Information in this document is subject to change without notice and should not be construed as a commitment by O2 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.


All rights reserved. No part of this publication can be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopy without prior written permission of O2 Technology.

O2, O2Engine API, O2C, O2DBAccess, O2Engine, O2Graph, O2Kit, O2Look, O2Store, O2Tools, and O2Web are registered trademarks of O2 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 O₂Tools, a complete graphical programming environment for the design and development of O₂ applications. It describes the browsers and editors available, as well as how to customize O₂Tools screens.

Other documents available are outlined, click below.

See O₂ Documentation set.
TABLE OF CONTENTS

This manual is divided into the following chapters:

• 1 - Introduction
• 2 - Browsers
• 3 - Programming Environment
• 4 - Customizing O₂Tools
# Introduction

## 1.1 System Overview

## 1.2 Features
- Integrated tools
- Increased productivity
- Software requirements

## 1.3 User Manual overview

## 1.4 Getting Started

## 1.5 Other operations
- Help
- Lock
- Commit
- Abort
- Alpha

## 1.6 To exit O2

## 1.7 Multiuser Development

# Browsers

## 2.1 Introducing the O2 Tools browser

## 2.2 The Schema Browser

## 2.3 The Class Browser

## 2.4 The Application Browser
### TABLE OF CONTENTS

Manipulating Application variables .................................................. 68

2.5 The Function Browser .................................................................. 72
  Manipulating Functions ................................................................ 73

2.6 The Persistent Type Browser ..................................................... 78
  Manipulating Persistent types ...................................................... 79

2.7 The Persistent Name Browser .................................................. 82
  Manipulating Persistent names .................................................... 83

3 Programming Environment ......................................................... 89

3.1 Source editors - overview ....................................................... 90
  Common features ........................................................................ 90
  Common menus .......................................................................... 92
  Common source menu items ....................................................... 93
  Drag and drop display facility ..................................................... 96
  Current Source Editor State ....................................................... 97
  Messages .................................................................................. 98

3.2 Body - overview ....................................................................... 99
  Body State ................................................................................. 100
  Common menu items .................................................................. 102
  Body compilation error messages ............................................. 102
  Body Compilation options ......................................................... 103
  Versions of the source body ...................................................... 105
  Printing a body ........................................................................ 107
  Opening a body ........................................................................ 108
  Saving a body ........................................................................... 108
  Text Editing ............................................................................... 109

3.3 Signature - overview .............................................................. 109
  Compile ..................................................................................... 109
  Print ........................................................................................ 109
  Text Editing ............................................................................... 110

3.4 Manipulating text .................................................................... 110

3.5 The Class Source Editor ......................................................... 114
  Editing and compiling a class .................................................... 115
  Compiling a class with renaming .............................................. 116
## 3.6 The Method Source Editor

- Editing and Compiling a method ........................................... 119
- Testing a method ................................................................. 119
- Testing a method on a subclass .......................................... 121
- Entering Documentation .................................................... 121
- Edit Class ............................................................................ 121
- Printing a method ............................................................... 121
- Saving a method .................................................................. 122
- Editing and Compiling a method signature ......................... 122
- Printing a method signature ................................................. 123
- Text manipulation commands ............................................. 123
- Editing and Compiling a method body ............................... 123
- Compilation options ........................................................... 124
- Versions of the method body .............................................. 124
- Printing a method body ...................................................... 124
- Opening a method body ..................................................... 125
- Saving a method body ....................................................... 125
- Text manipulation commands ............................................. 125

## 3.7 Program Source Editor

- Editing and Compiling a Program ........................................ 127
- Entering Documentation .................................................... 127
- Printing a program ............................................................. 128
- Saving a program ............................................................... 128
- Editing and Compiling a program signature ....................... 128
- Printing a program signature .............................................. 129
- Text manipulation commands ............................................. 129
- Editing and Compiling a program body ............................ 130
- Compilation options ........................................................... 131
- Versions of program body .................................................. 131
- Printing a program body .................................................... 131
- Opening a program body ................................................... 131
- Saving a program body ..................................................... 131
- Text manipulation commands ............................................. 132

## 3.8 Application Variable Source Editor

- 132
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Editing and Compiling an Application Variable</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>Entering Documentation</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>Printing an Application Variable</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>Saving an Application Variable</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>Text manipulation commands</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>3.9 The Function Source Editor</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>Editing and Compiling a Function</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Testing a function</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Entering Documentation</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Printing a Function</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Saving a Function</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Editing and Compiling a function signature</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>Printing a Function signature</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>Text manipulation commands</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>Editing and Compiling a function body</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>Compilation options</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Versions of function body</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Printing a function body</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Opening a function body</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Saving a function body</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Text manipulation commands</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>3.10 The Persistent Type Source Editor</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>Editing and Compiling a Persistent Type</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Entering Documentation</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Printing a Persistent type</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Saving a Persistent type</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>Text manipulation commands</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>3.11 The Persistent Name Source Editor</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>Editing and Compiling a Persistent Name</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>Entering Documentation</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>Printing a Persistent name</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>Saving a Persistent name</td>
<td>145</td>
</tr>
<tr>
<td></td>
<td>Text manipulation commands</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>3.12 The O2Shell Editor</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>The Edition menu</td>
<td>148</td>
</tr>
<tr>
<td></td>
<td>The File menu</td>
<td>148</td>
</tr>
</tbody>
</table>
The Window menu................................................................. 149
Running an alphanumeric command.................................... 150
Running an O2 query ......................................................... 152

### 3.13 Global options .......................................................... 154

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Customizing O2 Tools</td>
<td>159</td>
</tr>
<tr>
<td>4.1 Preconfiguring O2 Tools</td>
<td>160</td>
</tr>
<tr>
<td>4.2 Customizing the Dashboard</td>
<td>162</td>
</tr>
<tr>
<td>4.3 Customizing presentations and object masks</td>
<td>164</td>
</tr>
<tr>
<td>4.4 Customizing the schema browser</td>
<td>166</td>
</tr>
<tr>
<td>4.5 Customizing the application browser</td>
<td>168</td>
</tr>
<tr>
<td>4.6 Customizing the class browser</td>
<td>170</td>
</tr>
<tr>
<td>4.7 Customizing the function browser</td>
<td>173</td>
</tr>
<tr>
<td>4.8 Customizing the persistent type browser</td>
<td>175</td>
</tr>
<tr>
<td>4.9 Customizing the persistent name browser</td>
<td>177</td>
</tr>
<tr>
<td>4.10 Customizing the class source editor</td>
<td>179</td>
</tr>
<tr>
<td>4.11 Customizing the method source editor</td>
<td>181</td>
</tr>
<tr>
<td>4.12 Customizing the program source editor</td>
<td>183</td>
</tr>
<tr>
<td>4.13 Customizing the variable source editor</td>
<td>185</td>
</tr>
<tr>
<td>4.14 Customizing the function source editor</td>
<td>187</td>
</tr>
<tr>
<td>4.15 Customizing persistent type source editor</td>
<td>189</td>
</tr>
<tr>
<td>4.16 Customizing persistent name source editor</td>
<td>191</td>
</tr>
<tr>
<td>4.17 Customizing the Version editor</td>
<td>193</td>
</tr>
<tr>
<td>4.18 Customizing the O2Shell editor</td>
<td>195</td>
</tr>
<tr>
<td>4.19 Customizing the dialog boxes</td>
<td>197</td>
</tr>
<tr>
<td>Prompt dialog boxes</td>
<td>197</td>
</tr>
<tr>
<td>Question dialog boxes</td>
<td>197</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

Other dialog boxes.............................................................................. 198

INDEX ....................................................................................................... 203
Introduction

Congratulations! You are now a user of O₂Tools - the complete graphical programming environment for the design and development of object-oriented database applications.

O₂Tools 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:

- System Overview
- Features
- User Manual overview
- Getting Started
- Other operations
- To exit O2
- Multiuser Development
1.1 System Overview

The system architecture of O₂ is illustrated in Figure 1.1.

The O₂ 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, O₂ development tools and any standard development tool. Numerous External Interfaces are provided. All encompassing, O₂ is a versatile, portable, distributed, high-performance dynamic object-oriented database system.

**Database Engine:**

- **O₂Store**
  The database management system provides low level facilities, through O₂Store API, to access and manage a database: disk volumes, files, records, indices and transactions.

- **O₂Engine**
  The object database engine provides direct control of schemas, classes, objects and transactions, through O₂Engine API. It provides full text indexing and search capabilities with O₂Search and spatial indexing and retrieval capabilities with O₂Spatial. It includes a Notification manager for informing other clients connected to the same O₂ 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.
System Overview

Programming Languages:

O₂ objects may be created and managed using the following programming languages, utilizing all the features available with O₂ (persistence, collection management, transaction management, OQL queries, etc.)

- C      O₂ functions can be invoked by C programs.
- C++    ODMG compliant C++ binding.
- Java   ODMG compliant Java binding.
- O₂C    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 O₂ objects and methods.

O₂ Development Tools:

- O₂Graph   Create, modify and edit any type of object graph.
- O₂Look    Design and develop graphical user interfaces, provides interactive manipulation of complex and multimedia objects.
- O₂Kit     Library of predefined classes and methods for faster development of user applications.
- O₂Tools   Complete graphical programming environment to design and develop O₂ database applications.

Standard Development Tools:

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

External Interfaces:

- O₂Corba  Create an O₂/Orbix server to access an O₂ database with CORBA.
- O₂DBAccess  Connect O₂ applications to relational databases on remote hosts and invoke SQL statements.
- O₂ODBC   Connect remote ODBC client applications to O₂ databases.
- O₂Web    Create an O₂ World Wide Web server to access an O₂ database through the internet network.
1.2 Features

Application programming in O₂ consists of defining a schema and writing, editing, compiling and debugging methods and programs.

O₂Tools 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 O₂C and using O₂Look as its user interface, O₂Tools is itself an O₂ application and can be regarded as the O₂C graphical programming environment.

O₂Tools is aimed at two different types of user

- the software developer who uses O₂Tools to browse, edit and query the database and schema as well as edit, test and debug methods and programs.

- the end-user who uses O₂Tools to browse, query and edit the database and schema.

Integrated tools

O₂Tools 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 O₂Shell, to run alphanumeric commands and O₂ 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.
Features

Increased productivity

O₂Tools greatly reduces the programming design and development cycle by allowing incrementally designed schemas, running partially written code, incremental compiling and dynamic loading of O₂C code and interactive testing of methods and programs.
**Software requirements**

To run O2Tools the following software environment is required

- X windows - release X11 or later.
- O2Engine
- O2C
- O2Look
- 4.5 Mb RAM.

### 1.3 User Manual overview

This user manual is divided into the following chapters:

- **Chapter 1 - Introduction**

  This is an introductory chapter that outlines the O2 system and the O2Tools product architecture and details how to launch O2Tools 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 O2Tools user and describes how to use all the various browsers available with O2Tools 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 O2Tools.

  It describes the O2Tools Programming environment: editing and compiling sources of classes, methods, programs, application variables, functions, persistent types and persistent names. Also detailed are the O2Shell editor, dependency management, aborting and validating a transaction.

- **Chapter 4 - How to customize O2Tools.**
Getting Started

All the displays of O2Tools 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 O2Tools User Manual describes a graphical tool for developing O2 database applications. You should read this manual in conjunction with the O2C Reference Manual which provides the semantics for the commands presented in this manual. As well, the O2C Beginners Guide provides an overall presentation of object-oriented programming, the O2 database system and step by step examples of how to develop database applications with O2.

Important

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

1.4 Getting Started

To launch O2Tools type:

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

The O2HOME environment variable is mandatory and must be set before launching O2Tools (or any O2 program). This variable contains the name of the O2 installation directory.

The options recognized by O2Tools are the following:

- `--version` : Display the O2 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 O2 system to use.
- `--server <server_name>` : Indicates the name of the host server to use.
Introduction

-libs <lib1:lib2:...:libn> : Indicates the dynamic libraries used to execute O2 code.

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

Options to O2Tools can also be given in configuration files. These files are named .o2rc and are located in the O2 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 system that has a specific option value.

The different entries of the configuration file for O2Tools are:

 system-name = sysname

Defines a system name used by default when running O2Tools.

 [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 O2Tools for alternate swap files. If not defined, the swap subdirectory of the O2 installation directory is used.

 [sysname.]swapsize = size

size is a positive decimal integer representing the triggering size in kilobytes beyond which O2 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 O2C 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 O₂C 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 O₂Tools to display warnings and messages.

[sysname.]cachesize = size

Specifies the size of the database buffer for O₂Tools.

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 O₂Tools 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"
Introduction

(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 O2Tools command line override values given with O2OPTIONS variable and .o2rc files.

O2Tools starts up and displays the dashboard illustrated in Figure 1.2 below.

![Figure 1.2: O2Tools dashboard](image)

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 O2Shell button displays the full page graphical editor to carry out alphanumeric commands and O2 queries.

The O2Debug button triggers the debugger. (The graphical debugger is not implemented at the time of printing).

The O2Tools button displays a dialog box shown in Figure 1.3 with which you can save the current configuration of your O2Tools sessions into a configuration file called o2toolsrc or set the global options of O2Tools.

![Figure 1.3: O2Tools dialog box](image)
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 O₂Tools before we strongly recommend that you change the level to Novice until you are completely sure of O₂Tools. All dialog boxes are described in this manual.

At the beginning of each O₂Tools session only the Schema Browser and the O₂Shell 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 O₂Tools via the Dashboard: Help, Lock, Commit, Abort and Alpha.

![Figure 1.5: Other Dashboard buttons](image_url)
Other operations

Help

To obtain information about O2Tools, click on the Help button. The dialog box shown in (A) Figure 1.6 appears.

![Figure 1.6: Help dialog box](image)

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

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

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, O₂ Tools is frozen and can no longer be used. To unfreeze O₂ Tools, simply click on the Lock button once again and enter your Unix password in the dialog box that appears. You can use O₂ Tools once again.
Other operations

Fig. 1.8: Lock password

Commit

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

Fig. 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.

Fig. 1.10: Abort dialog box
Note

When you click on Commit or Abort, all the other windows on display disappear while the O₂ restart mechanism is triggered.

Alpha

When you click on Alpha, a dialog box appears in which you confirm that you want to leave O₂ Tools and return to the alphanumeric top-level prompt.

![Alpha dialog box](image)

Figure 1.11: Alpha dialog box

To return to O₂ Tools type the command `toolsgraphic`.

1.6 To exit O₂

To exit O₂, click on the Quit button (A) of the dashboard.

![Quit button](image)

Figure 1.12: Quit button

A dialog box then appears asking for confirmation.
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`, O2 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
This chapter introduces the browsers available with O2 Tools 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 O2 Tools 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.
2.1 Introducing the O₂ Tools browser

You access the browsers of O₂ Tools via the dashboard. When the O₂ Tools dashboard is displayed at the beginning of a session only the Schemas button can be selected (unless you have preset the O₂ Tools configuration using the O₂ Tools button).

Important

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

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
Introducing the O2 Tools browser

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.

  ![Window menu](image1)

  *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.

  ![Options menu](image2)

  *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 Close is 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 O2Tools 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”](image)

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](image)
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.
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](image)

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](image)

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.

![Delete confirmation](image)

*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.
• **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 O2 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](image)

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](image)
The Schema Browser

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.

![List of available volumes](image)

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:

![Rename a base](image)

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.

![Delete base confirmation]

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.

![Class Browser]

Figure 2.33: Classes button

To display the Class Browser, click on the Classes button (A) on the O₂Tools 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.
The Class Browser

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
## The Class Browser

**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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Classes</th>
<th>Methods</th>
<th>Names</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create...</td>
<td>Create and Compile...</td>
<td>Create Inherits...</td>
<td>Rename...</td>
<td></td>
</tr>
<tr>
<td>Rename...</td>
<td>Delete Inherits</td>
<td>Delete</td>
<td>Delete Hierarchy</td>
<td></td>
</tr>
<tr>
<td>Confirm All</td>
<td>Save All...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*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](image)

• **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.
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**

  ![Create inherits dialog box](image)
You can create an inheritance link in several ways.

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

  ![Create Inheritance dialog box](image)

- If you have selected a class in the Class list and you select Create Inheritance, 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 sub-class is FRcar and the super class Car.

  ![Interactive graphs to create inheritance links](image)

- **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.
The Class Browser

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](image)

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](image)
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.

- **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**
The Class Browser

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

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.

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.
**Browsers**

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.

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).
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.
• **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](image)

• **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](image)
The Class Browser

- **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](image)

  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](image)

  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](image)
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](image)

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.
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.
The Application Browser

- Creating an application

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

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.
Browsers

---

**Note**

When the application is running O₂Tools 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](image)

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.
The Application Browser

Figure 2.67: Print dialog box

- **Saving an application**

  To save a particular 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.

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.
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.
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.

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.
The Application Browser

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.

- 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](image)

**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](image)

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

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

![Figure 2.80: Uncompiled Variable](image)

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](image)

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.
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.
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.

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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Function</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rename</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save All</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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](image-url)
• Creating a function

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

![Create function dialog box](image)

**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.

![Uncompiled Function](image)

**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.
The Function Browser

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.

![Persistent Type Browser](image1)

**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.

![Persistent Type Browser Components](image2)

**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.
The Persistent Type Browser

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](image)

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](image)

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.
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](image)

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](image)

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.

![Persistent Name Browser](image)

*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.
The Persistent Name Browser

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).
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.
The Persistent Name Browser

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](image)

• **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:
• 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.

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.

Warning!

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

- **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](image)

  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](image)

  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
This chapter is aimed at the experienced O₂ programmer. You are accustomed to using O₂ and are familiar with the vocabulary used in the documentation. If this is not the case, you should refer to the O₂C Reference Manual at all times.

This chapter on the O₂ Tools programming environment is divided into the following sections:

- Source editors - overview
- Body - overview
- Signature - overview
- Manipulating text
- The Class Source Editor
- The Method Source Editor
- Program Source Editor
- Application Variable Source Editor
- The Function Source Editor
- The Persistent Type Source Editor
- The Persistent Name Source Editor
- The O2Shell Editor
- Global options
3.1 Source editors - overview

In the O₂Tools 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
Source editors - overview

All source editors 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.

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 O₂Tools source editor.

- **Window menu**

The Window menu has two entries as in Figure 3.112.

```plaintext
Window
Close  Alt-C
Store  Alt-W
```

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 any modification 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.

  ![Options menu](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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Source_Name</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt-X</td>
<td></td>
</tr>
<tr>
<td>Edit Documentation...</td>
<td>Alt-E</td>
<td></td>
</tr>
<tr>
<td>Print...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Common items](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](image)

• **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.
Source editors - overview

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.

When you click on Open, Append or Write As the dialog box shown in Figure 3.118 appears.
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, O₂Tools has a Drag and drop facility that enables you to display different sources using the same source editor.
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](image)

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](image)
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](image)

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 O₂ Shell. See Section 3.12.

For more information about dependencies see the Section on O₂C Compiler options in the O₂C Reference Manual.
Common menu items

All Body menus are as in Figure 3.124.

<table>
<thead>
<tr>
<th>Window</th>
<th>Program</th>
<th>Signature</th>
<th>Body</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Compile</td>
<td>Alt- B</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Options...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Versions</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Print</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Write As...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Toggle line number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Search...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Replace...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Go to line...</td>
<td></td>
</tr>
</tbody>
</table>

*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), O₂Tools 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).
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:
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 -l 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 O2C 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 O2 Tools 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.
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.

```
Create
Delete
```

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).

---

**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

**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](image)

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](image)

**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.

![Print dialog box](image)

**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:

<table>
<thead>
<tr>
<th>Window</th>
<th>Source</th>
<th>Signature</th>
<th>Body</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Compile</td>
<td>Alt-S</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Print</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Toggle line number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Search...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Replace...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go to line...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*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.
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 O₂Tools which greatly simplifies entering code.

![Text Items](image-url)
Manipulating text

- **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.

  ![](image)

  **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.
# Manipulating text

## Table 1.

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

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.
The Class Source Editor

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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Class</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt-X</td>
<td></td>
</tr>
<tr>
<td>Compile with Renaming...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Documentation...</td>
<td>Alt-E</td>
<td></td>
</tr>
<tr>
<td>Print...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toggle line number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replace...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Go to line...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 (c) 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:

![Compile with renaming dialog box](image)

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.
The Method Source Editor

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.
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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Method</th>
<th>Signature</th>
<th>Body</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt-X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test...</td>
<td>Alt-T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test on Subclass...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Documentation...Alt-E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Class...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*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.
The Method Source Editor

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.

![Subclasses window](image)

*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 alphanumerics `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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Method</th>
<th>Signature</th>
<th>Body</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt-S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toggle line number</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replace...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Go to line...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
The Method Source Editor

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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Method</th>
<th>Signature</th>
<th>Body</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt- B</td>
<td>Options...</td>
<td>Versions</td>
<td>Print</td>
</tr>
<tr>
<td>Open...</td>
<td>Write As...</td>
<td>Toggle line number</td>
<td>Search...</td>
<td>Replace...</td>
</tr>
<tr>
<td>Go to line...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*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.

![Program source program menu](image)

**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.
3 Programming Environment

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.

```
Window Program | Signature | Body Options
---------------|-----------|---------------
                  Compile | Alt-S     
                  Print

Toggle line number
Search...
Replace...
Go to line...
```

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.
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
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 {}.

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.

![Application Variable Source Editor Diagram]

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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Variable</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt-X</td>
<td></td>
</tr>
<tr>
<td>Edit Documentation</td>
<td>Alt-E</td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toggle line number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replace...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Go to line...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
3 Programming Environment

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.
The Function Source Editor

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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Function</th>
<th>Signature</th>
<th>Body</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt-X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test...</td>
<td>Alt-T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Documentation...</td>
<td>Alt-E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
The Function Source Editor

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.
**The Function Source Editor**

## Text manipulation commands

All the text manipulation commands are described in Section 3.4.

The Function body menu is shown in Figure 3.163.

<table>
<thead>
<tr>
<th>Window</th>
<th>Function</th>
<th>Signature</th>
<th>Body</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Compile Alt-B</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Options...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Versions</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Print</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Write As...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Toggle line number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Search...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Replace...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Go to line...</td>
<td></td>
</tr>
</tbody>
</table>

*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.

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.

<table>
<thead>
<tr>
<th>Window</th>
<th>Type</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Compile</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alt-X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Edit Documentation... Alt-E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Print...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Save...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Toggle line number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Search...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Replace...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Go to line...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Save...</td>
</tr>
</tbody>
</table>

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"`
The Persistent Name Source Editor

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.
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 Persistent Name Source Editor

The Name menu is shown in Figure 3.168.

<table>
<thead>
<tr>
<th>Window</th>
<th>Name</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile</td>
<td>Alt-X</td>
<td></td>
</tr>
<tr>
<td>Edit Documentation...</td>
<td>Alt-E</td>
<td></td>
</tr>
<tr>
<td>Print...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toggle line number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replace...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Go to line...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*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:

```plaintext
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 O₂Shell Editor

You use O₂ alphanumerically or carry out O₂ queries with O₂Shell.

To open the O₂Shell Editor, click on the O₂Shell button (A) on the O₂Tools dashboard.

The O₂Shell Editor now appears as shown in Figure 3.170.
The O2Shell Editor

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

**Important**

Two stars - ** - before the O2Shell 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 O2Shell editor](image)

Use the input subwindow (c) to enter an alphanumeric O2 command or an O2 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.
• 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

<table>
<thead>
<tr>
<th>Window</th>
<th>Edition</th>
<th>Execution</th>
<th>Query</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear Input</td>
<td>Alt-I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clear Output</td>
<td>Alt-O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Store</td>
<td>Alt-W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close</td>
<td>Alt-C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.175: Shell Window menu
3 Programming Environment

- **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 O2 Query.

### Running an alphanumeric command

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

```
<table>
<thead>
<tr>
<th>Window</th>
<th>Edition</th>
<th>Execution</th>
<th>Query</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Run</td>
<td></td>
<td>Alt-D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Run On Selection</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

*Figure 3.176: Shell Execution menu*

- **Run**

To run an O2 command, type into the input subwindow an O2 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.
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 O2 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 O2 commands, a dialog box appears, shown in Figure 3.178, asking you to choose the behavior of the commit: validate, commit or ignore.
Running an $O_2$ query

To query use the Query menu shown in Figure 3.179.

<table>
<thead>
<tr>
<th>Window</th>
<th>Edition</th>
<th>Execution</th>
<th>Query</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Query</td>
<td>Alt-Q</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Graphic Query</td>
<td>Alt-G</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Query On Selection</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Graphic Query On Selection</td>
<td></td>
</tr>
</tbody>
</table>

*Query*

To run an $O_2$ query, edit the input subwindow by entering 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 O₂ Tools button in the dashboard, shown in (A) in Figure 3.181 below.

![Figure 3.181: O₂ Tools button](image)

The following window then appears.

![Figure 3.182: O₂ Tools dialog box](image)

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 O₂ file selector. Enter the path name in the File Selector section of the window (A) and then on OK.
Global options

- **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: O₂Tools configuration
Global options

- **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 \textit{O}_2\textit{C Reference Manuals}.
3 Programming Environment
4 Customizing O₂ Tools

The O₂Tools user interface is built using O₂Look and can therefore be customized in the same way as O₂Look by modifying graphical resources. For more information about O₂Look resources refer to the O₂Look User Manual.

This chapter firstly explains how to preconfigure O₂Tools by creating a configuration file. It then gives the graphic resources you use to customize the following components of O₂Tools:

- 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
4.1 Preconfiguring O₂ Tools

If you know you are going to work on the same schema, base, and browsers, for several O₂ Tools 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 O₂ Tools is launched.

To create a configuration file, simply click on the O₂ Tools button on the dashboard marked (A) in Figure 4.184.
Preconfiguring O2 Tools

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: O2Tools dialog box](image)

Another dialog box then asks you to confirm your command

![Figure 4.186: Confirm file creation](image)

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](image)

**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.

![The dashboard](image)

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**

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>

**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.

```o2shell
O2Tools.ptoolsdashboard.ptoolsdashboard.foregroundColor:Midnight Blue
```

**Table 4.2**

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>

**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.

```o2shell
O2Tools.ptoolsdashboard.otoolsdashboard.otoolsdashboard.foregroundColor:Midnight Blue
```
Customizing the Dashboard

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` O2Tools 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:

```plaintext
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 O2Shell in the same way.

Table 4.4 lists the graphic resources you use to customize a browser, source editor or O2Shell 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 used 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 O2Shell editors have the following presentation names:

- `schema browser presentation`: `pschemabrowser`
- `application browser presentation`: `papplibrowser`
- `class browser presentation`: `pclasstexteditor`
- `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`: `ptypeditor`
- `persistent name source editor presentation`: `pnameeditor`
- `version editor presentation`: `pversionbrowser`
- `O2Shell presentation`: `pshell`
Customizing presentations and object masks

Table 4.5 lists the graphic resources to customize a browser, source editor or O2Shell 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
O2Tools*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 O2Shell 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`
- **O2Shell 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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>browserSeparatorOn</td>
<td>BrowserSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>browserWidth</td>
<td>BrowserWidth</td>
<td>short</td>
<td>380</td>
</tr>
<tr>
<td>listsHeight</td>
<td>ListsHeight</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelSchemasBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelSchemasForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelSchemasFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listSchemasBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listSchemasForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listSchemasFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>labelBasesBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelBasesForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelBasesFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>descriptionVisible</td>
<td>DescriptionVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>descriptionHeight</td>
<td>DescriptionHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>labelDescriptionBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textDescriptionBackgroundColor</td>
<td>TextBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

Table 4.6 lists the graphic resources used to customize the schema browser.
Customizing 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
- `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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>browserSeparatorOn</td>
<td>BrowserSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>browserWidth</td>
<td>BrowserWidth</td>
<td>short</td>
<td>550</td>
</tr>
<tr>
<td>listsHeight</td>
<td>ListsHeight</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelApplicationsBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelApplicationsForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelApplicationsFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listApplicationsBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listApplicationsForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listApplicationsFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>labelProgramsBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelProgramsForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelProgramsFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listProgramsBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listProgramsForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listProgramsFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>labelVariablesBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelVariablesForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelVariablesFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>descriptionVisible</td>
<td>DescriptionVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>descriptionHeight</td>
<td>DescriptionHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>labelDescriptionBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textDescriptionBackgroundColor</td>
<td>TextBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
Customizing the application browser

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
- **listHeight**: Height of browser list
- **labelApplicationsBackgroundColor**: Background color of Application list label
- **labelApplicationsForegroundColor**: Foreground color 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
- **listVariablesFontList**: Font 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.applibbrowser.applibbrowser.hierarchyVisible:True`

`O2Tools*labelHierarchyForegroundColor: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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>browserSeparatorOn</td>
<td>BrowserSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>browserWidth</td>
<td>BrowserWidth</td>
<td>short</td>
<td>True</td>
</tr>
<tr>
<td>listsHeight</td>
<td>ListsHeight</td>
<td>short</td>
<td>True</td>
</tr>
<tr>
<td>labelClassesBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelClassesForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelClassesFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listClassesBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listClassesForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listClassesFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>labelMethodsBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMethodsForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMethodsFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listMethodsBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listMethodsForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listMethodsFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>labelNamedObjectsBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelNamedObjectsForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelNamedObjectsFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listNamedObjectsBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listNamedObjectsForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listNamedObjectsFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>descriptionVisible</td>
<td>DescriptionVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>descriptionHeight</td>
<td>DescriptionHeight</td>
<td>short</td>
<td>True</td>
</tr>
<tr>
<td>labelDescriptionBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textDescriptionBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>hierarchyVisible</td>
<td>HierarchyVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>hierarchyHeight</td>
<td>HierarchyHeight</td>
<td>short</td>
<td>True</td>
</tr>
<tr>
<td>labelHierarchyBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelHierarchyForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelHierarchyFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>True</td>
</tr>
</tbody>
</table>
Customizing the class browser

Table 4.8

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

What each name specifies is listed below

- **backgroundColor**: Browser background color
- **foregroundColor**: Browser foreground color
- **browserSeparatorOn**: Browser subwindows resized or not
- **browserWidth**: Browser width
- **listHeight**: 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 color of Method list label
- **labelMethodsForegroundColor**: Foreground color of Method list label
- **labelMethodsFontList**: Font of Method list
- **listMethodsBackgroundColor**: Background color of Method list
- **listMethodsForegroundColor**: Foreground color of Method list
- **listMethodsFontList**: Font of Method list
- **labelNamedObjectsBackgroundColor**: Background color of Name list label
- **labelNamedObjectsForegroundColor**: Foreground color of Name list label
- **labelNamedObjectsFontList**: 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
Customizing O2 Tools

- `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
```

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>hgap</td>
<td>Gap</td>
<td>short</td>
<td>60</td>
</tr>
<tr>
<td>vgap</td>
<td>Gap</td>
<td>short</td>
<td>10</td>
</tr>
<tr>
<td>linkThickness</td>
<td>LinkThickness</td>
<td>short</td>
<td>0</td>
</tr>
<tr>
<td>linkColor</td>
<td>LinkColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>selThickness</td>
<td>SelThickness</td>
<td>short</td>
<td>4</td>
</tr>
<tr>
<td>selColor</td>
<td>SelColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>

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.
Customizing O2 Tools

Table 4.10

Function browser graphic resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BACKGROUND_COLOR</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>FOREGROUND_COLOR</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>browserSeparatorOn</td>
<td>BrowserSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>browserWidth</td>
<td>BrowserWidth</td>
<td>short</td>
<td>290</td>
</tr>
<tr>
<td>listsHeight</td>
<td>ListHeight</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelFunctionsBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelFunctionsForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelFunctionsFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listFunctionsBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listFunctionsForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listFunctionsFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>descriptionVisible</td>
<td>DescriptionVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>descriptionHeight</td>
<td>DescriptionHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>labelDescriptionBackgroundColor</td>
<td>LabelBrowsersBackgroundcolor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelDescriptionFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textDescriptionBackgroundColor</td>
<td>TextBrowsersBackgroundcolor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textDescriptionFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelBrowsersBackgroundcolor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextBrowsersBackgroundcolor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

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
Customizing the persistent type browser

- `labelDescriptionForegroundColor`: Foreground color of Description label
- `labelDescriptionFontList`: Font of Description label
- `textDescriptionBackgroundColor`: Background 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
- `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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>browserSeparatorOn</td>
<td>BrowserSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>browserWidth</td>
<td>BrowserWidth</td>
<td>short</td>
<td>290</td>
</tr>
<tr>
<td>listsHeight</td>
<td>ListsHeight</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelTypesBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelTypesForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelTypesFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listTypesBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>
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
Customizing the persistent name browser

- `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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>browserSeparatorOn</td>
<td>BrowserSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>browserWidth</td>
<td>BrowserWidth</td>
<td>short</td>
<td>290</td>
</tr>
<tr>
<td>listsHeight</td>
<td>ListsHeight</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelNamesBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelNamesForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelNamesFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>listNamesBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listNamesForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listNamesFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
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
- `listNamesForegroundColor` 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
Customizing the class source editor

- `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,

```plaintext
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>typeVisible</td>
<td>TypeVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorSeparatorOn</td>
<td>EditorSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorWidth</td>
<td>EditorWidth</td>
<td>short</td>
<td>360</td>
</tr>
<tr>
<td>labelInfoBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoBackgroundColor</td>
<td>InfoEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoForegroundColor</td>
<td>InfoEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoFontList</td>
<td>InfoEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoToggleSelectColor</td>
<td>InfoToggleSelectColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>typeHeight</td>
<td>TypeHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>labelTypeBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelTypeForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelTypeFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
### Table 4.13  Class source editor graphic resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>textTypeBackgroundColor</strong></td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeForegroundColor</strong></td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeSeparatorOn</strong></td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>textTypeShowLines</strong></td>
<td>TextTypeShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td><strong>textTypeNumWidth</strong></td>
<td>TextTypeNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td><strong>textTypeNumColor</strong></td>
<td>TextTypeNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeTabSize</strong></td>
<td>TextTypeTabSize</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td><strong>messageHeight</strong></td>
<td>MessageHeight</td>
<td>short</td>
<td>130</td>
</tr>
<tr>
<td><strong>labelMessageBackgroundColor</strong></td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>labelMessageForegroundColor</strong></td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>labelMessageFontList</strong></td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>textMessageBackgroundColor</strong></td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textMessageForegroundColor</strong></td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textMessageFontList</strong></td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>textTypeBackgroundColor</strong></td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeForegroundColor</strong></td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeFontList</strong></td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>textTypeShowLines</strong></td>
<td>TextTypeShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td><strong>textTypeNumWidth</strong></td>
<td>TextTypeNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td><strong>textTypeNumColor</strong></td>
<td>TextTypeNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeTabSize</strong></td>
<td>TextTypeTabSize</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td><strong>messageHeight</strong></td>
<td>MessageHeight</td>
<td>short</td>
<td>130</td>
</tr>
<tr>
<td><strong>labelMessageBackgroundColor</strong></td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>labelMessageForegroundColor</strong></td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>labelMessageFontList</strong></td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>textMessageBackgroundColor</strong></td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textMessageForegroundColor</strong></td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textMessageFontList</strong></td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>textTypeBackgroundColor</strong></td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeForegroundColor</strong></td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>textTypeFontList</strong></td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td><strong>textTypeShowLines</strong></td>
<td>TextTypeShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td><strong>textTypeNumWidth</strong></td>
<td>TextTypeNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td><strong>textTypeNumColor</strong></td>
<td>TextTypeNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>

*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
Customizing the method source editor

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>textSignatureShowLines</td>
<td>TextSignatureShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textSignatureNumWidth</td>
<td>TextSignatureNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>textSignatureNumColor</td>
<td>TextSignatureNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textSignatureTabSize</td>
<td>TextSignatureTabSize</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>bodyHeight</td>
<td>BodyHeight</td>
<td>short</td>
<td>400</td>
</tr>
<tr>
<td>labelBodyBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelBodyForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelBodyFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBodyBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBodyShowLines</td>
<td>TextBodyShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textBodyNumWidth</td>
<td>TextBodyNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>textBodyTabSize</td>
<td>TextBodyTabSize</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageShowLines</td>
<td>TextMessageShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textBodyBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBodyShowLines</td>
<td>TextBodyShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textBodyNumWidth</td>
<td>TextBodyNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>textBodyTabSize</td>
<td>TextBodyTabSize</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageShowLines</td>
<td>TextMessageShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textBodyBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBodyShowLines</td>
<td>TextBodyShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
</tbody>
</table>
Customizing the program source editor

- `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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>signatureVisible</td>
<td>SignatureVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>bodyVisible</td>
<td>BodyVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorSeparatorOn</td>
<td>EditorSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorWidth</td>
<td>EditorWidth</td>
<td>short</td>
<td>150</td>
</tr>
<tr>
<td>labelInfoBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoBackgroundColor</td>
<td>InfoEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoForegroundColor</td>
<td>InfoEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoFontList</td>
<td>InfoEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoToggleSelectColor</td>
<td>InfoToggleSelectColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>signatureHeight</td>
<td>SignatureHeight</td>
<td>short</td>
<td>120</td>
</tr>
</tbody>
</table>
Table 4.15 lists the graphic resources you can customize for the program source editor. Each name specification is listed below.

- **backgroundColor**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **foregroundColor**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **bodyVisible**
  - **Class**: LabelEditorsFontList
  - **Type**: Boolean
  - **Default**: True
- **signatureVisible**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **messageVisible**
  - **Class**: TextEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **editorSeparatorOn**
  - **Class**: TextEditorsFontList
  - **Type**: Boolean
  - **Default**: False
- **editorWidth**
  - **Class**: TextEditorsFontList
  - **Type**: Boolean
  - **Default**: False
- **labelInfoBackgroundColor**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelInfoForegroundColor**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelInfoFontList**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **labelSignatureBackgroundColor**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelSignatureForegroundColor**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelSignatureFontList**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **signatureHeight**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **signatureShowLines**
  - **Class**: TextEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **signatureNumWidth**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **bodyHeight**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **bodyShowLines**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **bodyNumWidth**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **messageHeight**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **messageShowLines**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **messageNumWidth**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **textSignatureBackgroundColor**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **textSignatureForegroundColor**
  - **Class**: TextEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **textSignatureFontList**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **textSignatureShowLines**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **textSignatureNumWidth**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **textSignatureTabSize**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed

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

- **backgroundColor**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **foregroundColor**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **bodyVisible**
  - **Class**: LabelEditorsFontList
  - **Type**: Boolean
  - **Default**: True
- **signatureVisible**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **messageVisible**
  - **Class**: TextEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **editorSeparatorOn**
  - **Class**: TextEditorsFontList
  - **Type**: Boolean
  - **Default**: False
- **editorWidth**
  - **Class**: TextEditorsFontList
  - **Type**: Boolean
  - **Default**: False
- **labelInfoBackgroundColor**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelInfoForegroundColor**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelInfoFontList**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **labelSignatureBackgroundColor**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelSignatureForegroundColor**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **labelSignatureFontList**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **signatureHeight**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **signatureShowLines**
  - **Class**: TextEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **signatureNumWidth**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **bodyHeight**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **bodyShowLines**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **bodyNumWidth**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **messageHeight**
  - **Class**: LabelEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **messageShowLines**
  - **Class**: LabelEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **messageNumWidth**
  - **Class**: LabelEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **textSignatureBackgroundColor**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **textSignatureForegroundColor**
  - **Class**: TextEditorsForegroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **textSignatureFontList**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **textSignatureShowLines**
  - **Class**: TextEditorsBackgroundColor
  - **Type**: Color
  - **Default**: Dynamic
- **textSignatureNumWidth**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
- **textSignatureTabSize**
  - **Class**: TextEditorsFontList
  - **Type**: Font
  - **Default**: Fixed
Customizing the variable source editor

- `labelSignatureFontList`: Font of Signature label
- `textSignatureBackgroundColor`: Background color of Signature text
- `textSignatureForegroundColor`: Foreground color of Signature text
- `textSignatureNumWidth`: Line number column width
- `textSignatureTabSize`: Tabulation size
- `textSignatureShowLines`: Show line numbers in Signature
- `textSignatureNumColor`: Line number color
- `textBodyBackgroundColor`: Background color of Body label
- `textBodyForegroundColor`: Foreground color of Body label
- `textBodyFontList`: Font of Body label
- `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.programeditor.qprogrameditor.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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>typeVisible</td>
<td>TypeVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorSeparatorOn</td>
<td>EditorSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorWidth</td>
<td>EditorWidth</td>
<td>short</td>
<td>250</td>
</tr>
<tr>
<td>labelInfoBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
</tbody>
</table>
Table 4.16 lists the graphic resources used to customize the application variable source editor.

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>typeVisible</td>
<td>LabelEditorsFontList</td>
<td>String</td>
<td>Dynamic</td>
</tr>
<tr>
<td>messageVisible</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>editorSeparatorOn</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>editorWidth</td>
<td>LabelEditorsBackgroundColor</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>labelInfoBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoForegroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoFontList</td>
<td>LabelEditorsBackgroundColor</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoForegroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoFontList</td>
<td>LabelEditorsBackgroundColor</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoToggleSelectColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>messageHeight</td>
<td>LabelEditorsBackgroundColor</td>
<td>int</td>
<td>130</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelEditorsBackgroundColor</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textForegroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textFontList</td>
<td>LabelEditorsBackgroundColor</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textTypeShowLines</td>
<td>LabelEditorsBackgroundColor</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textTypeNumWidth</td>
<td>LabelEditorsBackgroundColor</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>textTypeNumColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textTabSize</td>
<td>LabelEditorsBackgroundColor</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>messageHeight</td>
<td>LabelEditorsBackgroundColor</td>
<td>int</td>
<td>130</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelEditorsBackgroundColor</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>LabelEditorsBackgroundColor</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textTypeShowLines</td>
<td>LabelEditorsBackgroundColor</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textTypeNumWidth</td>
<td>LabelEditorsBackgroundColor</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>textTypeNumColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textTabSize</td>
<td>LabelEditorsBackgroundColor</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>typeHeight</td>
<td>TypeHeight</td>
<td>int</td>
<td>130</td>
</tr>
</tbody>
</table>

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
Customizing the function source editor

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

For example,

```plaintext
O2Tools.variableeditor.o1variableeditor.o2variableeditor.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>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>signatureVisible</td>
<td>SignatureVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>bodyVisible</td>
<td>BodyVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorSeparatorOn</td>
<td>EditorSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorWidth</td>
<td>EditorWidth</td>
<td>short</td>
<td>450</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>labelInfoBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoBackgroundColor</td>
<td>InfoEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoForegroundColor</td>
<td>InfoEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoFontList</td>
<td>InfoEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoToggleSelectColor</td>
<td>InfoToggleSelectColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>signatureHeight</td>
<td>SignatureHeight</td>
<td>Short</td>
<td>120</td>
</tr>
<tr>
<td>labelSignatureBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelSignatureForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelSignatureFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textSignatureBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textSignatureForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textSignatureFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textSignatureShowLines</td>
<td>TextSignatureShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textSignatureNumWidth</td>
<td>TextSignatureNumWidth</td>
<td>Int</td>
<td>50</td>
</tr>
<tr>
<td>textSignatureNumColor</td>
<td>TextSignatureNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textSignatureTabSize</td>
<td>TextSignatureTabSize</td>
<td>Int</td>
<td>4</td>
</tr>
<tr>
<td>bodyHeight</td>
<td>BodyHeight</td>
<td>Short</td>
<td>400</td>
</tr>
<tr>
<td>labelBodyBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelBodyForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelBodyFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBodyBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBodyShowLines</td>
<td>TextBodyShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>textBodyNumWidth</td>
<td>TextBodyNumWidth</td>
<td>Int</td>
<td>50</td>
</tr>
<tr>
<td>textBodyNumColor</td>
<td>TextBodyNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textBodyTabSize</td>
<td>TextBodyTabSize</td>
<td>Int</td>
<td>4</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>Short</td>
<td>150</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
### Customizing persistent type source editor

- **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.functioneditor.functioneditor.messageBackgroundColor: Midnight blue
O2Tools*functioneditor.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.
### Persistent type source editor resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>typeVisible</td>
<td>TypeVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorSeparatorOn</td>
<td>EditorSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>editorWidth</td>
<td>EditorWidth</td>
<td>short</td>
<td>250</td>
</tr>
<tr>
<td>labelInfoBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInfoFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoBackgroundColor</td>
<td>InfoEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoForegroundColor</td>
<td>InfoEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>infoFontList</td>
<td>InfoEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>infoToggleSelectColor</td>
<td>InfoToggleSelectColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>typeHeight</td>
<td>TypeHeight</td>
<td>short</td>
<td>130</td>
</tr>
<tr>
<td>labelTypeBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelTypeForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelTypeFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textMessageBackgroundColor</td>
<td>TextEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageForegroundColor</td>
<td>TextEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textMessageFontList</td>
<td>TextEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>130</td>
</tr>
<tr>
<td>labelMessageBackgroundColor</td>
<td>LabelEditorsBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageForegroundColor</td>
<td>LabelEditorsForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelMessageFontList</td>
<td>LabelEditorsFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>130</td>
</tr>
<tr>
<td>textNumWidth</td>
<td>TextTypeNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>textNumColor</td>
<td>TextTypeNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textTabSize</td>
<td>TextTypeTabSize</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>textTypeShowLines</td>
<td>TextTypeShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
</tbody>
</table>

**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
Customizing persistent name source editor

- `infoBackgroundColor` Background color of Visibility or State window
- `infoForegroundColor` Foreground color of Visibility or State window
- `infoFontList` Font of Visibility or State window
- `infoToggleSelectColor` Toggle color 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.messageBackgroundColor: Midnight blue
```

```
O2Toolsftypeeditor 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 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
## Customizing the Version editor

- **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 label
- **infoBackgroundColor**: Background col. of Visibility or State window label
- **infoForegroundColor**: Foreground col. of Visibility or State window label
- **infoFontList**: Font of Visibility or State window
- **infoToggleSelectColor**: Toggle for col. of Visibility or State window label
- **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**.
Customizing O2 Tools

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.
  ```

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>browserSeparatorOn</td>
<td>BrowserSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>browserWidth</td>
<td>BrowserModule</td>
<td>short</td>
<td>290</td>
</tr>
<tr>
<td>bodyVisible</td>
<td>BodyVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>messageVisible</td>
<td>MessageVisible</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>listsHeight</td>
<td>ListsHeight</td>
<td>short</td>
<td>190</td>
</tr>
<tr>
<td>listVersionsBackgroundColor</td>
<td>ListBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listVersionsForegroundColor</td>
<td>ListBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>listVersionFontList</td>
<td>ListBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>bodyBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>bodyForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>bodyFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>bodyShowLines</td>
<td>TextBodyShowLines</td>
<td>Boolean</td>
<td>True</td>
</tr>
<tr>
<td>bodyNumWidth</td>
<td>TextBodyNumWidth</td>
<td>int</td>
<td>50</td>
</tr>
<tr>
<td>bodyNumColor</td>
<td>TextBodyNumColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>bodyTabSize</td>
<td>TextBodyTabSize</td>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>messageHeight</td>
<td>MessageHeight</td>
<td>short</td>
<td>160</td>
</tr>
<tr>
<td>messageBackgroundColor</td>
<td>LabelBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>messageForegroundColor</td>
<td>LabelBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>messageFontList</td>
<td>LabelBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>messageBackgroundColor</td>
<td>TextBrowsersBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>messageForegroundColor</td>
<td>TextBrowsersForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>messageFontList</td>
<td>TextBrowsersFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>
Customizing the O2Shell editor

- 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
- listVersionsForegroundColor: 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.versionbrowser.o.versionbrowser.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 O2Shell editor

Table 4.21 lists the graphic resources you can modify to customize the Shell editor described in Section 3.12.

### Table 4.21

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>backgroundColor</td>
<td>BackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>foregroundColor</td>
<td>ForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>shellSeparatorOn</td>
<td>ShellSeparatorOn</td>
<td>Boolean</td>
<td>True</td>
</tr>
</tbody>
</table>
Customizing O2 Tools

Table 4.21

<table>
<thead>
<tr>
<th>Name</th>
<th>Class</th>
<th>Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>shellWidth</td>
<td>ShellWidth</td>
<td>short</td>
<td>500</td>
</tr>
<tr>
<td>inputHeight</td>
<td>InputHeight</td>
<td>short</td>
<td>200</td>
</tr>
<tr>
<td>labelInputBackgroundColor</td>
<td>LabelShellBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInputForegroundColor</td>
<td>LabelShellShellForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelInputFontList</td>
<td>LabelShellFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textInputBackgroundColor</td>
<td>TextShellBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textInputForegroundColor</td>
<td>TextShellForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textInputFontList</td>
<td>TextShellFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>outputHeight</td>
<td>OutputHeight</td>
<td>short</td>
<td>400</td>
</tr>
<tr>
<td>labelOutputBackgroundColor</td>
<td>LabelShellBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelOutputForegroundColor</td>
<td>LabelShellShellForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>labelOutputFontList</td>
<td>LabelShellFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
<tr>
<td>textOutputBackgroundColor</td>
<td>TextShellBackgroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textOutputForegroundColor</td>
<td>TextShellForegroundColor</td>
<td>Color</td>
<td>Dynamic</td>
</tr>
<tr>
<td>textOutputFontList</td>
<td>TextShellFontList</td>
<td>Font</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

Each name is specified below.

- **backgroundColor**  
  O2Shell background color
- **foregroundColor**  
  O2Shell foreground color
- **shellSeparatorOn**  
  O2Shell sub-windows resized or not
- **shellWidth**  
  O2Shell width
- **inputHeight**  
  Height of Input window
- **labelInputBackgroundColor**  
  Background color of Input label
- **labelInputForegroundColor**  
  Foreground color of Input label
- **labelInputFontList**  
  Font of Input window label
- **textInputBackgroundColor**  
  Background color of Input text
- **textInputForegroundColor**  
  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 O2Shell 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 O2Kit programming tool-box. Refer to the chapter O2Kit programmer's tool-box in the O2 User manual.

Prompt dialog boxes

![Prompt dialog box]

*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

```plaintext
O2Tools.dialog.prompt.columns: 20
```

Question dialog boxes

![Question dialog box]

*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

```plaintext
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

![Image of dialog box]

*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`
Customizing the dialog boxes

• **Commit**

This dialog box is made up of two toggle buttons to select a commit or validate action.

![Commit dialog box](image)

*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.

![Commit from Shell dialog box](image)

*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.ignore.fontList: 9x15
- **Help**

This dialog box is made up of two toggle buttons to select alphanumerical or graphic help.

![Figure 4.195: Dialog box used for help](image)

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](image)

The name of the dialog box presentation is `dialog`. The name of the dialog box is `keywords`, the name of the selection list `keyword_list`. For example
Customizing the dialog boxes

O2Tools.dialog.keywords.keyword_list.fontList: 9x15

- **Help text**

  This dialog box contains textual information.

  ![Image of dialog box](image)

  **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
INDEX

A

Abort 27
Accelerators 33, 93
Alpha 28
Alphanumeric mode 28
Append 95, 96, 149
Application 60–63
Create 61
Delete 63
Manipulation 60
Print 62
Rename 62
Testing 61
Application Browser 58
Application list 59
Application variable list 68
Applications menu 60
Apps button 58
Customization 168–170
Program list 64
Programs menu 64
Application variable 68–72, 132–134
Compile 133
Create 70
Delete 71
Manipulation 68
Menu 69
Rename 71
Text Editor 134
Application variable Source Editor 69,
132–133, 185
Customization 185–187
Graphic resources 185
Variable menu 133
Applications menu 60
Apps button 58
Architecture
O₂ 14
Automatic dependency mode 157

B

Base 38–41
Create on a Volume 41
Delete 42
Manipulation 38
Rename 41
Set 40
Show All 41
Bases menu 39
Body 100
Compile version 107
Delete version 107
Error messages 102
Function 139–140
Menu 105
Method 123–125
Open 108
Program 130–131
Save 108
Text 109
Body State 100
Browser 22
Application 58
Class 42
Function 72
General features 32
Persistent Name 82
Persistent Type 78
Schema 34

C

C 15, 50
C++
Interface 15
Class 42–50, 114–116
  Compile 115, 116
  Confirm 50
  Create 46
  Create and Compile 47
  Delete 49
  Description 43
  Edit 115
  Manipulation 43
  Other rename 116
  Rename 48, 116
  Save 50
Class Browser 42
  Class list 42
  Classes button 42
  Classes menu 45
  Customization 170–173
  Graphic resources 170
  Hierarchy pop-up menu 45
  Method list 50
  Methods menu 51
  Name list 54
  Names menu 55
Class Hierarchy 45
  Customization 173
  Delete Hierarchy 50
  Delete inherits 49
  Pop-up menu 45
Class Source Editor 46, 114–117, 179–181
  Class menu 115
  Customization 179–181
  Graphic resources 179
  Text Editor 117
Classes menu 45
Clear
  Input 150
  Ouput 150
Close 33, 92
Commit 27, 151
Compilation
  Options 155
Compilation options 103, 124, 131, 140, 155
Compile
  Application variable 133
  Class 47, 115
  Class with renaming 116
  Function 136
  Function Body 139
  Function Signature 138
  Method 119
  Method Body 124
  Method Signature 122
  Persistent Name 145
  Persistent Type 142
  Program 127
  Program Body 130
  Program Signature 128
  Version 107
Compiled flag 98–101
Configuration file 22, 160
Constant button 144
Constant dependency mode 157
Create
  And Compile 47, 115
  Application 61
  Application variable 70
  Base 40
  Base on a volume 41
  Class 46
  Function 76
  Inheritance 48
  library 38
  Method 53
  Named Object 57
  Persistent Names 86
  Persistent Type 80
  Program 66
  Schema 36
  Schema on a volume 37
  Version 106
Cross reference manager 16
Current version 118, 126, 135
<table>
<thead>
<tr>
<th>Customization</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Browser</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>variable Source Editor</td>
</tr>
<tr>
<td>Class Browser</td>
<td>Base</td>
</tr>
<tr>
<td>Class Source Editor</td>
<td>Class</td>
</tr>
<tr>
<td>Dashboard</td>
<td>Class Hierarchy</td>
</tr>
<tr>
<td>Dialog boxes</td>
<td>Function</td>
</tr>
<tr>
<td>Function Browser</td>
<td>Inheritance link</td>
</tr>
<tr>
<td>Function Source Editor</td>
<td>Method</td>
</tr>
<tr>
<td>Method Source Editor</td>
<td>Named Object</td>
</tr>
<tr>
<td>O2Shell Editor</td>
<td>Persistent Names</td>
</tr>
<tr>
<td>Persistent Name Browser</td>
<td>Persistent Type</td>
</tr>
<tr>
<td>Persistent Name Source</td>
<td>Program</td>
</tr>
<tr>
<td>Persistent Type Browser</td>
<td>Source Editor</td>
</tr>
<tr>
<td>Persistent Type Source</td>
<td>Version</td>
</tr>
<tr>
<td>Program Source Editor</td>
<td></td>
</tr>
<tr>
<td>Schema Browser</td>
<td></td>
</tr>
<tr>
<td>Version Editor</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Dependencies modes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>101</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application variable Source</td>
</tr>
<tr>
<td></td>
<td>Class</td>
</tr>
<tr>
<td></td>
<td>Function</td>
</tr>
<tr>
<td></td>
<td>Inheritance link</td>
</tr>
<tr>
<td></td>
<td>Method</td>
</tr>
<tr>
<td></td>
<td>Named Object</td>
</tr>
<tr>
<td></td>
<td>Persistent Name</td>
</tr>
<tr>
<td></td>
<td>Persistent Name Source Editor</td>
</tr>
<tr>
<td></td>
<td>Persistent Type Source Editor</td>
</tr>
<tr>
<td></td>
<td>Program</td>
</tr>
<tr>
<td></td>
<td>Schema</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dialog box</th>
<th>Dialog box</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Customization</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Display Value</th>
<th>Display Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>56, 85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Documentation</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Drag and drop</th>
<th>Drag and drop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>96</td>
</tr>
</tbody>
</table>
**INDEX**

**E**

**Edit**
- Application variable 133
- Class 115
- Function 136
- Function Body 139
- Function Signature 138
- Method 119
- Method Body 123
- Method Signature 122
- Persistent Name 145
- Persistent Type 142
- Program 127
- Program Body 130
- Program Signature 128

**Edit Documentation** 94
**Edition Menu** 148
**Editors, see Source Editor**
**Error messages** 102
**Execution menu**
- O₂Shell 151

**Expert utilization level** 157
**Exportable button** 114, 144

---

**F**

**File menu**
- Append 95
- Open 95, 96, 148
- Save As 149
- Shell Editor 148
- Source Editor 95, 96
- Write As 95, 96

**Flags** 98–101, 126, 132, 135, 141, 144
**Free dependency mode** 157

**Function** 72–77, 134–140
- Compile 136
- Create 76
- Delete 77
- Manipulation 73
- Rename 76
- Save All 76
- Signature 74
- Test 136

**Function Body** 139
- Compile 140
- Print 138
- Text Editor 141
- Versions 140

**Function Browser** 72
- Customization 173–175
- Function list 73
- Function menu 74
- Functions button 73

**Function menu** 74
**Function Signature** 138
- Compile 138

**Function Source Editor** 75, 134–140, 187–189
- Body menu 139
- Customization 187–189
- Graphic resources 187
- Signature menu 138
- Text Editor 139

---

**G**

**Garbage** 42
**Base** 42

**Global compilation options** 104
**Global options** 154
- Modification 154

**Go to line** 112
**Graphic Query** 153
**Graphic Query On Selection** 153
**Graphic resources**
- Application Browser 168–169
- Application variable 168–169
- Source
INDEX

Editor 185–187
Class Browser 170–173
Class Hierarchy 173
Class Source Editor 179–181
Dashboard 162–163
Function Browser 174–175
Function Source Editor 187–189
Method Source Editor 181–183
O2Shell Editor 195–196
Persistent Name Browser 177–179
Persistent Name Source Editor 191–193
Persistent Type Browser 175–177
Persistent Type Source Editor 189–191
Program Source Editor 183–185
Schema Browser 166–167
Version Editor 193–195
Graphical browsers 16
Graphical editors 16

Help 25
Dialog box 25
Hierarchy
Class 45
Delete 50

Ignore 151
Inheritance link
Create 47
Delete 49
Inherited methods 54
Input 147

Java 15

Keyboard accelerator 93

Local compilation options 103, 124, 131, 140
Lock 26
INDEX

M

Menu

Application variable Source 133
Applications 61
Bases 39
Body 105
Class Source 115
Classes 45
Edition 148
Execution 150
File 95, 148
Function 74
Function Body 139
Function Signature 138
Function Source 136
Method Body 124, 124, 131
Method Signature 122
Method Source 119
Methods 51
Name Source 145
Names 55
Options 33, 93
Persistent Names 84
Program Body 130
Program Signature 128
Programs 64
Query 153
Schemas 36
Type 79
Type Source 142
Variable Source 133
Variables 69
Window 33, 92, 149

Method 50–54, 117–125
Compile 119, 119
Create 52
Delete 54
Manipulation 50
Rename 53
Save All 54
Show Inherited 54
Show Local 54
Test 119
Uncompiled 53
Version 124

Method Body

Compilation options 124
Compile 124
Edit 123
Menu 123, 124, 131
Open 125
Print 124
Save 125

Method Signature 50, 122

Compile 122
Menu 122

Method Source Editor 51, 117–125, 181–183

Body menu 123
Customization 181–183
Graphic resources 181
Method menu 119
Signature menu 122
Text Editor 123, 125

Methods menu 51
Modified flag 98–101

N

Name

Display Value 85

Name menu

Class Browser 55
Persistent Name Browser 84

Name, Persistent 82–85
Named Object 54–56
Create 57
Delete 57
Display Value 56
Manipulation 54
Rename 57, 57
Save 58
New 148

nlook
Compilation option 105
Normal utilization level 157
Novice utilization level 157

O

O₂
Architecture 14
Exit 28
O₂ queries 152
O₂C 15
O₂Corba 15
O₂DBAccess 15
O₂Debug 22
O₂Engine 14
O₂Graph 15
O₂Kit 15
O₂Look 15
O₂ODBC 15
O₂Shell
Validate 151
O₂Shell Editor 16, 22, 146
  Alphanumeric command 150
  Commit 151
  Customization 195–99
  Execution menu 150
  Graphic resources 195
  Queries 146, 152
  Query menu 153
  Window menu 150
O₂Store 14
O₂Tools 15

O₂Tools
  Browsers 31
  Button 154, 160
  Dashboard 22
  Launching 19
  Preset configuration 22, 160
  Software requirements 18

o2toolsrc, see Configuration file
O₂Web 15
Obsolete flag 98–101
Open 95, 96, 108, 148
Optimize
  Compilation option 105
Optimize set option 155
Options 33, 93
  Compilation 103, 124, 131, 140, 155
  Global 104
  Local 103, 124, 131, 140
  Set 155
OQL 15
Output 147

P

Persistent Name 82–87, 143–145
  Compile 145
  Create 86
  Delete 87
  Display value 85
  Manipulation 83
  Rename 87
  Text Editor 146
Persistent Name Browser 82
  Customization 177–179
  Graphic resources 177
  Name menu 84
  Names button 82
Persistent Name Source Editor 85, 143, 191
  Customization 191–193
  Graphic resources 192
  Name menu 145
Persistent Type 78–81, 141–142
  Compile 142
  Create 80
  Delete 81
  Description 79
  Manipulation 79
  Text Editor 143
  Uncompiled type 81
Persistent Type Browser 78
  Customization 175–177
  Type list 79
  Type menu 79
  Types button 78
Persistent Type menu 79
Persistent Type Source Editor 80, 141, 189–191
  Customization 189–191
  Graphic resources 190
  Type menu 142
Preconfiguration 160
Print 94
  Application 62
  Application variable 134
  Function 137
  Function Signature 138
  Method Signature 123
  Persistent Name 145
  Persistent Type 142
  Program 128
  Program Signature 129
Private 114, 118, 126
Program 64–68, 125–131
  Compile 127
  Create 66
  Delete 68
  Manipulation 64
  Print 128
  Rename 67
  Save All 68
  Signature 67
  Test 127
Program Body
  Compile 130
  Text Editor 132
  Versions 131–??
Program Signature 128
  Compile 129
Program Source Editor 65, 125–131
  Body menu 130
  Customization 183–185
  Graphic resources 183
  Program menu 127
  Signature menu 128
  Text Editor 129
  Transaction box 126
Program Version Editor 131–??
Programs menu 64
Public 114, 118, 126
  Compilation option 105

Q

Query menu 153
Query On Selection 153
Quit 28

R

Read 114, 118, 126
Rename
  Application 62
  Application variable 71
  Base 41
  Class 48
  Function 76
  Method 53
  Other 116
  Persistent Name 87
  Program 67
  Schema 37
Replace item 111
Run 150
Run On Selection 151
### INDEX

**S**

- **Save**: 94  
  - Body: 108  
  - Function Body: 140  
  - Method Body: 125  
  - Program Body: 131  
  - Schema: 37  
- **Save As**: 149  
- **Schema**: 34–37  
  - Create: 36, 38  
  - Create on a volume: 37  
  - Delete: 38  
  - Manipulation: 35  
  - Rename: 37  
  - Save: 37  
  - Set: 36  
- **Schema Browser**: 34  
  - Bases list: 38  
  - Bases menu: 39  
  - Customization: 166–167  
  - Graphic resources: 166  
  - Schema list: 35  
  - Schemas button: 34  
  - Schemas menu: 36  
- **Schemas menu**: 36  
- **Search item**: 111  
- **Sensitive Editor area**: 97  
- **Set**:  
  - Base: 40  
  - Schema: 36  
  - TEST Status: 40  
- **Set options**: 155  
- **Setting**: 40  
- **Shadow flag**: 114, 118, 126, 133, 135, 142, 144  
- **Show All**: 41  
- **Show Inherited**: 54  
- **Show Local**: 54  

**Signature**  
- Compile: 109  
- Function: 138  
- Method: 122  
- Print: 109  
- Program: 128  
- Text: 110  

**Software requirements**: 18  

**Source Editor**  
- Application variable: 69  
- Body State: 100  
- Class: 46  
- Common menu items: 92  
- Compile: 93  
- Current State: 97  
- Edit Documentation: 94  
- File menu: 95  
- Function: 75  
- Messages: 98  
- Method: 51, 117  
- Options menu: 93  
- Overview: 90  
- Persistent Name: 143  
- Persistent Type: 141  
- Print: 94, 109  
- Program: 65  
- Save: 94  
- Window menu: 92  

**Source editor**  
- Common features: 90  

**Stack**  
- Compilation option: 105  

**Stack set option**: 155  

**Stat mode**: 157  

**State**  
- Body: 100  
- Source Editor: 97  

**Store**: 92, 94, 147, 150  

**System**  
- Architecture: 14
Test
  Application 61
  Function 136
  Method 119
  Program 127

Text
  Editor
    Go to Line 112

Text Editor 110
  Application variable 134
    Class 117
    Commands 112
    Function 139
    Function Body 141
    Menu 110
    Method 123
    Method body 125
    Persistent Name 146
    Persistent Type 143
    Program 129
    Program Body 132
    Replace 111
    Search 111
    Toggle 111

Trace
  Compilation option 105

Trace set option 155

Transaction box 126

Type
  Save All 82

Type menu 79

Type, Persistent 78–81, 141–142

Uncompiled
  Class 47
  Function 76
  Method 53
  Name 86
  Program 67
  Type 81
  Variable 71

User Manual overview 18

Utilization levels 157

Validate 27, 151

Variable
  Save All 72

Variable Source Editor, see Application variable Source Editor

Variables menu, see Application variable

Version
  Body 105
  Compile 107
  Create 106
  Current version 118, 126, 135
  Delete 107
  Function Body 140
  Method Body 124, 131

Version Editor 105–107
  Body menu 107
  Customization 193–195
  Graphic resources 194
  Versions menu 106

Visibility 114, 118, 126
INDEX

W

Window menu  92
  O₂Shell  150
Write As  95, 96, 108, 125, 131, 140

X

X windows  18