple method test you can edit this class instance and save any modifications by clicking on the Pencil button. Trigger the method from the methods menu bar and if any parameters are required you supply them interactively as in Figure 3.144.

Note

During the test, the programming environment is inhibited. It is available again when the test has finished.

Entering Documentation

Use the Edit Documentation item to enter and record any information relevant to the method in question. This option is common to all source editors and is described in detail in Section 3.1.

Edit Class

Click on the Edit Class item to display the Source editor of the method's corresponding class.

Printing a method

Use the Print item to "print" the method signature and body to an external file as with the two alphanumeric print commands, e.g.

```
print method fill_section_view in class Car “path_file”
print method body fill_section_view in class Car “path_file”
```

This item is described in detail in Section 3.1 as are all the other features common to all source editors.

Saving a method

Use the Save item to save a method source in a given Unix file. Refer to Section 3.1 for further details.

The Signature menu has two items as shown in Figure 3.146.

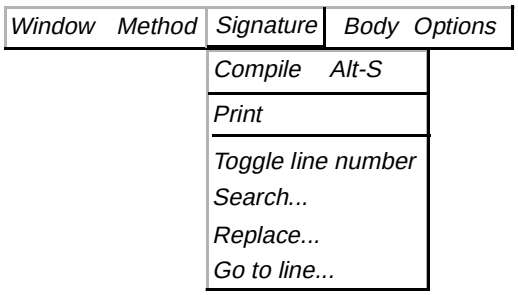


Figure 3.146: Method source signature menu

Editing and Compiling a method signature

Enter the method signature in the signature window as in the example given in Figure 3.147.

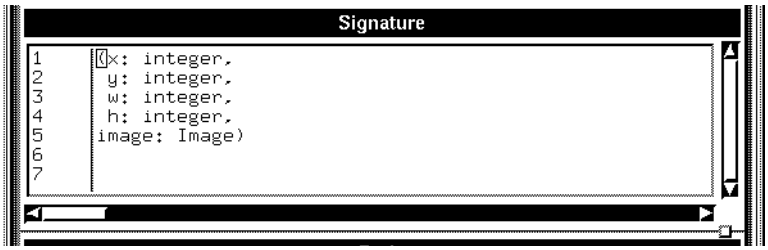


Figure 3.147: Method signature

Select Compile from the Signature pull-down menu shown in Figure 3.146.

Printing a method signature

Use the Print item in the Signature menu to “print” just the method signature to an external file as with the alphanumeric command print.

For example:

```
print method fill_section_view in class Car “path_file”
```

Refer to Section 3.3 for further details.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

The Method Body menu is shown in Figure 3.148 below.

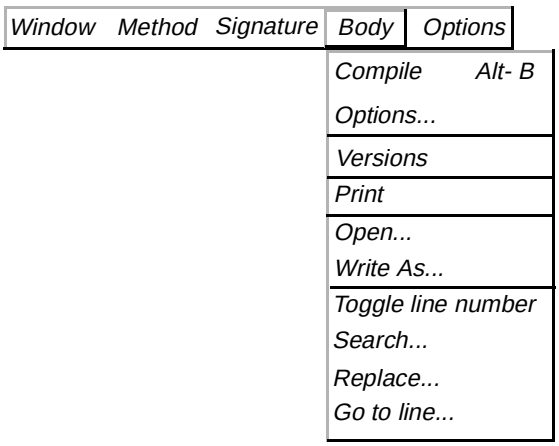


Figure 3.148: Method source body menu

Editing and Compiling a method body

Enter the method body in the body window as in Figure 3.149.

Note

The body window is initialized to {}.

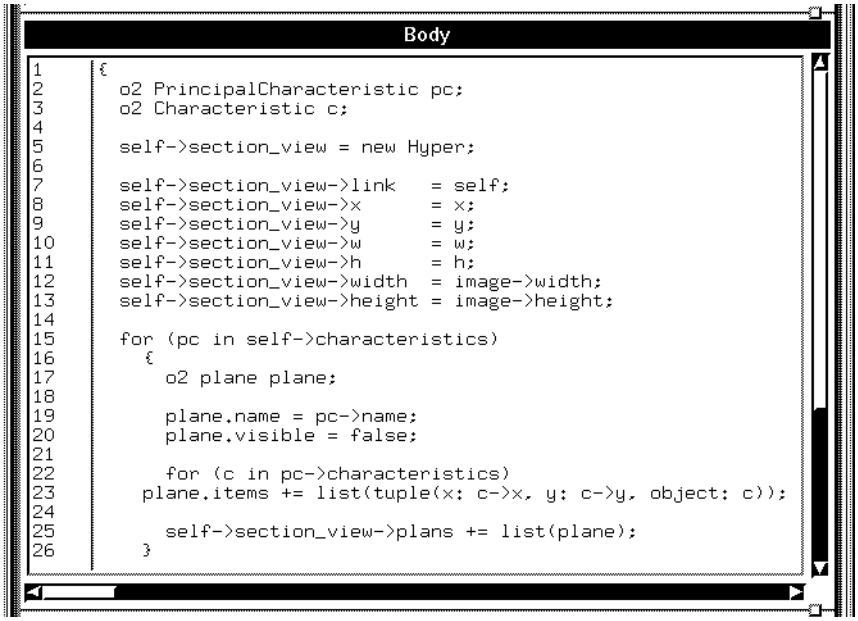


Figure 3.149: Method body

Select Compile from the Body pull-down menu shown in Figure 3.148. If you have body compilation errors, refer to Section 3.2.

Compilation options

You can also define local compilation options for a method body. Select Options from the Method Body pull-down menu.

Versions of the method body

You can create, delete and compile different versions of the method body using the version editor. From the Body menu shown in Figure 3.148 select Versions.

Printing a method body

Use the Print item in the Body menu to “print” just the method body in an external file as with the alphanumeric command print. For example:

```
print method body fill_section_view in class Car “path_file”
```

Opening a method body

Open allows you to modify the Body using the contents of an external file.

Saving a method body

Write As enables you to save the Body contents in an external file. Click on Write As.

Enter the file name and click on OK. The Body contents are now saved in that file.

Important

For full details on how to use these options that are common to method, program and function bodies, refer to Section 3.2.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

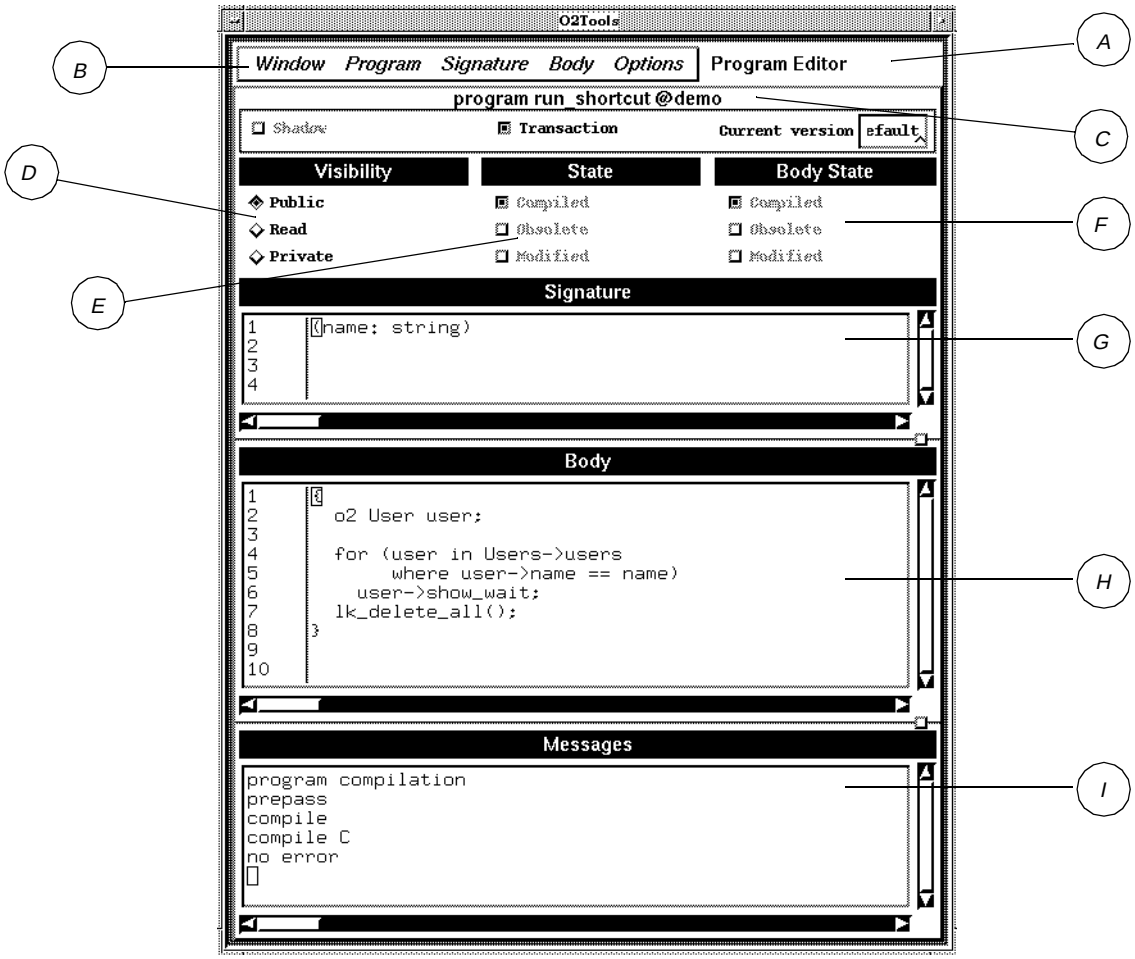
3.7 Program Source Editor

This section describes the Program Source Editor and details how to edit and compile the program signature and body, how to test a program and how to manage versions and dependencies of program bodies.

Display the Program Source Editor from the Application Browser, as explained in Section 2.4.

Figure 3.150: Program Source Editor components:

A: Source editor type
B: Menu bar
C: Sensitive source name
D: Visibility column
E: Current state of source editor
F: Body state
G: Program signature
H: Program body
I: Messages window



The visibility part (*D*) has three modifiable toggle buttons: private, read and public. Use these buttons to set or modify the visibility of the program.

Both the signature state column (*E*) and body state column (*F*) have three flags that cannot be modified: compiled, obsolete, modified.

You edit and modify the program signature in the signature window (*G*) and the program body in the body window (*H*). If the Shadow flag is highlighted, a non-existent class appears in the program signature.

If you select the Transaction box, the program is compiled as a transaction.

The Current version gives the name of the current version of the method.

This section now describes how to use the Program menu, the Signature menu and the Body menu.

The Program menu is made up of the following items shown in Figure 3.151.

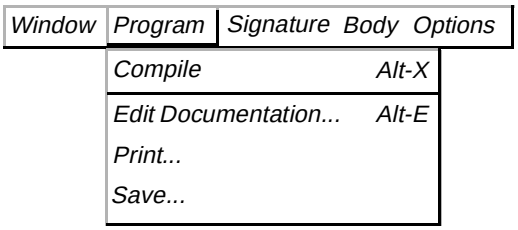


Figure 3.151: Program source program menu

Editing and Compiling a Program

To compile a program, you can compile the program signature and the program body separately or together.

You can also compile the signature and the body of the program together.

After entering the program signature and body, simply select Compile from the program pull-down menu, shown in Figure 3.151.

Note

If you want to test a program, run the application where the program is defined and trigger the program to be tested. To test the application see Section 2.4.

Entering Documentation

Use the Edit Documentation item to enter and record any information relevant to the program you are working on.

This option is a feature found in all source editors and is described in detail in Section 3.1.

Printing a program

Use the Print item to “print” the program body and signature in an external file as you would with the two alphanumeric print commands.

For example:

```
print program run_shortcut in application demo “path_file”
print program body run_shortcut in application demo “path_file”
```

This item is described in detail in Section 3.1 as are all the other features common to all source editors.

Saving a program

Use the Save item to save a program source in a given Unix file. Refer to Section 3.1 for further details.

The Program Signature menu has two items as shown in Figure 3.152.

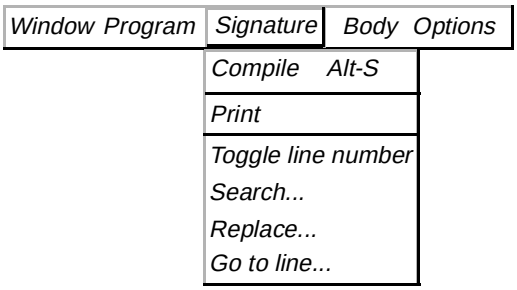


Figure 3.152: Program source signature menu

Editing and Compiling a program signature

Enter the program signature in the signature window as in the example given in Figure 3.153.

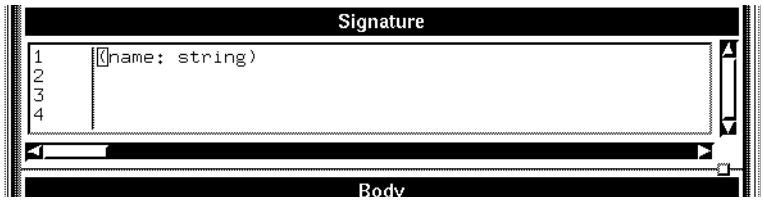


Figure 3.153: Program signature

Select Compile from the Signature pull-down menu shown in Figure 3.152 above.

Printing a program signature

Use the Print item in the Signature menu to “print” just the program signature in an external file as with the alphanumeric command print.

For example,

```
print program run_shortcut in application demo “path_file”
```

Refer to Section 3.1 for further details of this item.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

The Program Body menu is shown below in Figure 3.154

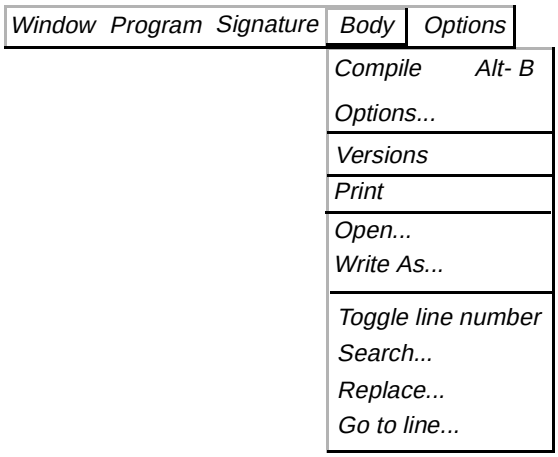


Figure 3.154: Program source body menu

Editing and Compiling a program body

Enter the program body in the body window as in Figure 3.155.

Note

The body window is initialized to {}.

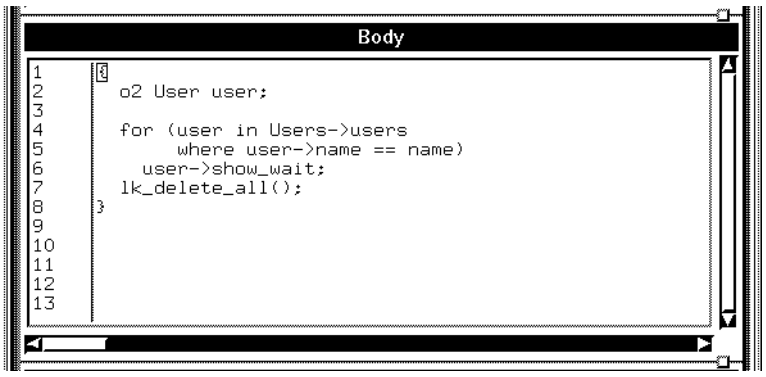


Figure 3.155: Program body

Select Compile from the Body pull-down menu shown in Figure 3.154. If you have body compilation errors, refer to Section 3.1.

Compilation options

You can also define local compilation options for a program body. Select Options from the Program Body pull-down menu.

Versions of program body

You can create, delete and compile different versions of the program body using the version editor. From the Body menu shown in Figure 3.148 select Versions.

Printing a program body

Use the Print item in the Body menu to "print" just the program body in an external file as with the alphanumeric command print. For example:

```
print program body run_shortcut in application demo "path_file"
```

Opening a program body

Open allows you to modify the Body using the contents of an external file.

Saving a program body

Write As enables you to save the Body contents in an external file.

Important

For full details on how to use these options that are common to method, program and function bodies, refer to Section 3.2.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

3.8 Application Variable Source Editor

This section describes the Application Variable Source Editor and details how to compile and edit an application variable.

Display the Application Variable Source Editor from the Application Browser, explained in Section 2.4.

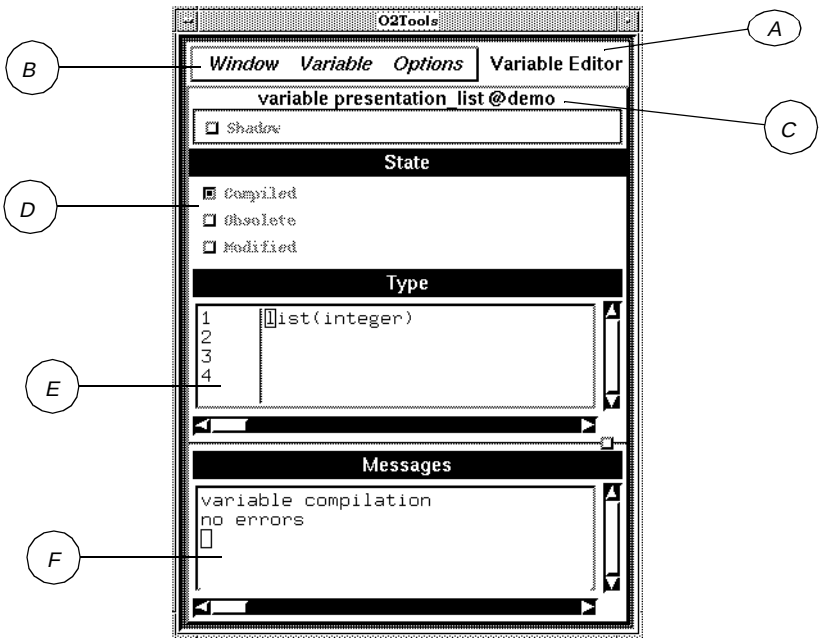


Figure 3.156: Application Variable Source Editor

A: Source Editor type
B: Menu bar
C: Sensitive source title
D: Current state of source
E: Type window
F: Messages window

The state column (*D*) has three flags that cannot be modified: compiled, obsolete, modified as explained in Section 3.1.

You edit and modify the type of the application variable in the Type window (*E*).

If the Shadow flag is highlighted, this means the application variable is partially defined.

The Variable menu is shown is Figure 3.157.

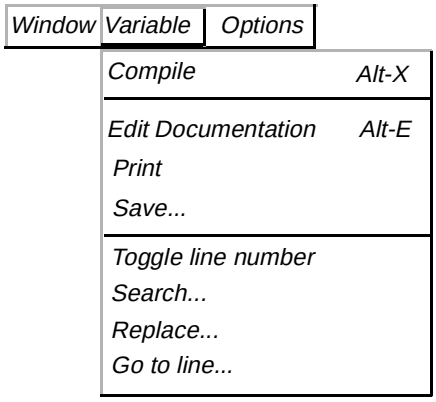


Figure 3.157: Variable menu

Editing and Compiling an Application Variable

To edit and compile an application variable, simply enter the type in the Type window as shown in (*E*) in Figure 3.156.

Pull down the Variable pull down menu and select the item Compile as in Figure 3.157.

Entering Documentation

Use the Edit Documentation item to enter and record any information relevant to the application variable in question. This option is common to all source editors and is described in detail in Section 3.1.

Printing an Application Variable

Use the Print item in the Variable menu to “print” just the application variable in an external file as with the alphanumeric command print. For example:

```
print variable presentation_list in application demo “path_file”
```

Refer to Section 3.1 for further details.‘

Saving an Application Variable

Use the Save item to save an application variable source in a given Unix file. Refer to Section 3.1 for further details.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

3.9 The Function Source Editor

This section describes the Function Source Editor and details how to edit and compile the function signature and body, how to test a function and how to manage the versions and dependencies of function bodies.

The Function Source Editor is displayed from the Function Browser as explained in Section 2.5.

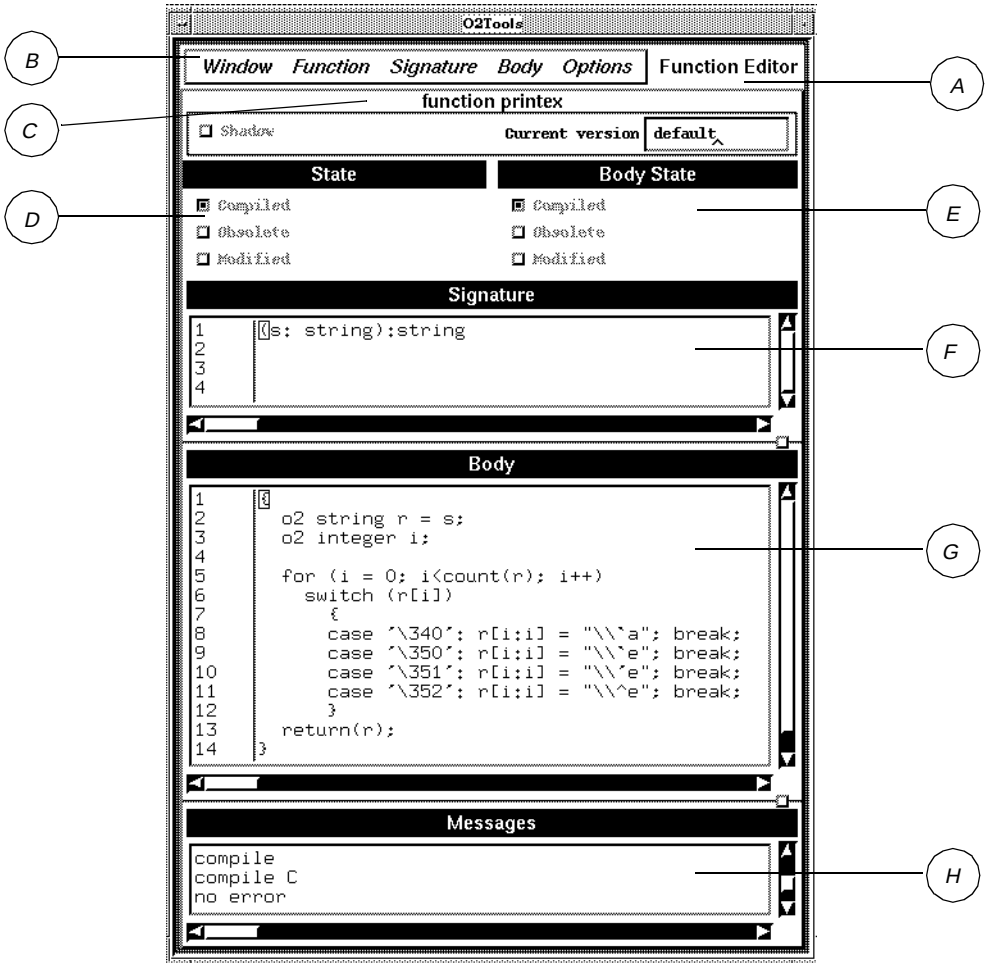


Figure 3.158: Function Source Editor components:

A: Source editor type	*B: Menu bar*
C: Sensitive source title	*D: Current state of source editor*
E: Body state	*F: Function signature*
G: Function body	*H: Messages window*

Both the signature state column (*D*) and body state column (*E*) have three flags that cannot be modified: compiled, obsolete, modified as described in Section 3.1.

You edit and modify the function signature in the signature window (*F*) and the function body in the body window (*G*).

If the Shadow flag is highlighted, this means that an non-existent class appears in the function signature.

The Current version gives the name of the current version of the function.

This section now describes how to use the Function menu, the Signature menu and the Body menu.

The Function menu is made up the following items shown in Figure 3.159.

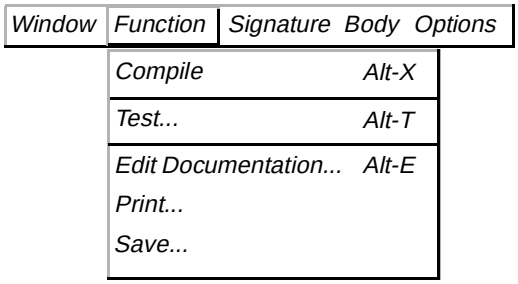


Figure 3.159: Function source function menu

Editing and Compiling a Function

To compile a function, you can compile the function signature and the function body. You can either compile them separately or together.

After entering the function signature and body, simply select Compile from the function pull-down menu, shown in Figure 3.159.

Testing a function

To test the compiled function, select Test from the Function menu as shown in Figure 3.159.

If the function requires certain parameters, a dialog box appears as shown in (*A*) in Figure 3.160 in which you interactively fill in the value. Click on the pencil button in order to save the modifications. If there is a result it is then displayed.

Click on the Eraser button to cancel the test.

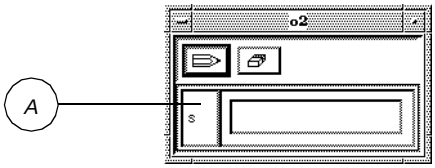


Figure 3.160: Function parameters

Entering Documentation

Use the Edit Documentation item to enter and record any information relevant to the function in question.

This option is common to all source editors and is described in detail in Section 3.1.

Printing a Function

Use the Print item in the Function menu to "print" the function signature and body in an external file as with the two alphanumeric print commands.

For example:

```
print function printex "path_file"
print function body printex "path_file"
```

Refer to Section 3.1 for further details.

Saving a Function

Use the Save item to save a function source in a given Unix file. Refer to Section 3.1 for further details.

The Function Signature menu is shown in Figure 3.161.

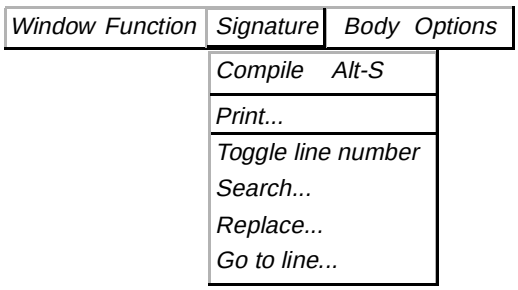


Figure 3.161: Function source signature menu

Editing and Compiling a function signature

Enter the function signature in the signature window as in the example given in Figure 3.162.

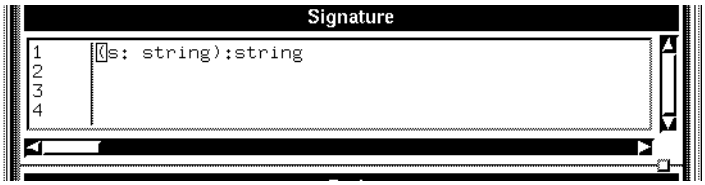


Figure 3.162: Function signature

Select Compile from the Signature pull-down menu shown in Figure 3.161.

Printing a Function signature

Use the Print item in the Function menu to “print” just the function signature in an external file as with the alphanumeric command print.

For example:

```
print function printex “path_file”
```

Refer to Section 3.1 for further details.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

The Function body menu is shown in Figure 3.163.

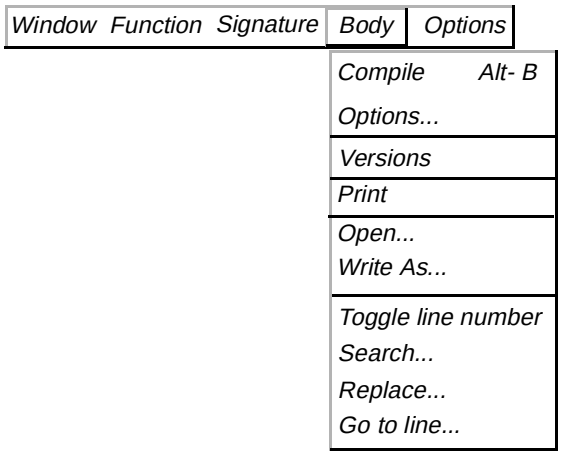


Figure 3.163: Function source body menu

Editing and Compiling a function body

Enter the function body in the body window as in Figure 3.164.

Note

The body window is initialized to {}.

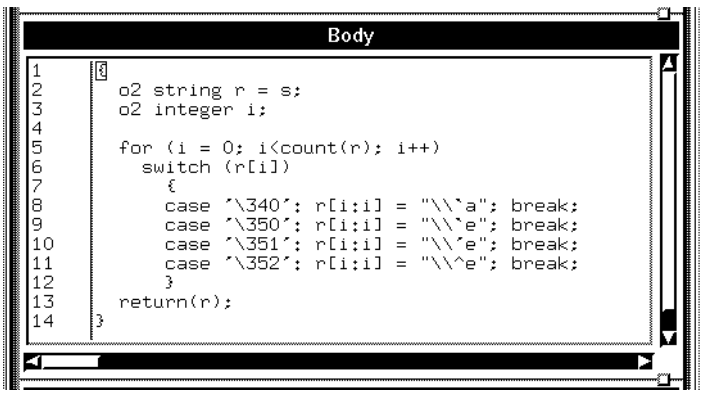


Figure 3.164: Function Body

Select Compile from the Body pull-down menu shown in Figure 3.163. If you have body compilation errors, refer to Section 3.2.

Compilation options

You can also define local compilation options for a function body. Select Options from the Function Body pull-down menu.

Refer to Section 3.2 on body compilation options.

Versions of function body

You can create, delete and compile different versions of the function body using the version editor. From the Body menu select Versions.

Printing a function body

Use the Print item in the Body menu to "print" just the function body in an external file as with the alphanumeric command print. For example:

```
print function body printex "path_file"
```

Opening a function body

Open allows you to modify the Body using the contents of an external file.

Saving a function body

Write As enables you to save the Body contents in an external file.

Important

For full details on how to use these options that are common to method, program and function bodies, refer to Section 3.2.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

3.10 The Persistent Type Source Editor

This section describes the Persistent Type Source Editor and details how to compile and edit a persistent type.

Display the Persistent Type Source Editor from the Persistent Type Browser, explained in Section 2.6.

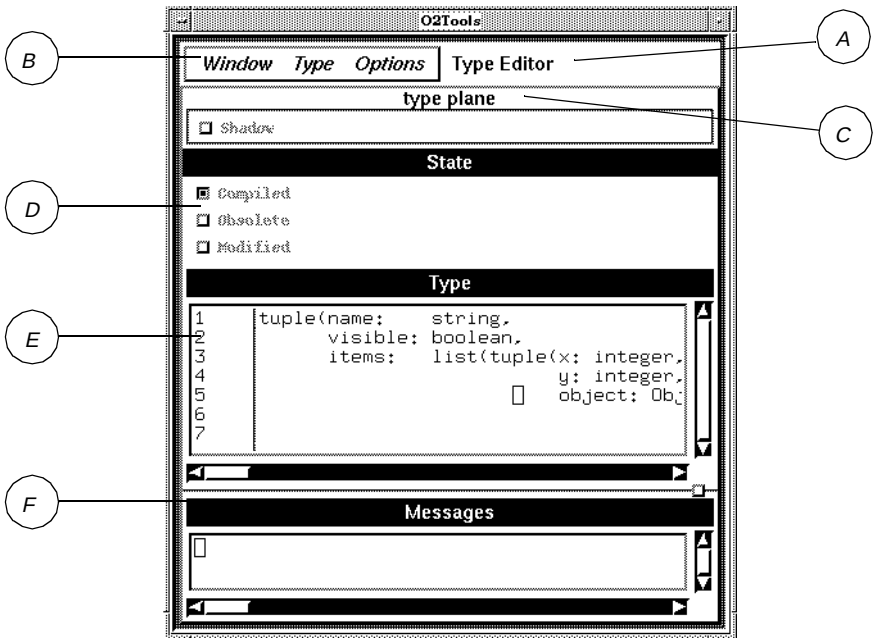


Figure 3.165: Persistent Type Source Editor

A: Source Editor type

B: Menu bar

C: Sensitive title

D: Current state of source editor

E: Type window

F: Messages window

The State column (*C*) has three flags that cannot be modified: compiled, obsolete, modified as explained in Section 3.1.

You edit and modify the type of the Persistent Type in the Type window (*D*).

If the Shadow flag is highlighted, this means the persistent type is partially defined.

The Type menu is shown in Figure 3.166.

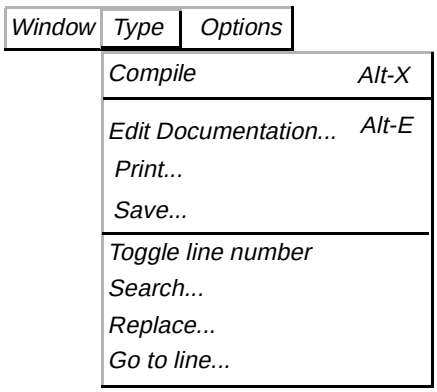


Figure 3.166: Persistent type source menu

Editing and Compiling a Persistent Type

To edit and compile a persistent type, simply enter the type in the Type window as shown in (*E*) in Figure 3.165. Pull down the Type pull-down menu and select the item Compile as in Figure 3.166.

Entering Documentation

Use the Edit Documentation item to enter and record any information relevant to the persistent type in question. This option is common to all source editors and is described in detail in Section 3.1.

Printing a Persistent type

Use the Print item in the Type menu to “print” the persistent type in an external file as with the alphanumeric command print. For example:

```
print type plane “path_file”
```

Refer to Section 3.1 for further details.

Saving a Persistent type

Use the Save item to save a persistent type source in a given Unix file. Refer to Section 3.1 for further details.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

3.11 The Persistent Name Source Editor

This section describes the Persistent Name Source Editor and details how to compile and edit a persistent name.

Display the Persistent Name Source Editor from the Persistent Name Browser, explained in Section 2.7.

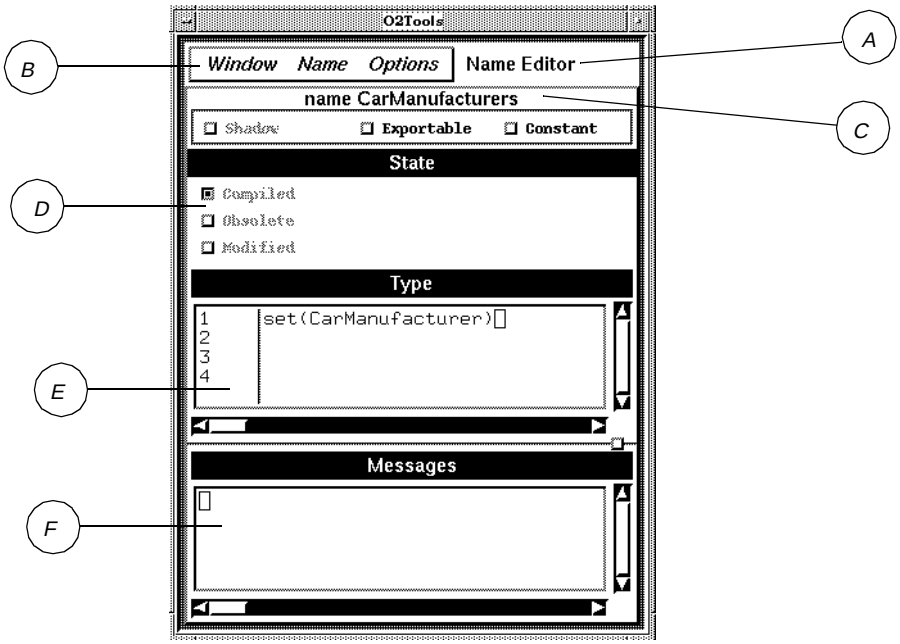


Figure 3.167: Persistent Name Source Editor

A: Source Editor type *B: Menu bar*

C: Sensitive source title *D: Current state of source*

E: Type window *F: Messages window*

The State column (*D*) has three flags that cannot be modified: compiled, obsolete, modified as explained in Section 3.1.

You edit and modify the type of the Persistent Name in the Type window (*E*). If the Shadow flag is highlighted, this means the persistent name is partially defined.

You select the Exportable button so that the name can be exported to other schemas.

Select the Constant button to specify whether the name can be modified or not. Refer to the *O_2C Reference Manual* for details on using Constant names.

The Name menu is shown in Figure 3.168.

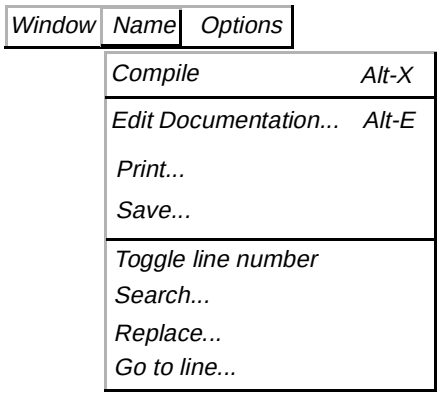


Figure 3.168: Persistent name source menu

Editing and Compiling a Persistent Name

To edit and compile a persistent name, simply enter the type in the Type window as shown in (*E*) in Figure 3.167. Click the Name pull-down menu and select the item Compile as in Figure 3.168.

Entering Documentation

Use the Edit Documentation item to enter and record any information relevant to the persistent name in question. This option is common to all source editors and is described in detail in Section 3.1.

Printing a Persistent name

Use the Print item in the Name menu to "print" the persistent name in an external file as with the alphanumeric command print. For example:

```
print name Carmanufacturers "path_file"
```

Refer to Section 3.1 for further details.

Saving a Persistent name

Use the Save item to save a persistent name source in a given Unix file. Refer to Section 3.1 for further details.

Text manipulation commands

All the text manipulation commands are described in Section 3.4.

3.12 The O_2Shell Editor

You use O_2 alphanumerically or carry out O_2 queries with O_2Shell.

To open the O_2Shell Editor, click on the O_2Shell button (*A*) on the O_2Tools dashboard.

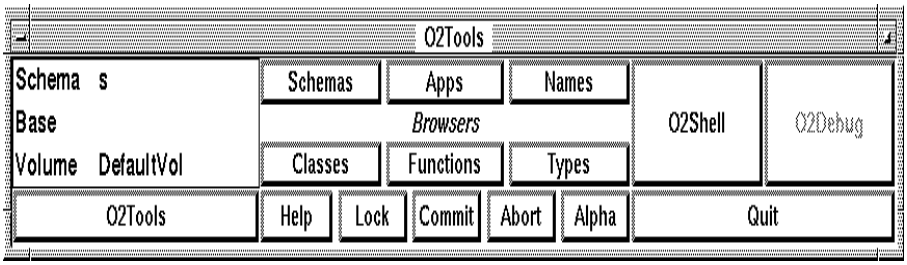


Figure 3.169: O_2Shell button

The O_2Shell Editor now appears as shown in Figure 3.170.

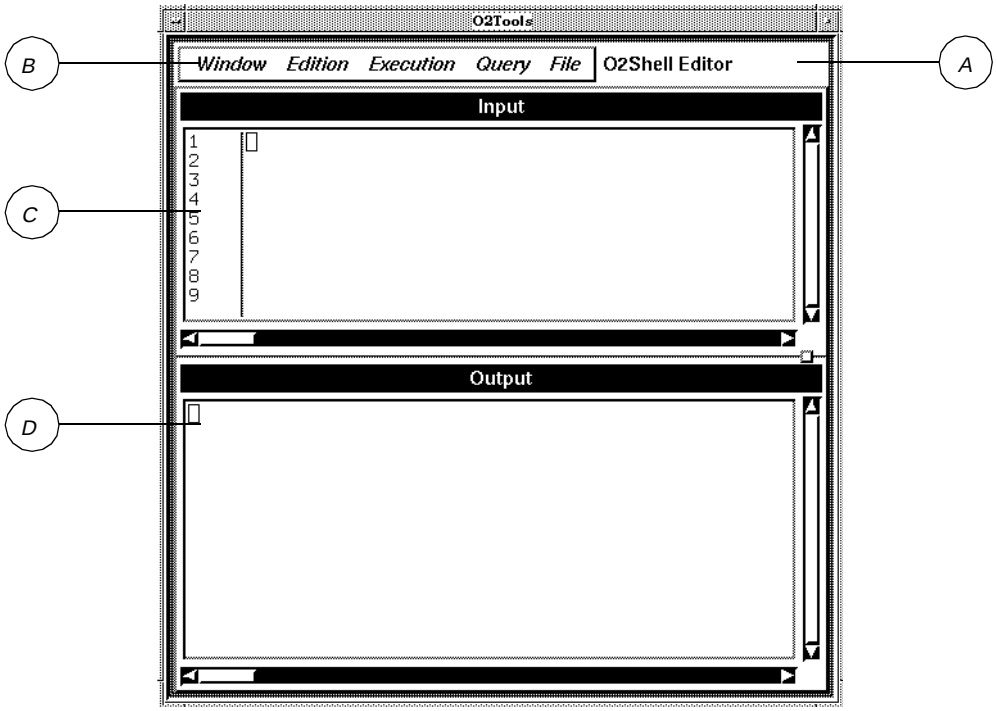


Figure 3.170: O_2Shell Editor components

A: Editor title
B: Menu bar
C: Input window
D: Output window

Important

Two stars - ** -before the O_2Shell editor title, as in Figure 3.171, means that you have interactively modified the editor but you have not yet saved your work.

The stars disappear when you select the Store item in the Window menu.

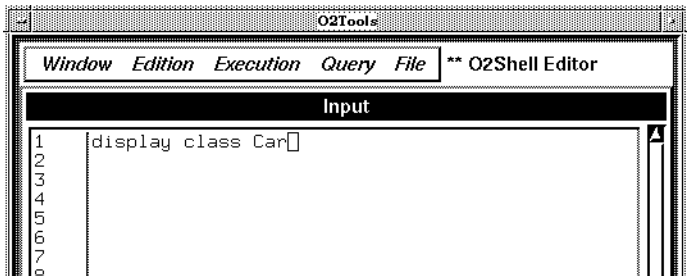


Figure 3.171: Modified O_2Shell editor

Use the input subwindow (*C*) to enter an alphanumeric O_2 command or an O_2 query in conjunction with the File menu and the Window menu.

The output window is not editable and visualizes the result of command entered in the input subwindow.

This section now describes how to use the Edition menu, File menu, Window menu, Execution menu and Query menu.

The Edition menu

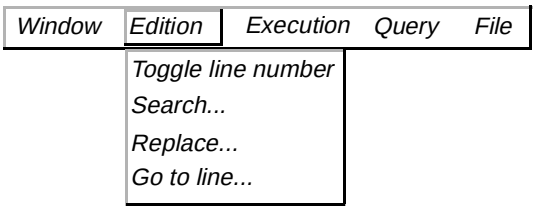


Figure 3.172: The Edition menu

Refer to Section 3.4 for full details on how to use these menu options.

The File menu

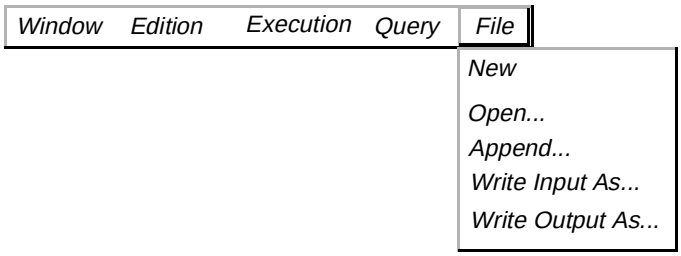


Figure 3.173: Shell file menu

- ***New***

  The New item of the File menu clears the Input window.

- ***Open***

  When you select the Open item, you load an ASCII file into the input window. Simply enter the name of the file in the Dialog box that appears, shown in Figure 3.174 and click on OK. The file contents are now displayed in the input window.

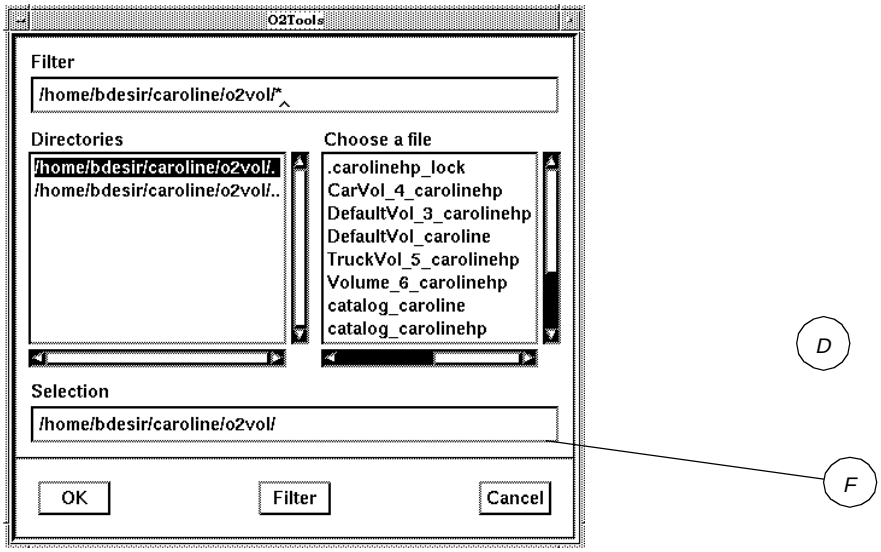


Figure 3.174: Dialog box

- ***Append***

  With the Append item, you add on the contents of an ASCII file to what is already in the input window. Enter the file name in the Dialog box shown in Figure 3.174 and click on OK. The file contents are immediately added onto the contents of the input window.

- ***Save Input and Output***

  Use the Save Input As item to save the contents of the input window and the Save Output As item to save the contents of the output window in an external file. Enter the name of the file which you want to save the input or output in the Dialog box shown in Figure 3.174 and click on OK.

The Window menu

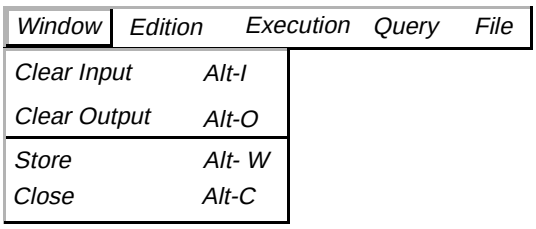


Figure 3.175: Shell Window menu

- ***Clear Input and Output***

  The input and output subwindows can be cleared by using the Window pull-down menu and by selecting Clear Input and Clear Output respectively.

- ***Store and Close***

  The Store and Close items are explained in Section 3.1.

  This section now details how to run an alphanumeric command and an O_2 Query.

Running an alphanumeric command

You run alphanumerical commands using the Execution menu shown in Figure 3.176 below.

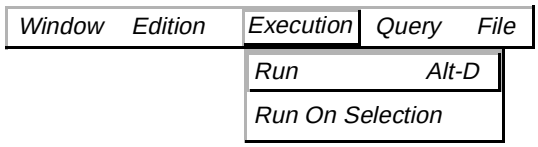


Figure 3.176: Shell Execution menu

- ***Run***

  To run an O_2 command, type into the input subwindow an O_2 command and then select Run from the Execution pull-down menu. All the contents of the input subwindow are run and the result is displayed in the output subwindow.

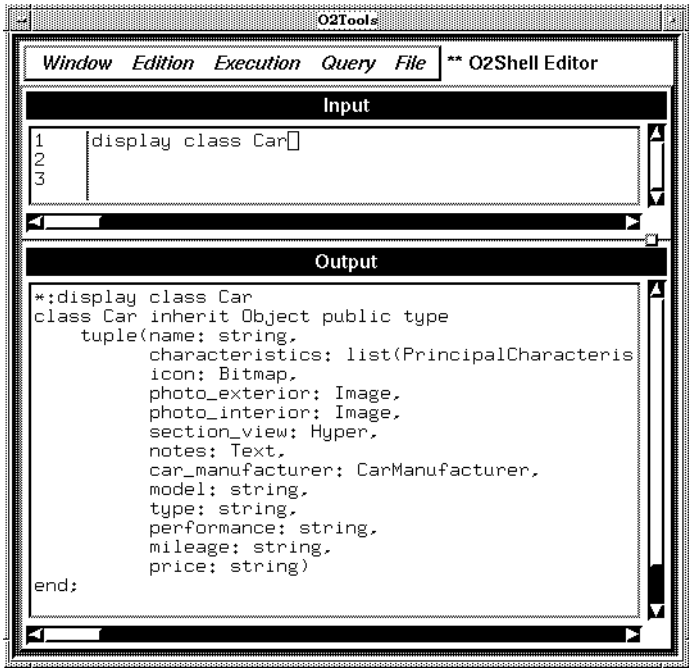


Figure 3.177: Entering an O_2 command

Note

Complete coherence is maintained with the graphical environment. An alphanumeric command has the same effect as the corresponding graphic command. For example, deleting a class using the O_2 Shell updates and refreshes the Class Browser if it is displayed.

- ***Run On Selection***

  To run only part of the input window command, select the part of the text you want to be run and then select Run On Selection from the Execution menu.

Note

When a commit command is reached when you are running O_2 commands, a dialog box appears, shown in Figure 3.178, asking you to choose the behavior of the commit: validate, commit or ignore.

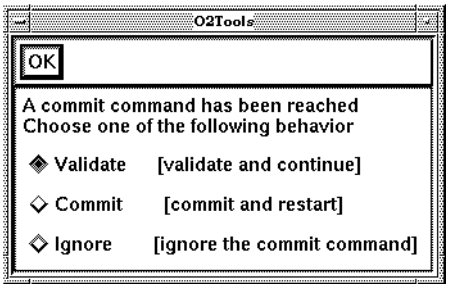


Figure 3.178: O_2 Shell commit

Running an O_2 query

To query use the Query menu shown in Figure 3.179.

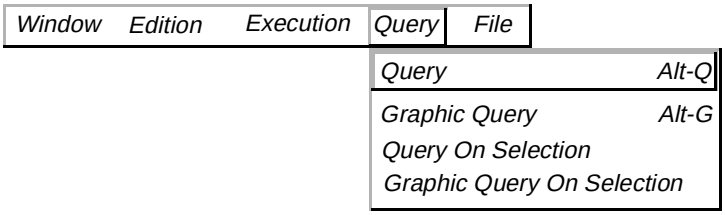


Figure 3.179: Shell Query menu

- ***Query***

  To run an O_2 query, edit the input subwindow by entering a query.

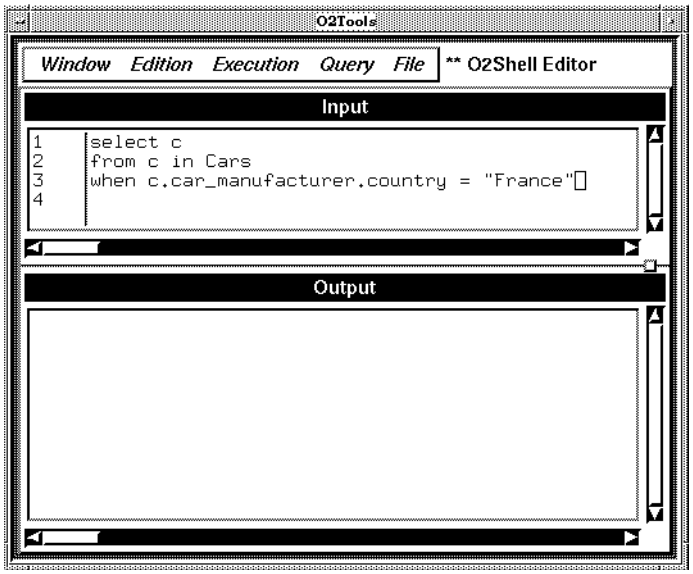


Figure 3.180: Running a Query

Now select Query from the Query pull-down menu. The result is displayed in the output subwindow.

- ***Graphic Query***

  To display the result of a query graphically select Graphic Query.

- ***Query On Selection***

  To only run part of the input window query, select the part of the text you want to be run and then display the Query pull-down menu and select Query On Selection.

- ***Graphic Query On Selection***

  To display the result of query graphically select Graphic Query On Selection.

3.13 Global options

To change the global options of your programming environment, you must firstly click on the O_2Tools button in the dashboard, shown in (*A*) in Figure 3.181 below.

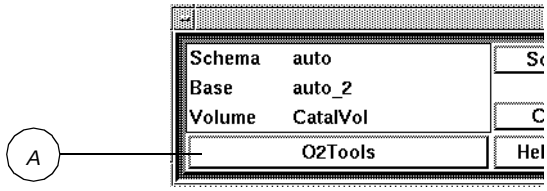


Figure 3.181: O_2Tools button

The following window then appears.

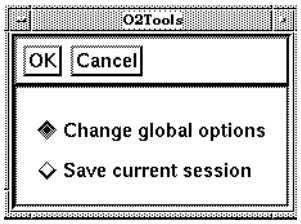


Figure 3.182: O_2Tools dialog box

This box allows you to either change the global options or save the current session. (To save current session refer to Section 4.1).

To change the global options, select Change global options as in Figure 3.182 and click on OK.

You now see the Configuration window shown in Figure 3.183.

This dialog box enables you to specify the following:

- ***Default path***
  This path is used by default by the O_2 file selector. Enter the path name in the File Selector section of the window (*A*) and then on OK.

- ***Compilation options***

  These options, shown in Figure 3.183 are used in the same way as the compilation options described in Sections 3.6, 3.7 and 3.9 for compiling Bodies.

  To enter include paths, click on the I (include path) button, (*B*) in Figure 3.183, and enter one or more paths in the window. Each path must be prefixed by -I as shown in Figure 3.183 (*C*).

  To enter #define directives, click on the D (defines) button, and enter all the names, each preceded by -D in the window (*E*).

  To enter #undefine directives, click on the U (undefines) button (*F*) and enter the names, each preceded by -U in the window (*G*).

  If you want to add any of the set options: g (debug), t (trace), s (stack) or O (optimize), click on the relevant button(s) and OK.

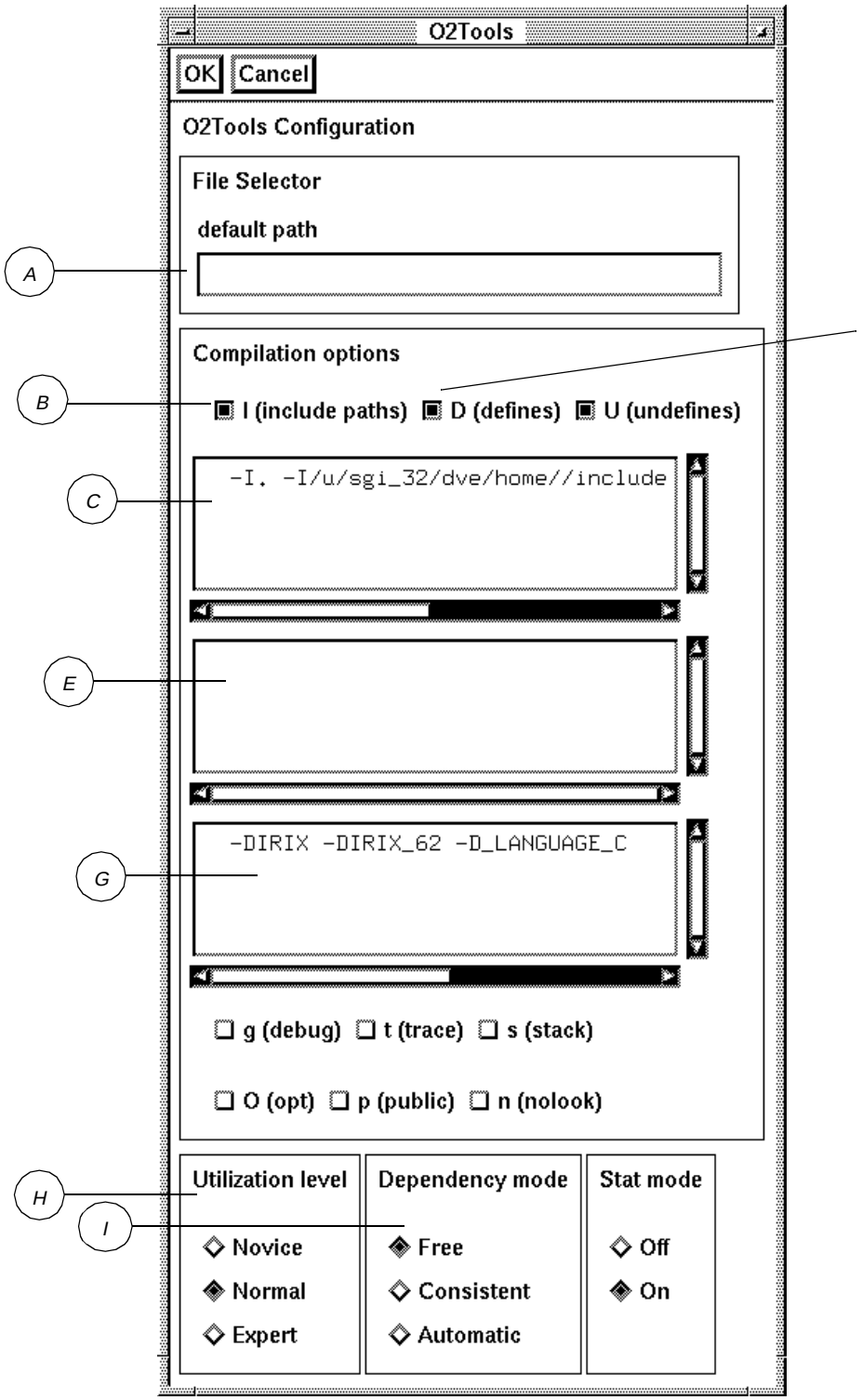


Figure 3.183: O_2Tools configuration

- ***Utilization levels***

  The utilization level (*H*) determines how much confirmation is asked for after various commands. Click on the level you want and on OK. Novice indicates confirmation is required each time you carry out dangerous commands such as a delete. Normal means confirmation is only asked for when you use the abort, alpha, quit commands. At the Expert level none of these confirmations are required. The window is initialized to Normal.

- ***Dependency modes***

  To set the dependency mode (*I*) to Free, Constant or Automatic, click on the relevant button and OK. The window is initialized to Free.

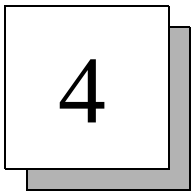
- ***Stat mode***

  The stat mode determines if the information display on schemas, bases and volumes is made using the stat option thereby enabling you to obtain statistical information.

Note

For details on how to use these various compilation, set dependency and stat options, refer to the *O_2C Reference Manuals*.

3 Programming Environment



Customizing O_2 Tools

The O_2Tools user interface is built using O_2Look and can therefore be customized in the same way as O_2Look by modifying graphical resources. For more information about O_2Look resources refer to the *O_2Look User Manual*.

This chapter firstly explains how to preconfigure O_2Tools by creating a configuration file. It then gives the graphic resources you use to customize the following components of O_2Tools:

- Customizing the Dashboard
- Customizing presentations and object masks
- Customizing the schema browser
- Customizing the application browser
- Customizing the class browser
- Customizing the function browser
- Customizing the persistent type browser
- Customizing the persistent name browser
- Customizing the class source editor
- Customizing the method source editor
- Customizing the program source editor
- Customizing the variable source editor
- Customizing the function source editor
- Customizing persistent type source editor
- Customizing persistent name source editor
- Customizing the Version editor
- Customizing the O2Shell editor
- Customizing the dialog boxes

4 Customizing O2 Tools

4.1 Preconfiguring O_2 Tools

If you know you are going to work on the same schema, base, and browsers, for several O_2Tools sessions, you can create a configuration file so that the schema, base and browsers you want to work with automatically appear, open and selected, each time O_2Tools is launched.

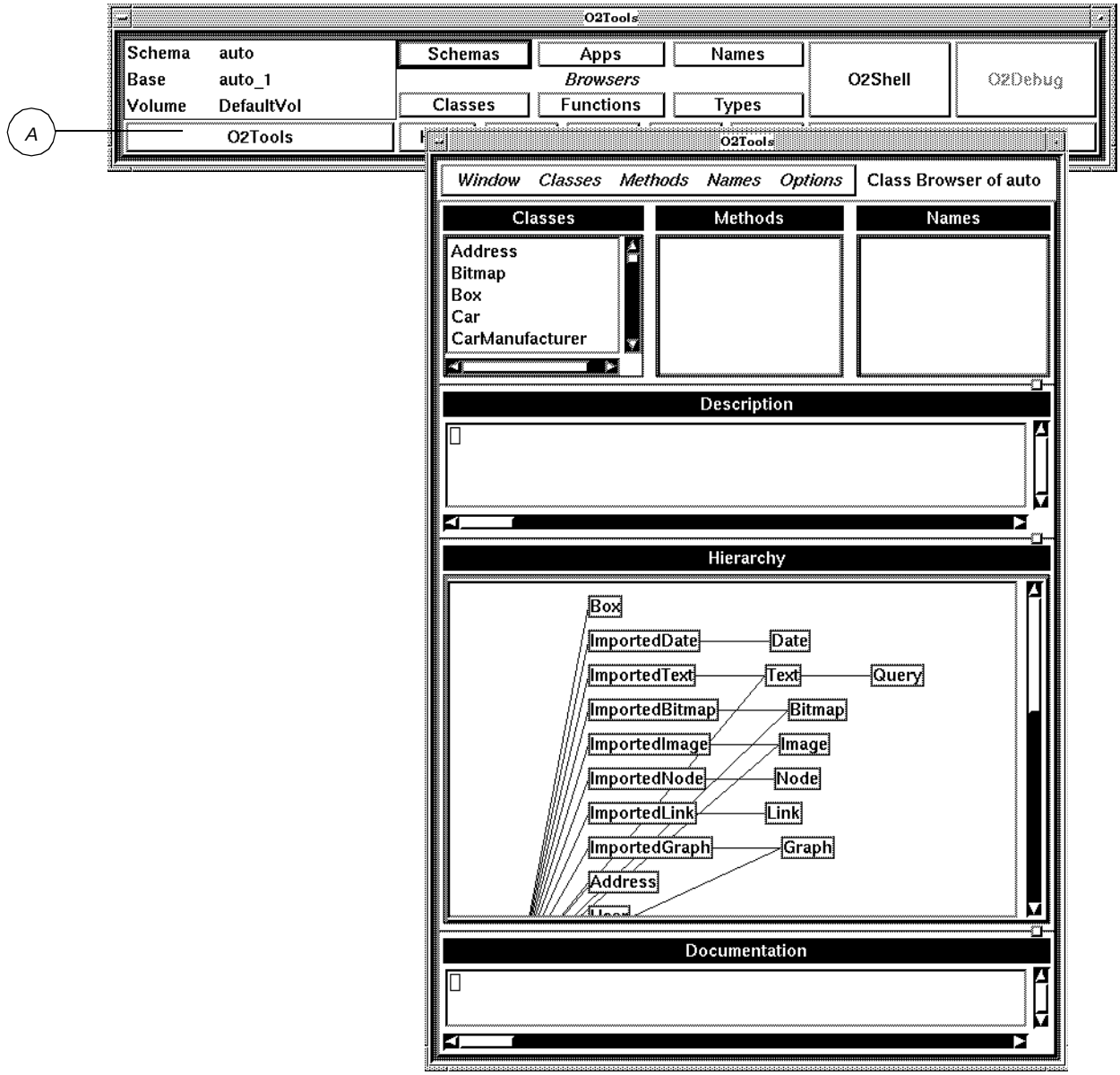


Figure 4.184: O_2Tools

To create a configuration file, simple click on the O_2Tools button on the dashboard marked (*A*) in Figure 4.184.

A dialog box, shown in Figure 4.185 now appears. Select “Save current session” and click on OK. (An introduction to this box is given in Section 1.4)

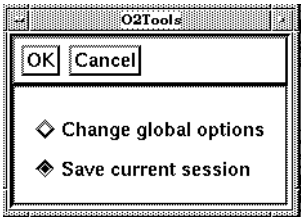


Figure 4.185: O_2Tools dialog box

Another dialog box then asks you to confirm your command

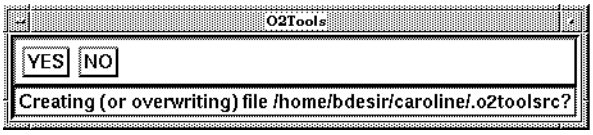


Figure 4.186: Confirm file creation

A configuration file called **.o2toolsrc** is created in your home directory which contains information corresponding to the current session and which you can consult.

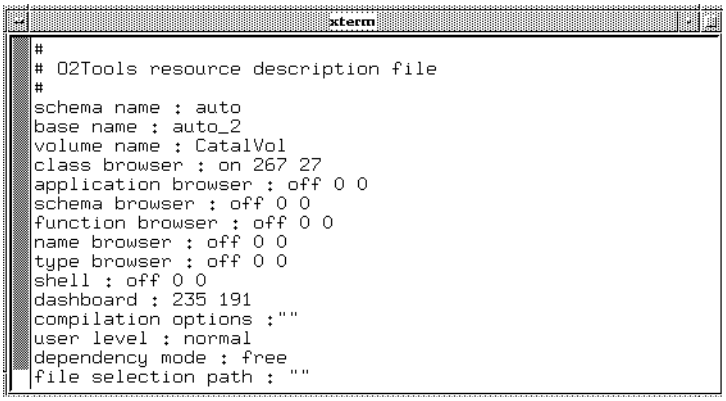


Figure 4.187: .o2toolsrc file

Note

The **.o2toolsrc** file can be edited and updated separately.

4.2 Customizing the Dashboard

This section gives the graphic resources need to customize the dashboard described in Section 2.1

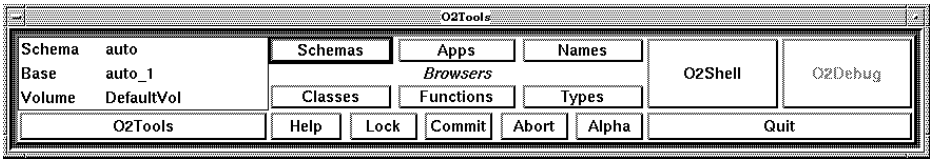


Figure 4.188: The dashboard

It firstly gives the graphic resources for the dashboard presentation and then the dashboard object mask. The resources for the dashboard itself.

Table 4.1

Graphic resources for the Dashboard presentation

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic

Table 4.1 lists the graphic resources you use to customize the dashboard presentation with each name specifying:

- ***backgroundColor*** Presentation background color
- ***foregroundColor*** Presentation foreground color

The dashboard presentation name is **ptoolsdashboard**, e.g.

```
O2Tools.ptoolsdashboard.foregroundColor:Midnight Blue
```

Table 4.2

Graphic resources for the dashboard object mask

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic

Table 4.2 lists the graphic resources used to customize the dashboard object mask with each name specifying:

- ***backgroundColor*** Object mask background color
- ***foregroundColor*** Object mask foreground color

The dashboard object mask name is **otoolsdashboard**, e.g.

```
O2Tools.ptoolsdashboard.otoolsdashboard.foregroundColor:Midnight Blue
```

Table 4.3

Dashboard graphical resources

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserButtonFontList	DashboardButtonsFontList	Font	Fixed
browserLabelFontList	BrowserLabelFontList	Font	Fixed
shellButtonFontList	DashboardButtonsFontList	Font	Fixed
debugButtonFontList	DashboardButtonsFontList	Font	Fixed
infoFontList	InfoFontList	Font	Fixed
O2ToolsButtonFontList	DashboardButtonsFontList	Font	Fixed
exitButtonFontList	DashboardButtonsFontList	Font	Fixed
helpButtonFontList	DashboardButtonsFontList	Font	Fixed
lockButtonFontList	DashboardButtonsFontList	Font	Fixed
commitButtonFontList	DashboardButtonsFontList	Font	Fixed
abortButtonFontList	DashboardButtonsFontList	Font	Fixed
alphaButtonFontList	DashboardButtonsFontList	Font	Fixed

Table 4.3 lists the graphic resources with which you can customize the dashboard.

Each name specifies the following:

- ***backgroundColor*** Dashboard background color
- ***foregroundColor*** Dashboard foreground color
- ***browserButtonFontList*** Browser button font
- ***browserLabelFontList*** Browser panel label font
- ***shellButtonFontList*** Shell button font
- ***debugButtonFontList*** Debug button font
- ***infoFontList*** Information part font
- ***O2ToolsButtonFontList*** O_2Tools button font
- ***exitButtonFontList*** Exit button font
- ***helpButtonFontList*** Help button font
- ***lockButtonFontList*** Lock button font
- ***commitButtonFontList*** Commit button font
- ***abortButtonFontList*** Abort button font
- ***alphaButtonFontList*** Alpha button font

The name of the dashboard is **`toolsdashboard`**.

For example:

```
O2Tools.ptoolsdashboard.backgroundColor: Yellow
O2Tools.ptoolsdashboard*toolsdashboard.shellFontList:9x15
```

4.3 Customizing presentations and object masks

You customize the presentations and object masks for all the browsers, source editors and O_2Shell in the same way.

Table 4.4

Presentation resources

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic

Table 4.4 lists the graphic resources you use to customize a browser, source editor or O_2Shell presentation.

Each name specifies the following:

- ***backgroundColor*** Presentation background color
- ***foregroundColor*** Presentation foreground color

For example, the schema browser presentation name is **pschemabrowser** and is you use it in the following way:

```
O2Tools.pschemabrowser.foregroundColor:Midnight Blue
```

The application browser presentation name is **papplibrowser**, and is used as follows:

```
O2Tools.papplibrowser.foregroundColor:Midnight Blue
```

The browsers, source and O_2Shell editors have the following presentation names:

- ***schema browser presentation*** **pschemabrowser**
- ***application browser presentation*** **papplibrowser**
- ***class browser presentation*** **pclassbrowser.**
- ***function browser presentation*** **pfunctionbrowser.**
- ***persistent type browser presentation*** **ptypebrowser.**
- ***persistent name browser presentation*** **pnamebrowser.**
- ***class source editor presentation*** **pclasseditor.**
- ***method source editor presentation*** **pmethodeditor.**
- ***program source editor presentation*** **pprogrameditor.**
- ***variable source editor presentation*** **pvariableeditor.**
- ***function source editor presentation*** **pfunctioneditor**
- ***persistent type source editor presentation*** **ptypeeditor**
- ***persistent name source editor presentation*** **pnameeditor,**
- ***version editor presentation*** **pversionbrowser.**
- ***O_2Shell presentation*** **pshell.**

Table 4.5

Object mask resources

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
titleFontList	TitleFontList	Font	Fixed
menuBackgroundColor	MenuBackgroundColor	Color	Dynamic
menuForegroundColor	MenuForegroundColor	Color	Dynamic
menuFontList	MenuFontList	Font	Fixed

Table 4.5 lists the graphic resources to customize a browser, source editor or O_2Shell object mask with each name specifying the following:

- ***backgroundColor*** Object mask background color
- ***foregroundColor*** Object mask foreground color
- ***titleFontList*** Object mask title font
- ***menuBackgroundColor*** Browser menu background color
- ***menuForegroundColor*** Browser menu foreground color
- ***menuFontList*** Menu font

For example the schema browser object mask name is **oschemabrowser** and you use it in the following way:

```
O2Tools.pschemabrowser.oschemabrowser.foregroundColor:Midnight Blue
O2Tool*oschemabrowser.menuForegroundColor:Red
```

The application browser object mask name is **oapplibrowser**, and can be used as follows:

```
O2Tools.papplibrowser.oapplibrowser.foregroundColor: Midnight Blue
O2Tools*oapplibrowser.menuForegroundColor:Red
```

The browsers, the source and O_2Shell editors have the following object mask names:

- ***class browser object mask*** **oclassbrowser**
- ***function browser object mask*** **ofunctionbrowser**
- ***persistent type browser object mask*** **otypebrowser**
- ***persistent name browser object mask*** **onamebrowser**
- ***class source editor object mask*** **oclasseditor**
- ***method editor object mask*** **omethodeditor**
- ***program source editor object mask*** **oprogrameditor**
- ***application variable source editor object mask*** **ovariableeditor.**
- ***function source editor object mask*** **ofunctioneditor**
- ***persistent type source editor object mask*** **otypeeditor**
- ***persistent name source editor object mask*** **onameeditor**
- ***version editor object mask*** **oversionbrowser**
- ***O_2Shell object mask*** **oshell.**

4.4 Customizing the schema browser

This section gives the graphic resources for the schema browser described in Section 2.2.

Table 4.6

Graphic resources for the schema browser

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserSeparatorOn	BrowserSeparatorOn	Boolean	True
browserWidth	BrowserWidth	short	380
listsHeight	ListsHeight	short	150
labelSchemasBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelSchemasForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelSchemasFontList	LabelBrowsersFontList	Font	Fixed
listSchemasBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listSchemasForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listSchemasFontList	ListBrowsersFontList	Font	Fixed
labelBasesBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelBasesForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelBasesFontList	LabelBrowsersFontList	Font	Fixed
listBasesBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listBasesForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listBasesFontList	ListBrowsersFontList	Font	Fixed
descriptionVisible	DescriptionVisible	Boolean	True
descriptionHeight	DescriptionHeight	short	160
labelDescriptionBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelDescriptionForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelDescriptionFontList	LabelBrowsersFontList	Font	Fixed
textDescriptionBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textDescriptionForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textDescriptionFontList	TextBrowsersFontList	Font	Fixed
messageVisible	MessageVisible	Boolean	True
messageHeight	MessageHeight	short	160
labelMessageBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMessageFontList	LabelBrowsersFontList	Font	Fixed
textMessageBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textMessageFontList	TextBrowsersFontList	Font	Fixed

Table 4.6 lists the graphic resources used to customize the schema browser.

What each name in the table specifies is listed below:

- ***backgroundColor*** Schema browser background color
- ***foregroundColor*** Schema browser foreground color
- ***browserSeparatorOn*** Browser subwindows resized or not
- ***browserWidth*** Browser width
- ***listsHeight*** Height of browser lists
- ***labelSchemasBackgroundColor*** Background color of Schema list label
- ***labelSchemasForegroundColor*** Foreground color of Schema list label
- ***labelSchemasFontList*** Font of Schema list label
- ***listSchemasBackgroundColor*** Background color of Schema list
- ***listSchemasForegroundColor*** Foreground color of Schema list
- ***listSchemasFontList*** Font of Schema list
- ***labelBasesBackgroundColor*** Background color of Base list label
- ***labelBasesForegroundColor*** Foreground color of Base list label
- ***labelBasesFontList*** Font of Base list label
- ***listBasesBackgroundColor*** Background color of Base list
- ***listBasesForegroundColor*** Foreground color of Base list
- ***listBasesFontList*** Font of Base list
- ***descriptionVisible*** Description window visible or not.
- ***descriptionHeight*** Description window height
- ***labelDescriptionBackgroundColor*** Background color of Description label
- ***labelDescriptionForegroundColor*** Foreground color of Description label
- ***labelDescriptionFontList*** Font of Description label
- ***textDescriptionBackgroundColor*** Background color of Description text
- ***textDescriptionForegroundColor*** Foreground color of Description text
- ***textDescriptionFontList*** Font of Description text
- ***messageVisible*** Message window visible or not
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The schema browser name is **`schemabrowser`**.

For example:

```
O2Tools.pschemabrowser.oschemabrowser.schemabrowser.messageVisible:False
O2Tools*labelDescriptionForegroundColor: Yellow
```

To customize the schema browser presentation and object mask, refer to Section 4.3.

4.5 Customizing the application browser

This section gives the graphic resources for the application browser described in Section 2.4.

Table 4.7

Graphic resources for the Application browser

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserSeparatorOn	BrowserSeparatorOn	Boolean	True
browserWidth	BrowserWidth	short	550
listsHeight	ListsHeight	short	150
labelApplicationsBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelApplicationsForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelApplicationsFontList	LabelBrowsersFontList	Font	Fixed
listApplicationsBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listApplicationsForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listApplicationsFontList	ListBrowsersFontList	Font	Fixed
labelProgramsBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelProgramsForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelProgramsFontList	LabelBrowsersFontList	Font	Fixed
listProgramsBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listProgramsForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listProgramsFontList	ListBrowsersFontList	Font	Fixed
labelVariablesBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelVariablesForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelVariablesFontList	LabelBrowsersFontList	Font	Fixed
listVariablesBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listVariablesForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listVariablesFontList	ListBrowsersFontList	Font	Fixed
descriptionVisible	DescriptionVisible	Boolean	True
descriptionHeight	DescriptionHeight	short	160
labelDescriptionBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelDescriptionForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelDescriptionFontList	LabelBrowsersFontList	Font	Fixed
textDescriptionBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textDescriptionForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textDescriptionFontList	TextBrowsersFontList	Font	Fixed
messageVisible	MessageVisible	Boolean	True
messageHeight	MessageHeight	short	160
labelMessageBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMessageFontList	LabelBrowsersFontList	Font	Fixed
textMessageBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textMessageFontList	TextBrowsersFontList	Font	Fixed

Table 4.7 lists the graphic resources used to customize the application browser.

What each name in the above table specifies is given below:

- ***backgroundColor*** Browser background color
- ***foregroundColor*** Browser foreground color
- ***BrowserSeparatorOn*** Browser subwindows resized or not
- ***BrowserWidth*** Browser width
- ***listsHeight*** Height of browser lists
- ***labelApplicationsBackgroundColor*** Background col. of Application list label
- ***labelApplicationsForegroundColor*** Foreground col. of Application list label
- ***labelApplicationsFontList*** Font of Application list label
- ***listApplicationsBackgroundColor*** Background color of Application list
- ***listApplicationsForegroundColor*** Foreground color of Application list
- ***listApplicationsFontList*** Font of Application list
- ***labelProgramsBackgroundColor*** Background color of Program list label
- ***labelProgramsForegroundColor*** Foreground color of Program list label
- ***labelProgramsFontList*** Font of Program list label
- ***listProgramsBackgroundColor*** Background color of Program list
- ***listProgramsForegroundColor*** Foreground color of Program list
- ***listProgramsFontList*** Font of Program list
- ***labelVariablesBackgroundColor*** Background color of Variable list label
- ***labelVariablesForegroundColor*** Foreground color of Variable list label
- ***labelVariablesFontList*** Font of Variable list label
- ***listVariablesBackgroundColor*** Background color of Program list
- ***listVariablesForegroundColor*** Foreground color of Program list
- ***descriptionVisible*** Description window visible or not
- ***descriptionHeight*** Description window height
- ***labelDescriptionBackgroundColor*** Background color of Description label
- ***labelDescriptionForegroundColor*** Foreground color of Description label
- ***labelDescriptionFontList*** Font of Description label
- ***textDescriptionBackgroundColor*** Background color of Description text
- ***textDescriptionForegroundColor*** Foreground color of Description text
- ***textDescriptionFontList*** Font of Description text
- ***messageVisible*** Message window visible or not
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The application browser name is **`applibrowser`**, e.g.

```
O2Tools.papplicationbrowser.oapplibrowser.applibrowser.hierarchyVisible:True
O2Tools*lableHierarchyForegroundColor:Yellow
```

To customize the application browser presentation and object mask, refer to Section 4.3.

4.6 Customizing the class browser

This section gives you the resources for the class browser described in Section 2.3. Table 4.8 lists the graphic resources used to customize the class browser. :

Table 4.8 **Graphic resources for the Class Browser**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserSeparatorOn	BrowserSeparatorOn	Boolean	True
browserWidth	BrowserWidth	short	550
listsHeight	ListsHeight	short	150
labelClassesBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelClassesForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelClassesFontList	LabelBrowsersFontList	Font	Fixed
listClassesBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listClassesForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listClassesFontList	ListBrowsersFontList	Font	Fixed
labelMethodsBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMethodsForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMethodsFontList	LabelBrowsersFontList	Font	Fixed
listMethodsBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listMethodsForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listMethodsFontList	ListBrowsersFontList	Font	Fixed
labelNamedObjectsBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelNamedObjectsForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelNamedObjectsFontList	LabelBrowsersFontList	Font	Fixed
listNamedObjectsBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listNamedObjectsForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listNamedObjectsFontList	ListBrowsersFontList	Font	Fixed
descriptionVisible	DescriptionVisible	Boolean	True
descriptionHeight	DescriptionHeight	short	160
labelDescriptionBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelDescriptionForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelDescriptionFontList	LabelBrowsersFontList	Font	Fixed
textDescriptionBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textDescriptionForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textDescriptionFontList	TextBrowsersFontList	Font	Fixed
hierarchyVisible	HierarchyVisible	Boolean	True
hierarchyHeight	HierarchyHeight	short	320
labelHierarchyBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelHierarchyForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelHierarchyFontList	LabelBrowsersFontList	Font	Fixed
messageVisible	MessageVisible	Boolean	True
messageHeight	MessageHeight	short	160

Table 4.8

Graphic resources for the Class Browser

Name	Class	Type	Default
labelMessageBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMessageFontList	LabelBrowsersFontList	Font	Fixed
textMessageBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textMessageFontList	TextBrowsersFontList	Font	Fixed

What each name specifies is listed below

- ***backgroundColor*** Browser background color
- ***foregroundColor*** Browser foreground color
- ***browserSeparatorOn*** Browser subwindows resized or not
- ***browserWidth*** Browser width
- ***listsHeight*** Height of browser lists
- ***labelClassesBackgroundColor*** Background color of Class list label
- ***labelClassesForegroundColor*** Foreground color of Class list label
- ***labelClassesFontList*** Font of Class list label
- ***listClassesBackgroundColor*** Background color of Class list
- ***listClassesForegroundColor*** Foreground color of Class list
- ***listClassesFontList*** Font of Class list
- ***labelMethodsBackgroundColor*** Background col. of Method list label
- ***labelMethodsForegroundColor*** Foreground col. of Method list label
- ***labelMethodsFontList*** Font of Class Method label
- ***listMethodsBackgroundColor*** Background color of Method list
- ***listMethodsForegroundColor*** Foreground color of Method list
- ***listMethodsFontList*** Font of Method list
- ***labelNamedObjectsBackgroundColor*** Background col. of Name list label
- ***labelNamedObjectsForegroundColor*** Foreground color of Name list label
- ***labelNamedObjectsFontList*** Font of Name list label
- ***listNamedObjectsBackgroundColor*** Background color of Name list
- ***listNamedObjectsForegroundColor*** Foreground color of Name list
- ***listNamedObjectsFontList*** Font of Name list
- ***descriptionVisible*** Description window visible or not.
- ***descriptionHeight*** Description window height
- ***labelDescriptionBackgroundColor*** Background col. of Description label
- ***labelDescriptionForegroundColor*** Foreground col. of Description label
- ***labelDescriptionFontList*** Font of Description label
- ***textDescriptionBackgroundColor*** Background col. of Description text
- ***textDescriptionForegroundColor*** Foreground col. of Description text
- ***textDescriptionFontList*** Font of Description text
- ***messageVisible*** Message window visible or not
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

- ***hierarchyVisible*** Hierarchy window visible or not
- ***hierarchyHeight*** Hierarchy window height
- ***labelHierarchyBackgroundColor*** Background color of Hierarchy label
- ***labelHierarchyForegroundColor*** Foreground col. of Hierarchy label
- ***labelHierarchyFontList*** Font of Hierarchy label

The class browser name is **`classbrowser`**, e.g.

```
O2Tools.pclassbrowser.oclassbrowser.classbrowser.hierarchyVisible:True
O2Tools*labelHierarchyForegroundColor: Yellow
```

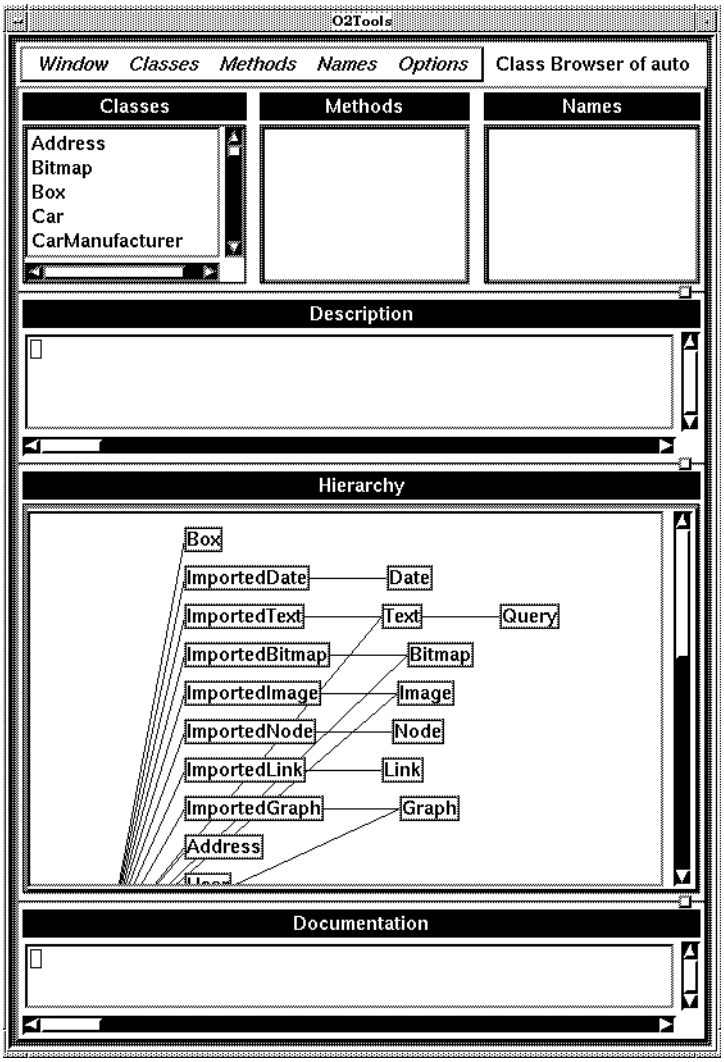


Figure 4.189: Class Browser showing hierarch window

To customize the class browser presentation and object mask, refer to Section 4.3.

Table 4.9 **Graphic resources for the class browser class hierarchy**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
hgap	Hgap	short	60
vgap	Vgap	short	10
linkThickness	LinkThickness	short	0
linkColor	LinkColor	Color	Dynamic
selThickness	SelThickness	short	4
selColor	selColor	Color	Dynamic

Table 4.9 lists the graphic resources you can customize for the class browser class hierarchy.

- ***backgroundColor*** Class hierarchy background color
- ***foregroundColor*** Class hierarchy foreground color
- ***hgap*** Horizontal distance between class hierarchy nodes
- ***vgap*** Vertical distance between class hierarchy nodes
- ***linkThickness*** Thickness of class hierarchy links
- ***linkColor*** Color of class hierarchy links
- ***selThickness*** Thickness of selected class hierarchy links
- ***selColor*** Color of selected class hierarchy links

The name of the class hierarchy mask is **`hierarchy`**, e.g.

```
O2Tools.pclassbrowser.oclassbrowser.classbrowser.hierarchy.vgap: 10
```

4.7 Customizing the function browser

This section gives you the resources for the function browser described in Section 2.5. Table 4.10 lists the graphic resources used to customize the function browser.

Table 4.10 **Function browser graphic resources**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserSeparatorOn	BrowserSeparatorOn	Boolean	True
browserWidth	BrowserWidth	short	290
listsHeight	ListsHeight	short	150
labelFunctionsBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelFunctionsForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelFunctionsFontList	LabelBrowsersFontList	Font	Fixed
listFunctionsBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listFunctionsForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listFunctionsFontList	ListBrowsersFontList	Font	Fixed
descriptionVisible	DescriptionVisible	Boolean	True
descriptionHeight	DescriptionHeight	short	160
labelDescriptionBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelDescriptionForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelDescriptionFontList	LabelBrowsersFontList	Font	Fixed
textDescriptionBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textDescriptionForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textDescriptionFontList	TextBrowsersFontList	Font	Fixed
messageVisible	MessageVisible	Boolean	True
messageHeight	MessageHeight	short	160
labelMessageBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMessageFontList	LabelBrowsersFontList	Font	Fixed
textMessageBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textMessageFontList	TextBrowsersFontList	Font	Fixed

What each name specifies is listed below

- ***backgroundColor*** Browser background color
- ***foregroundColor*** Browser foreground color
- ***browserSeparatorOn*** Browser sub-windows resized or not
- ***BrowserWidth*** Browser width
- ***listsHeight*** Height of browser lists
- ***labelFunctionsBackgroundColor*** Background col. of Function list label
- ***labelFunctionsForegroundColor*** Foreground col. of Function list label
- ***labelFunctionsFontList*** Font of Function list label
- ***listFunctionsBackgroundColor*** Background color of Function list
- ***listFunctionsForegroundColor*** Foreground color of Function list
- ***listFunctionsFontList*** Font of Function list
- ***descriptionVisible*** Description window visible or not.
- ***descriptionHeight*** Description window height
- ***labelDescriptionBackgroundColor*** Background color of Description label

- ***labelDescriptionForegroundColor*** Foreground color of Description label
- ***labelDescriptionFontList*** Font of Description label
- ***textDescriptionBackgroundColor*** Background color of Description text
- ***textDescriptionForegroundColor*** Foreground color of Description text
- ***textDescriptionFontList*** Font of Description text
- ***messageVisible*** Message window visible or not
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text.

The function browser name is **`functionbrowser`**, e.g.

```
O2Tools.pfunctionbrowser.ofunctionbrowser.functionbrowser.messageVisible:True

O2Tools*labelDescriptionForegroundColor: Yellow.
```

To customize the function browser presentation and object mask, refer to Section 4.3.

4.8 Customizing the persistent type browser

This section gives you the resources for the Persistent type browser described in Section 2.6.

Table 4.11 **Persistent type browser graphic resources**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserSeparatorOn	BrowserSeparatorOn	Boolean	True
browserWidth	BrowserWidth	short	290
listsHeight	ListsHeight	short	150
labelTypesBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelTypesForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelTypesFontList	LabelBrowsersFontList	Font	Fixed
listTypesBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic

Table 4.11

Persistent type browser graphic resources

Name	Class	Type	Default
listTypesForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listTypesFontList	ListBrowsersFontList	Font	Fixed
descriptionVisible	DescriptionVisible	Boolean	True
descriptionHeight	DescriptionHeight	short	160
labelDescriptionBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelDescriptionForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelDescriptionFontList	LabelBrowsersFontList	Font	Fixed
textDescriptionBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textDescriptionForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textDescriptionFontList	TextBrowsersFontList	Font	Fixed
messageVisible	MessageVisible	Boolean	True
messageHeight	MessageHeight	short	160
labelMessageBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMessageFontList	LabelBrowsersFontList	Font	Fixed
textMessageBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textMessageFontList	TextBrowsersFontList	Font	Fixed

Table 4.11 lists the graphic resources you can customize for the persistent type browser.

What each name specifies is listed below:

- ***backgroundColor*** Browser background color
- ***foregroundColor*** Browser foreground color
- ***browserSeparatorOn*** Browser sub-windows resized or not
- ***browserWidth*** Browser width
- ***listsHeight*** Height of browser lists
- ***labelTypesBackgroundColor*** Background color of Type list label
- ***labelTypesForegroundColor*** Foreground color of Type list label
- ***labelTypesFontList*** Font of Type list label
- ***listTypesBackgroundColor*** Background color of Type list
- ***listTypesForegroundColor*** Foreground color of Type list
- ***listTypesFontList*** Font of Type list
- ***descriptionVisible*** Description window visible or not.
- ***descriptionHeight*** Description window height
- ***labelDescriptionBackgroundColor*** Background color of Description label
- ***labelDescriptionForegroundColor*** Foreground color of Description label
- ***labelDescriptionFontList*** Font of Description label
- ***textDescriptionBackgroundColor*** Background color of Description text
- ***textDescriptionForegroundColor*** Foreground color of Description text
- ***textDescriptionFontList*** Font of Description text

- ***messageVisible*** Message window visible or not
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The name of the persistent type browser is **`typebrowser`**.

For example

```
O2Tools.ptypebrowser.otypebrowser.typebrowser.messageVisible:True
```

```
O2Tools*labelDescriptionForegroundColor: Yellow
```

To customize the persistent type browser presentation and object mask, refer to Section 4.3.

4.9 Customizing the persistent name browser

This section gives you the resources for the Persistent type browser described in Section 2.7.

Table 4.12 **Persistent name browser graphic resources**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserSeparatorOn	BrowserSeparatorOn	Boolean	True
browserWidth	BrowserWidth	short	290
listsHeight	ListsHeight	short	150
labelNamesBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelNamesForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelNamesFontList	LabelBrowsersFontList	Font	Fixed
listnamesBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listNamesForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listNamesFontList	ListBrowsersFontList	Font	Fixed

Table 4.12 **Persistent name browser graphic resources**

Name	Class	Type	Default
descriptionVisible	DescriptionVisible	Boolean	True
descriptionHeight	DescriptionHeight	short	160
labelDescriptionBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelDescriptionForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelDescriptionFontList	LabelBrowsersFontList	Font	Fixed
textDescriptionBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textDescriptionForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textDescriptionFontList	TextBrowsersFontList	Font	Fixed
messageVisible	MessageVisible	Boolean	True
messageHeight	MessageHeight	short	160
labelMessageBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMessageFontList	LabelBrowsersFontList	Font	Fixed
textMessageBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textMessageFontList	TextBrowsersFontList	Font	Fixed

Table 4.11 lists the graphic resources you can customize for the persistent name browser. What each name specifies is listed below.

What each name in the table specifies is listed below:

- ***backgroundColor*** Browser background color
- ***foregroundColor*** Browser foreground color
- ***browserSeparatorOn*** Browser sub-windows resized or not
- ***browserWidth*** Browser width
- ***listsHeight*** Height of browser lists
- ***labelNamesBackgroundColor*** Background color of Name list label
- ***labelNamesForegroundColor*** Foreground color of Name list label
- ***labelNamesFontList*** Font of Name list label
- ***listNamesBackgroundColor*** Background color of Name list
- ***listNamesForegroerundColor*** Foreground color of Name list
- ***listNamesFontList*** Font of Name list
- ***descriptionVisible*** Description window visible or not.
- ***descriptionHeight*** Description window height
- ***labelDescriptionBackgroundColor*** Background color of Description label
- ***labelDescriptionForegroundColor*** Foreground color of Description label
- ***labelDescriptionFontList*** Font of Description label
- ***textDescriptionBackgroundColor*** Background color of Description text
- ***textDescriptionForegroundColor*** Foreground color of Description text
- ***textDescriptionFontList*** Font of Description text
- ***messageVisible*** Message window visible or not
- ***messageHeight*** Message window height

- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The name of the persistent name browser is **`namebrowser`**.

For example,

```
O2Tools.pnamebrowser.onamebrowser.namebrowser. messageVisible: True
O2Tools*labelDescriptionForegroundColor:Yellow
```

To customize the persistent name browser presentation and object mask, refer to Section 4.3.

4.10 Customizing the class source editor

This section gives you the resources for the Persistent type browser described in Section 3.5.

Table 4.13 **Class source editor graphic resources**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
typeVisible	TypeVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
editorSeparatorOn	EditorSeparatorOn	Boolean	True
editorWidth	EditorWidth	short	360
labelInfoBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelInfoForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelInfoFontList	LabelEditorsFontList	Font	Fixed
infoBackgroundColor	InfoEditorsBackgroundColor	Color	Dynamic
infoForegroundColor	InfoEditorsForegroundColor	Color	Dynamic
infoFontList	InfoEditorsFontList	Font	Fixed
infoToggleSelectColor	InfoToggleSelectColor	Color	Dynamic
typeHeight	TypeHeight	short	160
labelTypeBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelTypeForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelTypeFontList	LabelEditorsFontList	Font	Fixed

Table 4.13

Class source editor graphic resources

Name	Class	Type	Default
textTypeBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textTypeForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textTypeFontList	TextEditorsFontList	Font	Fixed
textTypeShowLines	TextTypeShowLines	Boolean	True
textTypeNumWidth	TextTypeNumWidth	int	50
textTypeNumColor	TextTypeNumColor	Color	Dynamic
textTypeTabSize	TextTypeTabSize	int	4
messageHeight	MessageHeight	short	130
labelMessageBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelMessageFontList	LabelEditorsFontList	Font	Fixed
textMessageBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textMessageFontList	TextEditorsFontList	Font	Fixed

Table 4.13 lists the graphic resources you can customize for the class source editor.

Each name is specified below.

- ***backgroundColor*** Editor background color
- ***foregroundColor*** Editor foreground color
- ***typeVisible*** Type window visible or not.
- ***messageVisible*** Message window visible or not
- ***editorSeparatorOn*** Editor sub-windows resized or not
- ***editorWidth*** Editor width
- ***labelInfoBackgroundColor*** Background col. of Visibility or State label
- ***labelInfoForegroundColor*** Foreground col. of Visibility or State label
- ***labelInfoFontList*** Font of Visibility or State window label
- ***infoBackgroundColor*** Background col. of Visibility or State window
- ***infoForegroundColor*** Foreground col.of Visibility or State window
- ***infoFontList*** Font of Visibility or State window
- ***infoToggleSelectColor*** Toggle for col. of Visibility or State window
- ***typeHeight*** Type window height
- ***labelTypeBackgroundColor*** Background color of Type label
- ***labelTypeForegroundColor*** Foreground color of Type label
- ***labelTypeFontList*** Font of Type label
- ***textTypeBackgroundColor*** Background color of Type text
- ***textTypeForegroundColor*** Foreground color of Type text
- ***textTypeFontList*** Font of Type text
- ***textTypeShowLines*** Show the line numbers in window
- ***textTypeNumWidth*** Line number column width
- ***textTypeNumColor*** Line number color

- ***textTypeTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The name of the class source editor is **classeditor**. e.g

```
O2Tools.pclasseditor.oclasseditor.classeditor. messageBackgroundColor: Midnight blue
O2Tools*classeditor.messageVisible: True
```

To customize the source editor presentation and object mask, refer to Section 4.3.

4.11 Customizing the method source editor

This section describes the graphic resources you need to customize the method source editor described in Section 3.6.

Table 4.14

Graphic resources for the Method source editor

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
signatureVisible	SignatureVisible	Boolean	True
bodyVisible	BodyVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
editorSeparatorOn	EditorSeparatorOn	Boolean	True
editorWidth	EditorWidth	short	550
labelInfoBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelInfoForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelInfoFontList	LabelEditorsFontList	Font	Fixed
infoBackgroundColor	InfoEditorsBackgroundColor	Color	Dynamic
infoForegroundColor	InfoEditorsForegroundColor	Color	Dynamic
infoFontList	InfoEditorsFontList	Font	Fixed
infoToggleSelectColor	InfoToggleSelectColor	Color	Dynamic
signatureHeight	SignatureHeight	short	120
labelSignatureBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelSignatureForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelSignatureFontList	LabelEditorsFontList	Font	Fixed
textSignatureBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textSignatureForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textSignatureFontList	TextEditorsFontList	Font	Fixed

Table 4.14

Graphic resources for the Method source editor

Name	Class	Type	Default
textSignatureShowLines	TextSignatureShowLines	Boolean	True
textSignatureNumWidth	TextSignatureNumWidth	int	50
textSignatureNumColor	TextSignatureNumColor	Color	Dynamic
textSignatureTabSize	TextSignatureTabSize	int	4
bodyHeight	BodyHeight	short	400
labelBodyBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelBodyForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelBodyFontList	LabelEditorsFontList	Font	Fixed
textBodyBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textBodyForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textBodyFontList	TextEditorsFontList	Font	Fixed
textBodyShowLines	TextBodyShowLines	Boolean	True
textBodyNumWidth	TextBodyNumWidth	int	50
textBodyNumColor	TextBodyNumColor	Color	Dynamic
textBodyTabSize	TextBodyTabSize	int	4
messageHeight	MessageHeight	short	150
labelMessageBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelMessageFontList	LabelEditorsFontList	Font	Fixed
textMessageBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textMessageFontList	TextEditorsFontList	Font	Fixed

Table 4.14 lists the graphic resources for the method source editor. The specifications of each name are detailed below:

- ***backgroundColor*** Editor background color
- ***foregroundColor*** Editor foreground color
- ***bodyVisible*** Body window visible or not.
- ***signatureVisible*** Signature window visible or not.
- ***messageVisible*** Message window visible or not
- ***editorSeparatorOn*** Editor sub-windows resized or not
- ***editorWidth*** Editor width
- ***labelInfoBackgroundColor*** Background col. of Visibility or State label
- ***labelInfoForegroundColor*** Foreground col. of Visibility or State label
- ***labelInfoFontList*** Font of Visibility or State window label
- ***infoBackgroundColor*** Background col. of Visibility or State window
- ***infoForegroundColor*** Foreground col.of Visibility or State window
- ***infoFontList*** Font of Visibility or State window
- ***infoToggleSelectColor*** Toggle for col. of Visibility or State window
- ***signatureHeight*** Signature window height
- ***labelSignatureBackgroundColor*** Background color of Signature label
- ***labelSignatureForegroundColor*** Foreground color of Signature label
- ***labelSignatureFontList*** Font of Signature label
- ***textSignatureBackgroundColor*** Background color of Signature text
- ***textSignatureForegroundColor*** Foreground color of Signature text
- ***textSignatureFontList*** Font of Signature text
- ***textSignatureShowLines*** Show line numbers in Signature
- ***textSignatureNumWidth*** Line number column width

- ***textSignatureNumColor*** Line number color
- ***textSignatureTabSize*** Tabulation size
- ***bodyHeight*** Body window height
- ***labelBodyBackgroundColor*** Background color of Body label
- ***labelBodyForegroundColor*** Foreground color of Body label
- ***labelBodyFontList*** Font of Body label
- ***textBodyBackgroundColor*** Background color of Body text
- ***textBodyForegroundColor*** Foreground color of Body text
- ***textBodyFontList*** Font of Body text
- ***textBodyShowLines*** Show line numbers in the Body
- ***textBodyNumWidth*** Line number column width
- ***textBodyNumColor*** Line number color
- ***textBodyTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text.

The name of the method source editor is **`methodeditor`**. e.g.

```
O2Tools.pmethodeditor.omethodeditor.methodeditor.messageBackgroundColor:Midnightblue
O2Tools*methodeditor.messageVisible: True
```

4.12 Customizing the program source editor

This section details the graphic resources for the Program source editor described in Section 3.7.

Table 4.15

Graphic resources for the Program source editor

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
signatureVisible	SignatureVisible	Boolean	True
bodyVisible	BodyVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
editorSeparatorOn	EditorSeparatorOn	Boolean	True
editorWidth	EditorWidth	short	150
labelInfoBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelInfoForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelInfoFontList	LabelEditorsFontList	Font	Fixed
infoBackgroundColor	InfoEditorsBackgroundColor	Color	Dynamic
infoForegroundColor	InfoEditorsForegroundColor	Color	Dynamic
infoFontList	InfoEditorsFontList	Font	Fixed
infoToggleSelectColor	InfoToggleSelectColor	Color	Dynamic
signatureHeight	SignatureHeight	short	120

Table 4.15

Graphic resources for the Program source editor

Name	Class	Type	Default
labelSignatureBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelSignatureForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelSignatureFontList	LabelEditorsFontList	Font	Fixed
textSignatureBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textSignatureForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textSignatureFontList	TextEditorsFontList	Font	Fixed
textSignatureShowLines	TextSignatureShowLines	Boolean	True
textSignatureNumWidth	TextSignatureNumWidth	int	50
textSignatureNumColor	TextSignatureNumColor	Color	Dynamic
textSignatureTabSize	TextSignatureTabSize	int	4
bodyHeight	BodyHeight	short	400
labelBodyBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelBodyForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelBodyFontList	LabelEditorsFontList	Font	Fixed
textBodyBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textBodyForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textBodyFontList	TextEditorsFontList	Font	Fixed
textBodyShowLines	TextBodyShowLines	Boolean	True
textBodyNumWidth	TextBodyNumWidth	int	50
textBodyNumColor	TextBodyNumColor	Color	Dynamic
textBodyTabSize	TextBodyTabSize	int	4
messageHeight	MessageHeight	short	150
labelMessageBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelMessageFontList	LabelEditorsFontList	Font	Fixed
textMessageBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textMessageFontList	TextEditorsFontList	Font	Fixed

Table 4.15 lists the graphic resources you can customize for the program source editor. Each name specification is listed below.

- ***backgroundColor*** Editor background color
- ***foregroundColor*** Editor foreground color
- ***bodyVisible*** Body window visible or not.
- ***signatureVisible*** Signature window visible or not.
- ***messageVisible*** Message window visible or not
- ***editorSeparatorOn*** Editor sub-windows resized or not
- ***editorWidth*** Editor width
- ***labelInfoBackgroundColor*** Background col. of Visibility or State label
- ***labelInfoForegroundColor*** Foreground col. of Visibility or State label
- ***labelInfoFontList*** Font of Visibility or State window label
- ***infoBackgroundColor*** Background col. of Visibility or State window
- ***infoForegroundColor*** Foreground col.of Visibility or State window
- ***infoFontList*** Font of Visibility or State window
- ***infoToggleSelectColor*** Toggle for col. of Visibility or State window
- ***signatureHeight*** Signature window height
- ***labelSignatureBackgroundColor*** Background color of Signature label
- ***labelSignatureForegroundColor*** Foreground color of Signature label

- ***labelSignatureFontList*** Font of Signature label
- ***textSignatureBackgroundColor*** Background color of Signature text
- ***textSignatureForegroundColor*** Foreground color of Signature text
- ***textSignatureFontList*** Font of Signature text
- ***textSignatureShowLines*** Show line numbers in Signature
- ***textSignatureNumWidth*** Line number column width
- ***textSignatureNumColor*** Line number color
- ***textSignatureTabSize*** Tabulation size
- ***bodyHeight*** Body window height
- ***labelBodyBackgroundColor*** Background color of Body label
- ***labelBodyForegroundColor*** Foreground color of Body label
- ***labelBodyFontList*** Font of Body label
- ***textBodyBackgroundColor*** Background color of Body text
- ***textBodyForegroundColor*** Foreground color of Body text
- ***textBodyFontList*** Font of Body text
- ***textBodyShowLines*** Show line numbers in the Body
- ***textBodyNumWidth*** Line number column width
- ***textBodyNumColor*** Line number color
- ***textBodyTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The name of the program source editor is **programeditor**. e.g.

```
O2Tools.pprogrameditor.oprogrameditor.programeditor.messageBackgroundColor:Midnight blue
O2Tools*programeditor.messageVisible: True
```

4.13 Customizing the variable source editor

This section details the graphic resources for the Application variable source editor described in Section 3.8.

Table 4.16 **Application variable source editor resources**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
typeVisible	TypeVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
editorSeparatorOn	EditorSeparatorOn	Boolean	True
editorWidth	EditorWidth	short	250
labelInfoBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic

Table 4.16

Application variable source editor resources

Name	Class	Type	Default
labelInfoForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelInfoFontList	LabelEditorsFontList	Font	Fixed
infoBackgroundColor	InfoEditorsBackgroundColor	Color	Dynamic
infoForegroundColor	InfoEditorsForegroundColor	Color	Dynamic
infoFontList	InfoEditorsFontList	Font	Fixed
infoToggleSelectColor	InfoToggleSelectColor	Color	Dynamic
typeHeight	TypeHeight	short	130
labelTypeBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelTypeForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelTypeFontList	LabelEditorsFontList	Font	Fixed
textTypeBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textTypeForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textTypeFontList	TextEditorsFontList	Font	Fixed
textTypeShowLines	TextTypeShowLines	Boolean	True
textTypeNumWidth	TextTypeNumWidth	int	50
textTypeNumColor	TextTypeNumColor	Color	Dynamic
textTypeTabSize	TextTypeTabSize	int	4
messageHeight	MessageHeight	short	130
labelMessageBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelMessageFontList	LabelEditorsFontList	Font	Fixed
textMessageBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textMessageFontList	TextEditorsFontList	Font	Fixed

Table 4.16 lists the graphic resources used to customize the application variable source editor.

Each name specification is listed below.

- ***backgroundColor*** Editor background color
- ***foregroundColor*** Editor foreground color
- ***typeVisible*** Type window visible or not.
- ***messageVisible*** Message window visible or not
- ***editorSeparatorOn*** Editor sub-windows resized or not
- ***editorWidth*** Editor width
- ***labelInfoBackgroundColor*** Background col. of Visibility or State label
- ***labelInfoForegroundColor*** Foreground col. of Visibility or State label
- ***labelInfoFontList*** Font of Visibility or State window label
- ***infoBackgroundColor*** Background col. of Visibility or State window
- ***infoForegroundColor*** Foreground col.of Visibility or State window
- ***infoFontList*** Font of Visibility or State window
- ***infoToggleSelectColor*** Toggle for col. of Visibility or State window

- ***typeHeight*** Type window height
- ***labelTypeBackgroundColor*** Background color of Type label
- ***labelTypeForegroundColor*** Foreground color of Type label
- ***labelTypeFontList*** Font of Type label
- ***textTypeBackgroundColor*** Background color of Type text
- ***textTypeForegroundColor*** Foreground color of Type text
- ***textTypeFontList*** Font of Type text
- ***textTypeShowLines*** Show the line numbers in window
- ***textTypeNumWidth*** Line number column width
- ***textTypeNumColor*** Line number color
- ***textTypeTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The name of the variable source editor is **variableeditor**.

For example,

```
O2Tools.pvariableeditor.ovariableeditor.variableeditor.messageBackgroundColor:Midnightblue
O2Tools*variableeditor.messageVisible: True
```

To customize the source editor presentation and object mask, refer to Section 4.3.

4.14 Customizing the function source editor

This section details the graphic resources for the Function source editor described in Section 3.9.

Table 4.17

Function source editor graphic resources

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
signatureVisible	SignatureVisible	Boolean	True
bodyVisible	BodyVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
editorSeparatorOn	EditorSeparatorOn	Boolean	True
editorWidth	EditorWidth	short	450

Table 4.17

Function source editor graphic resources

Name	Class	Type	Default
labelInfoBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelInfoForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelInfoFontList	LabelEditorsFontList	Font	Fixed
infoBackgroundColor	InfoEditorsBackgroundColor	Color	Dynamic
infoForegroundColor	InfoEditorsForegroundColor	Color	Dynamic
infoFontList	InfoEditorsFontList	Font	Fixed
infoToggleSelectColor	InfoToggleSelectColor	Color	Dynamic
signatureHeight	SignatureHeight	short	120
labelSignatureBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelSignatureForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelSignatureFontList	LabelEditorsFontList	Font	Fixed
textSignatureBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textSignatureForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textSignatureFontList	TextEditorsFontList	Font	Fixed
textSignatureShowLines	TextSignatureShowLines	Boolean	True
textSignatureNumWidth	TextSignatureNumWidth	int	50
textSignatureNumColor	TextSignatureNumColor	Color	Dynamic
textSignatureTabSize	TextSignatureTabSize	int	4
bodyHeight	BodyHeight	short	400
labelBodyBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelBodyForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelBodyFontList	LabelEditorsFontList	Font	Fixed
textBodyBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textBodyForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textBodyFontList	TextEditorsFontList	Font	Fixed
textBodyShowLines	TextBodyShowLines	Boolean	True
textBodyNumWidth	TextBodyNumWidth	int	50
textBodyNumColor	TextBodyNumColor	Color	Dynamic
textBodyTabSize	TextBodyTabSize	int	4
messageHeight	MessageHeight	short	150
labelMessageBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelMessageFontList	LabelEditorsFontList	Font	Fixed
textMessageBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textMessageFontList	TextEditorsFontList	Font	Fixed

Table 4.17 lists the graphic resources used to customize the function source editor. What each name specifies is listed below:

- ***backgroundColor*** Editor background color
- ***foregroundColor*** Editor foreground color
- ***bodyVisible*** Body window visible or not.
- ***signatureVisible*** Signature window visible or not.
- ***messageVisible*** Message window visible or not
- ***editorSeparatorOn*** Editor sub-windows resized or not
- ***editorWidth*** Editor width
- ***labelInfoBackgroundColor*** Background col. of Visibility or State label
- ***labelInfoForegroundColor*** Foreground col. of Visibility or State label
- ***labelInfoFontList*** Font of Visibility or State window label

- ***infoBackgroundColor*** Background col. of Visibility or State window
- ***infoForegroundColor*** Foreground col.of Visibility or State window
- ***infoFontList*** Font of Visibility or State window
- ***infoToggleSelectColor*** Toggle for col. of Visibility or State window
- ***signatureHeight*** Signature window height
- ***labelSignatureBackgroundColor*** Background color of Signature label
- ***labelSignatureForegroundColor*** Foreground color of Signature label
- ***labelSignatureFontList*** Font of Signature label
- ***textSignatureBackgroundColor*** Background color of Signature text
- ***textSignatureForegroundColor*** Foreground color of Signature text
- ***textSignatureFontList*** Font of Signature text
- ***textSignatureShowLines*** Show line numbers in Signature
- ***textSignatureNumWidth*** Line number column width
- ***textSignatureNumColor*** Line number color
- ***textSignatureTabSize*** Tabulation size
- ***bodyHeight*** Body window height
- ***labelBodyBackgroundColor*** Background color of Body label
- ***labelBodyForegroundColor*** Foreground color of Body label
- ***labelBodyFontList*** Font of Body label
- ***textBodyBackgroundColor*** Background color of Body text
- ***textBodyForegroundColor*** Foreground color of Body text
- ***textBodyFontList*** Font of Body text
- ***textBodyShowLines*** Show line numbers in the Body
- ***textBodyNumWidth*** Line number column width
- ***textBodyNumColor*** Line number color
- ***textBodyTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The name of the function source editor is **`functioneditor`**, e.g.

```
O2Tools.pfunctioneditor.ofunctioneditor.functioneditor.messageBackgroundColor:Midnight blue
O2Tools*functioneditor.messageVisible: True
```

4.15 Customizing persistent type source editor

This section details the graphic resources for the Persistent type source editor described in Section 3.10.

Table 4.18 **Persistent type source editor resources**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
typeVisible	TypeVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
editorSeparatorOn	EditorSeparatorOn	Boolean	True
editorWidth	EditorWidth	short	250
labelInfoBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelInfoForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelInfoFontList	LabelEditorsFontList	Font	Fixed
infoBackgroundColor	InfoEditorsBackgroundColor	Color	Dynamic
infoForegroundColor	InfoEditorsForegroundColor	Color	Dynamic
infoFontList	InfoEditorsFontList	Font	Fixed
infoToggleSelectColor	InfoToggleSelectColor	Color	Dynamic
typeHeight	TypeHeight	short	130
labelTypeBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelTypeForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelTypeFontList	LabelEditorsFontList	Font	Fixed
textTypeBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textTypeForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textTypeFontList	TextEditorsFontList	Font	Fixed
textTypeShowLines	TextTypeShowLines	Boolean	True
textTypeNumWidth	TextTypeNumWidth	int	50
textTypeNumColor	TextTypeNumColor	Color	Dynamic
textTypeTabSize	TextTypeTabSize	int	4
messageHeight	MessageHeight	short	130
labelMessageBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelMessageFontList	LabelEditorsFontList	Font	Fixed
textMessageBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textMessageFontList	TextEditorsFontList	Font	Fixed

Table 4.18 lists the graphic resources used to customize the persistent type source editor with each name specified below.

- ***backgroundColor*** Editor background color
- ***foregroundColor*** Editor foreground color
- ***typeVisible*** Type window visible or not.
- ***messageVisible*** Message window visible or not
- ***editorSeparatorOn*** Editor sub-windows resized or not
- ***editorWidth*** Editor width
- ***labelInfoBackgroundColor*** Background col. of Visibility or State label
- ***labelInfoForegroundColor*** Foreground col. of Visibility or State label
- ***labelInfoFontList*** Font of Visibility or State window label

- ***infoBackgroundColor*** Background col. of Visibility or State window
- ***infoForegroundColor*** Foreground col.of Visibility or State window
- ***infoFontList*** Font of Visibility or State window
- ***infoToggleSelectColor*** Toggle for col. of Visibility or State window
- ***typeHeight*** Type window height
- ***labelTypeBackgroundColor*** Background color of Type label
- ***labelTypeForegroundColor*** Foreground color of Type label
- ***labelTypeFontList*** Font of Type label
- ***textTypeBackgroundColor*** Background color of Type text
- ***textTypeForegroundColor*** Foreground color of Type text
- ***textTypeFontList*** Font of Type text
- ***textTypeShowLines*** Show the line numbers in window
- ***textTypeNumWidth*** Line number column width
- ***textTypeNumColor*** Line number color
- ***textTypeTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The persistent type source name is **`typeeditor`**, e.g.

```
O2Tools.ptypeeditor.otypeeditor.typeeditor.messageBackgroundColo: Midnight blue

O2Tools*typeeditor messageVisible True
```

To customize the source editor presentation and object mask, refer to Section 4.3.

4.16 Customizing persistent name source editor

This section details the graphic resources for the Persistent name source editor described in Section 3.11.

Table 4.19

Persistent name source editor resources

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
typeVisible	TypeVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
editorSeparatorOn	EditorSeparatorOn	Boolean	True
editorWidth	EditorWidth	short	350
labelInfoBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelInfoForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelInfoFontList	LabelEditorsFontList	Font	Fixed
infoBackgroundColor	InfoEditorsBackgroundColor	Color	Dynamic
infoForegroundColor	InfoEditorsForegroundColor	Color	Dynamic
infoFontList	InfoEditorsFontList	Font	Fixed
infoToggleSelectColor	InfoToggleSelectColor	Color	Dynamic
typeHeight	TypeHeight	short	130
labelTypeBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelTypeForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelTypeFontList	LabelEditorsFontList	Font	Fixed
textTypeBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textTypeForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textTypeFontList	TextEditorsFontList	Font	Fixed
textTypeShowLines	TextTypeShowLines	Boolean	True
textTypeNumWidth	TextTypeNumWidth	int	50
textTypeNumColor	TextTypeNumColor	Color	Dynamic
textTypeTabSize	TextTypeTabSize	int	4
messageHeight	MessageHeight	short	130
labelMessageBackgroundColor	LabelEditorsBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelEditorsForegroundColor	Color	Dynamic
labelMessageFontList	LabelEditorsFontList	Font	Fixed
textMessageBackgroundColor	TextEditorsBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextEditorsForegroundColor	Color	Dynamic
textMessageFontList	TextEditorsFontList	Font	Fixed

Table 4.19 lists the graphic resources you can customize for the persistent name source editor.

Each name is specified below.

- ***backgroundColor*** Editor background color
- ***foregroundColor*** Editor foreground color
- ***typeVisible*** Type window visible or not.
- ***messageVisible*** Message window visible or not
- ***editorSeparatorOn*** Editor sub-windows resized or not

- ***editorWidth*** Editor width
- ***labelInfoBackgroundColor*** Background col. of Visibility or State label
- ***labelInfoForegroundColor*** Foreground col. of Visibility or State label
- ***labelInfoFontList*** Font of Visibility or State window label
- ***infoBackgroundColor*** Background col. of Visibility or State window
- ***infoForegroundColor*** Foreground col.of Visibility or State window
- ***infoFontList*** Font of Visibility or State window
- ***infoToggleSelectColor*** Toggle for col. of Visibility or State window
- ***typeHeight*** Type window height
- ***labelTypeBackgroundColor*** Background color of Type label
- ***labelTypeForegroundColor*** Foreground color of Type label
- ***labelTypeFontList*** Font of Type label
- ***textTypeBackgroundColor*** Background color of Type text
- ***textTypeForegroundColor*** Foreground color of Type text
- ***textTypeFontList*** Font of Type text
- ***textTypeShowLines*** Show the line numbers in window
- ***textTypeNumWidth*** Line number column width
- ***textTypeNumColor*** Line number color
- ***textTypeTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The persistent name source editor name is **nameeditor**, e.g.

```
O2Tools.pnameeditor.onameeditor.nameeditor. messageBackgroundColor: Midnight blue
O2Tools*nameeditor.messageVisible: True
```

To customize the source editor presentation and object mask, refer to Section 4.3.

4.17 Customizing the Version editor

This section details the graphic resources for the Version source editor described in Section 3.2.

Table 4.20

Version editor graphic resources

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
browserSeparatorOn	BrowserSeparatorOn	Boolean	True
browserWidth	BrowserWidth	short	290
bodyVisible	BodyVisible	Boolean	True
messageVisible	MessageVisible	Boolean	True
listsHeight	ListsHeight	short	150
labelVersionsBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelVersionsForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelVersionsFontList	LabelBrowsersFontList	Font	Fixed
listVersionsBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
listVersionsForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
listVersionsFontList	ListBrowsersFontList	Font	Fixed
messageHeight	MessageHeight	short	160
labelBodyBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelBodyForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelBodyFontList	LabelBrowsersFontList	Font	Fixed
textBodyBackgroundColor	ListBrowsersBackgroundColor	Color	Dynamic
textBodyForegroundColor	ListBrowsersForegroundColor	Color	Dynamic
textBodyFontList	ListBrowsersFontList	Font	Fixed
textBodyShowLines	TextBodyShowLines	Boolean	True
textBodyNumWidth	TextBodyNumWidth	int	50
textBodyNumColor	TextBodyNumColor	Color	Dynamic
textBodyTabSize	TextBodyTabSize	int	4
messageHeight	MessageHeight	short	160
labelMessageBackgroundColor	LabelBrowsersBackgroundColor	Color	Dynamic
labelMessageForegroundColor	LabelBrowsersForegroundColor	Color	Dynamic
labelMessageFontList	LabelBrowsersFontList	Font	Fixed
textMessageBackgroundColor	TextBrowsersBackgroundColor	Color	Dynamic
textMessageForegroundColor	TextBrowsersForegroundColor	Color	Dynamic
textMessageFontList	TextBrowsersFontList	Font	Fixed

Table 4.20 lists the graphic resources you can modify to customize the Version editor.

Each name is specified below.

- ***backgroundColor*** Browser background color
- ***foregroundColor*** Browser foreground color
- ***browserSeparatorOn*** Browser sub-windows resized or not
- ***browserWidth*** Browser width
- ***bodyVisible*** Body window visible or not.

- ***messageVisible*** Message window visible or not
- ***listsHeight*** Height of browser lists
- ***labelVersionsBackgroundColor*** Background color of Version list label
- ***labelVersionsForegroundColor*** Foreground color of Version list label
- ***labelVersionsFontList*** Font of Version list label
- ***listVersionsBackgroundColor*** Background color of Version list
- ***listVersionsForegroerundColor*** Foreground color of Version list
- ***listVersionsFontList*** Font of Version list
- ***bodyHeight*** Body window height
- ***labelBodyBackgroundColor*** Background color of Body label
- ***labelBodyForegroundColor*** Foreground color of Body label
- ***labelBodyFontList*** Font of Body label
- ***textBodyBackgroundColor*** Background color of Body text
- ***textBodyForegroundColor*** Foreground color of Body text
- ***textBodyFontList*** Font of Body text
- ***textBodyShowLines*** Show line numbers in the Body
- ***textBodyNumWidth*** Line number column width
- ***textBodyNumColor*** Line number color
- ***textBodyTabSize*** Tabulation size
- ***messageHeight*** Message window height
- ***labelMessageBackgroundColor*** Background color of Message label
- ***labelMessageForegroundColor*** Foreground color of Message label
- ***labelMessageFontList*** Font of Message label
- ***textMessageBackgroundColor*** Background color of Message text
- ***textMessageForegroundColor*** Foreground color of Message text
- ***textMessageFontList*** Font of Message text

The Version editor name is **`versionbrowser`**.e.g.

```
O2Tools.pversionbrowser.oversionbrowser. versionbrowser.messageVisible: True
O2Tools*labelBody.ForegroundColor: Yellow
```

To customize the source editor presentation and object mask, refer to Section 4.3.

4.18 Customizing the O_2Shell editor

Table 4.21 lists the graphic resources you can modify to customize the Shell editor described in Section 3.12.

Table 4.21 **O_2Shell graphic resources**

Name	Class	Type	Default
backgroundColor	BackgroundColor	Color	Dynamic
foregroundColor	ForegroundColor	Color	Dynamic
shellSeparatorOn	ShellSeparatorOn	Boolean	True

Table 4.21

O_2Shell graphic resources

Name	Class	Type	Default
shellWidth	ShellWidth	short	500
inputHeight	InputHeight	short	200
labelInputBackgroundColor	LabelShellBackgroundColor	Color	Dynamic
labelInputForegroundColor	LabelShellShellForegroundColor	Color	Dynamic
labelInputFontList	LabelShellFontList	Font	Fixed
textInputBackgroundColor	TextShellBackgroundColor	Color	Dynamic
textInputForegroundColor	TextShellForegroundColor	Color	Dynamic
textInputFontList	TextShellFontList	Font	Fixed
outputHeight	OutputHeight	short	400
labelOutputBackgroundColor	LabelShellBackgroundColor	Color	Dynamic
labelOutputForegroundColor	LabelShellShellForegroundColor	Color	Dynamic
labelOutputFontList	LabelShellFontList	Font	Fixed
textOutputBackgroundColor	TextShellBackgroundColor	Color	Dynamic
textOutputForegroundColor	TextShellForegroundColor	Color	Dynamic
textOutputFontList	TextShellFontList	Font	Fixed

Each name is specified below.

- ***backgroundColor*** O_2Shell background color
- ***foregroundColor*** O_2Shell foreground color
- ***shellSeparatorOn*** O_2Shell sub-windows resized or not
- ***shellWidth*** O_2Shell width
- ***inputHeight*** Height of Input window
- ***labelInputBackgroundColor*** Background color of Input label
- ***labelInputForegroundColor*** Foreground color of Input label
- ***labelInputFontList*** Font of Input window label
- ***textnputBackgroundColor*** Background color of Input text
- ***textInputForegroerundColor*** Foreground color of Input text
- ***textInputFontList*** Font of Input text
- ***outputHeight*** Output window height
- ***labelOutputBackgroundColor*** Background color of Output label
- ***labelOutputForegroundColor*** Foreground color of Output label
- ***labelOutputFontList*** Font of Output label
- ***textOutputBackgroundColor*** Background color of Output text
- ***textOutputForegroundColor*** Foreground color of Output text
- ***textOutputFontList*** Font of Output text

The O_2Shell name is **shell**.e.g.

```
O2Tools.pshell.oshell.shell.shellSeparatorOn: True
O2Tools*labelInput.ForegroundColor: Yellow
```

4.19 Customizing the dialog boxes

Dialog boxes are components of the O_2Kit programming tool-box. Refer to the chapter O_2Kit programmer's tool-box in the O_2 User manual.

Prompt dialog boxes

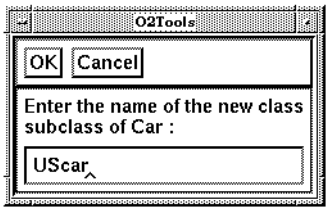


Figure 4.190: Dialog box to enter a class name

The name of the dialog box presentation is **`dialog`**. The name of the dialog box is **`prompt`**.

For example

```
O2Tools.dialog.prompt.columns: 20
```

Question dialog boxes

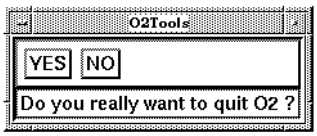


Figure 4.191: Dialog box to confirm the source save

The name of the dialog box presentation is **`dialog`**. The name of the dialog box is **`question`**.

For example

```
O2Tools.dialog.question.fontlist: 9x15
```

Other dialog boxes

- ***Renaming of inherited properties***

  The dialog box is divided up into three parts:

  - two toggle buttons to select whether the property type is an attribute or a method.
  - three prompts to give the class from which the property is inherited, the previous name and the new name.
  - a button to rename more than one property

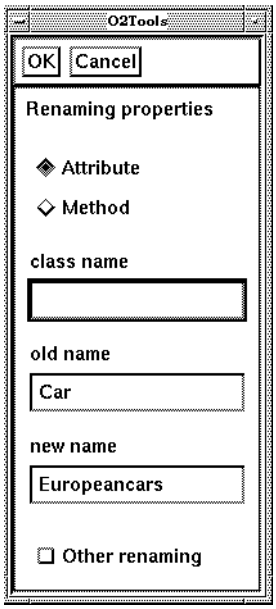


Figure 4.192: Dialog box to rename inherited properties

The name of the dialog box presentation is **`dialog`**. The name of the dialog box is **`renaming`**.

The radio box including the two toggle buttons is called **`property`**, the toggle button “attribute” is called **`attribute`**, the toggle button “method” **`method`**. For example,

```
O2Tools.dialog.renaming.property.attribute.fontlist: 9x15
```

The name of each prompt is **`prompt.`** For example,

```
O2Tools.dialog.renaming.prompt.backgroundColor: Yellow
```

The name of the button “other renaming” is **`other_renaming`**:

```
O2Tools.dialog.renaming.other_renaming.backgroundColor: Yellow
```

- ***Commit***

  This dialog box is made up of two toggle buttons to select a commit or validate action.

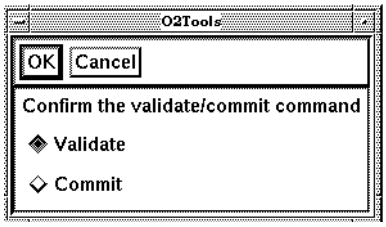


Figure 4.193: Dialog box to commit a transaction

The name of the dialog box presentation is **`dialog`.**

The radio box including the two toggle buttons is called **`commit_choice`**, the toggle button “commit” is called **`commit`**, and the toggle button “validate” **`validate`**. For example

```
O2Tools.dialog.commit_choice.commit.fontList: 9x15
```

- ***Commit from the Shell editor***

  This dialog box is made up of three toggle buttons.

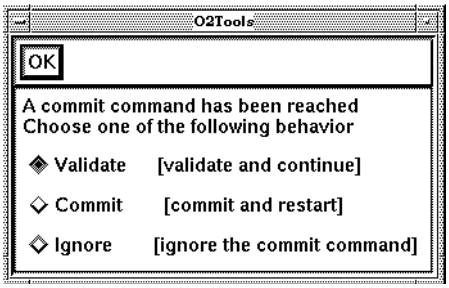


Figure 4.194: Dialog box used when a commit occurs through the shell

The name of the dialog box presentation is **`dialog`.**

The radio box including the two toggle buttons is called **`commit_choice`**, the toggle button “commit” is called **`commit`**, the toggle button “validate” **`validate`**, and the toggle button “ignore” **`ignore`.**For example

```
O2Tools.dialog.commit_choice*fontList: 9x15
```

- ***Help***

  This dialog box is made up of two toggle buttons to select alphanumeric or graphic help.

Figure 4.195: Dialog box used for help

The name of the dialog box presentation is **`dialog.`**

The radio box including the two toggle buttons is called **`help_choice`**, the toggle button “graphical environment” is called **`graphical`**, and the toggle button “command language” **`alphanumeric`**. For example

```
O2Tools.dialog.help_choice.graphic.fontList: 9x15
```

- ***Help keywords***

  This dialog box is made up of a list of all the available keywords.

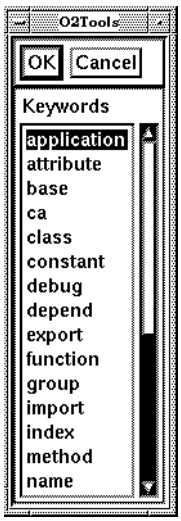


Figure 4.196: Help keywords

The name of the dialog box presentation is **`dialog.`**

The name of the dialog box is **`keywords`**, the name of the selection list **`keywor_list`**. For example

O2Tools.dialog.keywords.keyword_list.fontList: 9x15

- ***Help text***

  This dialog box is contains textual information.

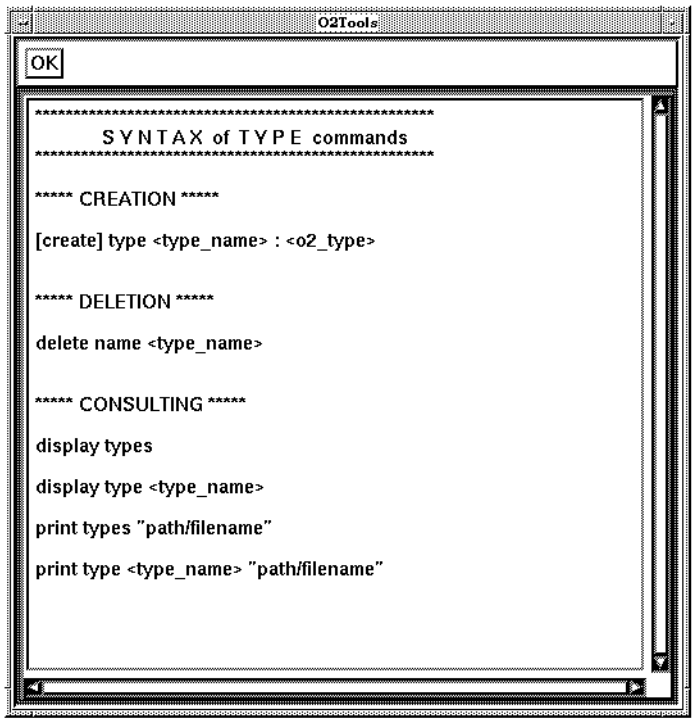


Figure 4.197: Information

The name of the presentation is **ptexthelp**,

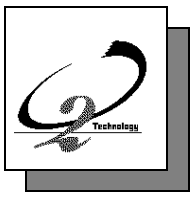
The name of the object mask of the text window is **otexthelp**.

The name of the window is **texthelp**.

For example

O2Tools.ptexthelp.otexthelp.texthelp.fontList: 9x15

4 Customizing O2 Tools



INDEX

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X