# Natural

## Using Natural Studio

Version 9.3.3

October 2025

ADABAS & NATURAL

This document applies to Natural Version 9.3.3 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

**Document ID: NATWIN-NNATWINUSING-933-20251014**

# Table of Contents

# Preface

This documentation explains how to work with Natural Studio, which is the development environment for Natural.

When you are working with Natural Studio in a remote development environment, see also the *Remote Development Using SPoD* documentation which describes several SPoD-specific features of Natural Studio.

This documentation is organized under the following headings:

| | |
|---|---|
| **Starting and Terminating Natural Studio** | How to start and terminate Natural Studio. |
| **Elements of the Natural Studio Window** | General information on the menu bar, toolbar, library workspace, workspace, application workspace, results window, command line, context menus and the status bar. |
| **Environments and Views in the Library Workspace** | Information on the local environment and the remote environment. How the objects in a library are presented in the logical view, flat view and file view. An overview of the icons used for the different types of nodes and how to display the properties for a node. |
| **Using Natural Libraries** | How to log on to a library, create a new library, filter libraries and objects, and find objects in a library. How to copy, move, rename and delete libraries, and how to define your own logon library. |
| **Creating, Maintaining and Executing Natural Objects** | Describes the Natural system commands for which menu commands are provided with Natural Studio. How to invoke an editor. How to list, check, save, stow, catalog, run and execute objects. How to clear an editor window. |
| **Managing Natural Objects** | How to copy, move, export, import, rename, and delete objects. How to change the type of a Natural object, print objects and views, and refresh the display. |
| **Managing Resources** | How to create and edit resources. |
| **Managing the Different Types of Windows** | How to work with list view windows and dockable windows. How to arrange and activate list view windows and editor windows. |
| **Issuing Commands in the Command Line** | How to use the command line. |
| **Customizing Natural Studio** | How to change Natural Studio's command, toolbar and shortcut key assignments. |
| **Setting the Options** | Information on the options that can be set in Natural Studio (for example, workspace options or program editor options). |
| **Using Session Parameters** | How to set the session parameters for the current session. |
| **Displaying System Information** | Information on the commands that are provided in the **Tools > System Information** menu. |

| | |
|---|---|
| Using Development Tools | Information on the commands that are provided in the **Tools > Development Tools** menu. |
| Using Configuration Tools | Information on the commands that are provided in the **Tools > Configuration Tools** menu. |
| Using Help | Information on the commands that are provided in the **Help** menu. |
| Shortcut Keys | An overview of the shortcut keys used in Natural Studio. |
| Rules and Naming Conventions | Natural-specific rules and naming conventions for objects, libraries and user-defined variables. |

# 1 About this Documentation

## Document Conventions

| Convention | Description |
|---|---|
| **Bold** | Identifies elements on a screen. |
| `Monospace font` | Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties. |
| *`Italic`* | Identifies:<br>Variables for which you must supply values specific to your own situation or environment.<br>New terms the first time they occur in the text.<br>References to other documentation sources. |
| `Monospace font` | Identifies:<br>Text you must type in.<br>Messages displayed by the system.<br>Program code. |
| { } | Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols. |
| \| | Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the \| symbol. |
| [ ] | Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols. |
| ... | Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...). |

## Online Information and Support

### Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

### Product Training

You can find helpful product training material on our Learning Portal at **https://learn.softwareag.com**.

### Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software GmbH resources.

**Product Support**

Support for Software GmbH products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

## Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

# 2 Starting and Terminating Natural Studio

It is also possible to start and terminate a Natural batch session. See *Natural in Batch Mode* in the *Operations* documentation.

## Starting Natural Studio

When Natural is installed, an entry for starting Natural is automatically added to the Windows **Start** menu.

**Note:** When Natural has been installed in an environment protected by the Windows firewall and can therefore not be started, see *Using the Windows Firewall with Natural* in the *Operations* documentation.

When you start Natural, Natural Studio, the development environment for Natural, appears. The toolbars and windows are placed at the same positions as when Natural Studio was left the last time. See ***Elements of the Natural Studio Window*** for further information.

When you have previously worked in a remote development environment and have therefore collapsed the node for the local environment, the local environment also appears collapsed when you restart Natural Studio.

When Natural Security is active, all definitions made in Natural Security have an influence on Natural Studio. Thus, the use of specific libraries, objects, commands and other features may be restricted to specific users only.

**Notes:**

1. You can set up your own environment by specifying dynamic parameters when starting Natural Studio. See *Dynamic Assignment of Parameter Values* in the *Operations* documentation.
2. When Natural Studio has already been started, you can also set session parameters for the current session. See ***Using Session Parameters***.
3. While Natural Studio is loaded, the Natural startup map is displayed informing you about the loading progress. See also ***Suppressing the Natural Startup Map***.

## Terminating Natural Studio

When you terminate Natural Studio, the positions of the toolbars and windows are stored.

### To terminate Natural Studio

- From the **Object** menu, choose **Exit**.

Or:

Choose the following toolbar button:

Or:

Press ALT+F4.

Or:

From the Control menu, choose **Close**.

Or:

Choose the corresponding standard button at the right of the title bar.

Or:

Enter the system command `FIN` in the **command line**.

Or:

Execute a program that contains a `TERMINATE` statement.

**Note:** Termination methods can also be defined with the Configuration Utility. See also *Program Loading and Deletion* in the *Overview of Profile Parameters* section of the *Configuration Utility* documentation.

## Suppressing the Natural Startup Map

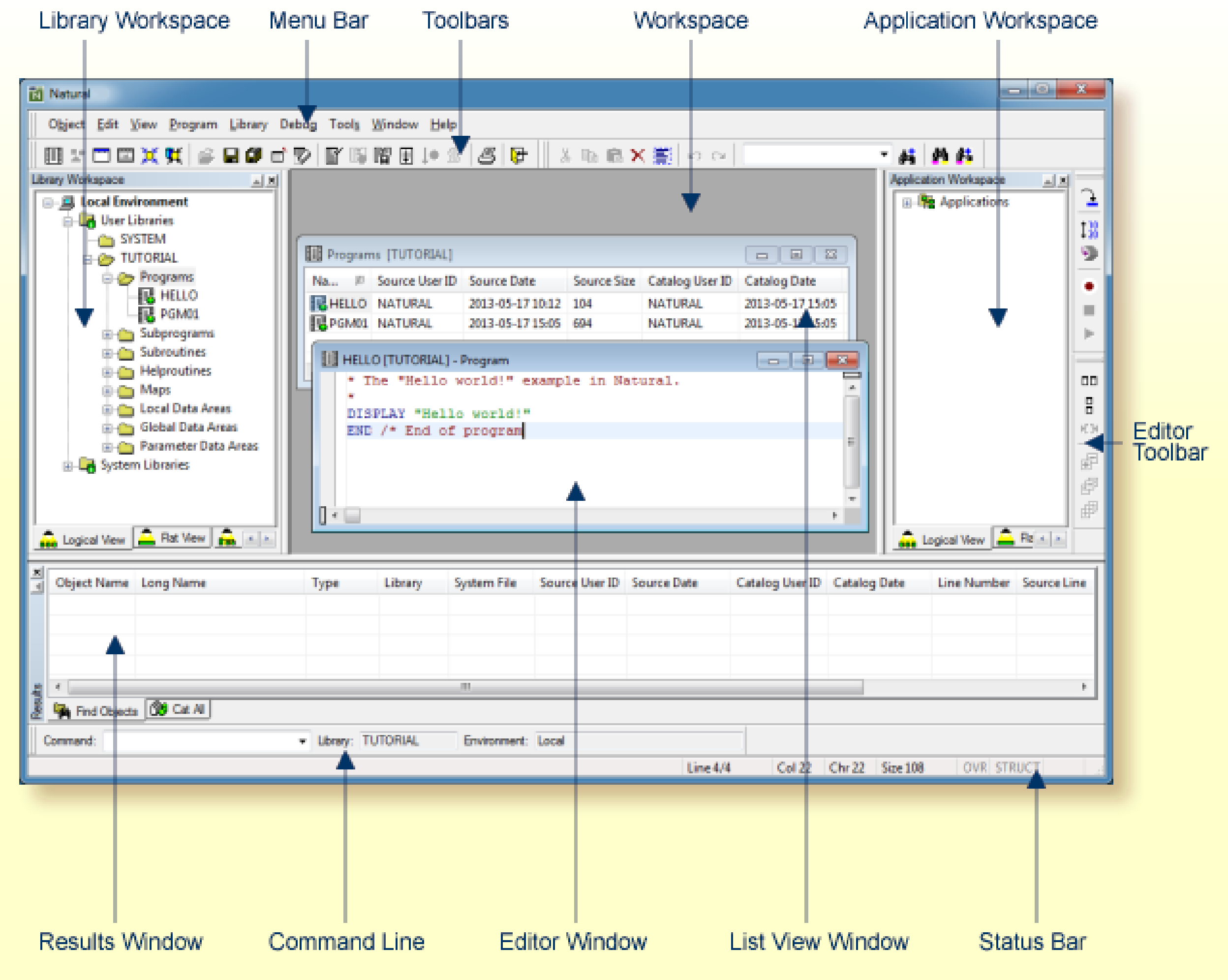
You can suppress the display of the Natural startup map. To do so, create a shortcut on your desktop for starting Natural Studio and change its properties. To suppress the startup map, specify a colon (:) after *natural.exe* in the **Target** text box. For example:

```
C:\SoftwareAG\Natural\Bin\natural.exe :
```

# 3 Elements of the Natural Studio Window

When you start Natural Studio for the very first time, only the library workspace is displayed on the left side of the Natural Studio window with the current logon library selected.

When several elements of the Natural Studio window have been activated and you are currently editing a program, the window may look as follows:



Programs [TUTORIAL]

| Na... | Source User ID | Source Date | Source Size | Catalog User ID | Catalog Date |
|---|---|---|---|---|---|
| HELLO | NATURAL | 2013-05-17 10:12 | 104 | NATURAL | 2013-05-17 15:05 |
| PGM01 | NATURAL | 2013-05-17 15:05 | 694 | NATURAL | 2013-05-17 15:05 |

HELLO [TUTORIAL] - Program

```
* The "Hello world!" example in Natural.
*
DISPLAY "Hello world!"
END /* End of program
```

| Object Name | Long Name | Type | Library | System File | Source User ID | Source Date | Catalog User ID | Catalog Date | Line Number | Source Line |
|---|---|---|---|---|---|---|---|---|---|---|

When you start Natural Studio, your startup library is automatically selected in the library workspace; see also *Defining Your Own Startup Library*. The startup library and many other profile parameters can be changed permanently using the Configuration Utility. Thus you can set up your own user profile. See the *Configuration Utility* documentation for further information. If you only want to change specific settings for the current session, you change them from within Natural Studio; see *Using Session Parameters*.

## Menu Bar

The following menus are always available:

| Menu | Using the commands in this menu, you can ... |
|---|---|
| **Object** | Manage (for example, create or delete) Natural objects, or exit Natural Studio. The commands shown in this menu depend on the editor window or list view window which is currently active. |
| **View** | Show or hide the various elements of the Natural Studio window, define a filter for the libraries or objects shown in the library workspace, refresh the display, print a tree view or list view, or display the properties of an object. |
| **Library** | Create a new library, catalog all objects in a library, or find objects. |
| **Debug** | Debug stowed or cataloged Natural programs and dialogs. |
| **Tools** | Display system information, invoke development and configuration tools, specify session parameters, customize Natural Studio (for example, define the toolbars to be shown), define options (for example, for a specific editor), map a remote environment or an application server. |
| **Help** | Invoke the online documentation. |

See the *Debugger* documentation for detailed information on the commands in the **Debug** menu.

Additional menus are available when a list view window or an editor window has been opened. When at least one of these windows is open, the **Windows** menu is available. See ***Managing the Different Types of Windows*** for further information.

When an editor window is active, additional menus are shown. For example, when the program editor is currently active, the menus **Edit** and **Program** are shown, and when the data area editor is active, the menus **Edit**, **Insert** and **Field** are shown. Additional editor-specific commands are then also available from the menus listed in the above table. See ***Invoking Editors*** for further information.

You can also define different commands (including your own user commands) to be shown in a menu. See ***Customizing Natural Studio*** for further information.

**Notes:**

1. The commands shown in a menu are determined by the active editor. They have precedence over the commands which are shown when an editor is not active. Thus it may happen that certain commands are only available for the selected object in the library workspace after the corresponding editor has been activated.
2. When Natural Security is active, all commands which are not allowed to be used are dimmed.

Many commands can also be chosen from a **toolbar** and/or **context menu**.

## Toolbars

You can execute the most important functions using a toolbar button.

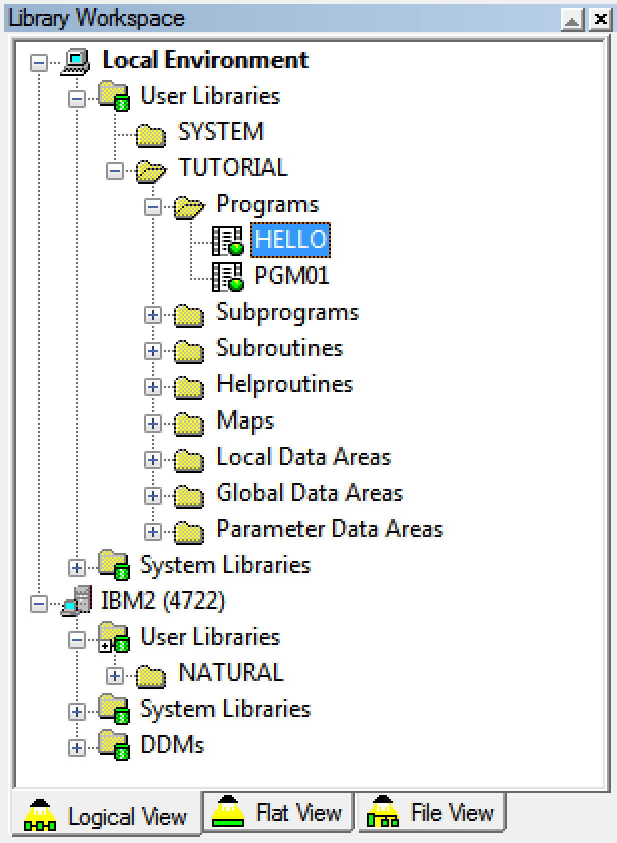
Natural Studio provides several toolbars. Only a few of them are initially shown in the Natural Studio window. Each editor has one or more special toolbars which provide fast access to the commands available in the editor-specific menus. When an editor window is active, the toolbars for this editor are shown. When you invoke the **context menu** by clicking the workspace or a toolbar, you can enable or disable the display of specific toolbars which apply to the current context (for example, when the program editor is currently active, you can enable further toolbars which apply to the program editor).

When you move the mouse pointer over a toolbar, a brief description for the button is shown in the status bar. In addition, tooltips may appear for the buttons when this feature has been switched on in the **Customize** dialog box. In the **Customize** dialog box, it is also possible to switch the display of a specific toolbar on and off, to add your own toolbars, or to define different buttons for an existing toolbar. See ***Customizing Natural Studio*** for further information.

**Note:** When Natural Security is active, all toolbar buttons which are not allowed to be used are dimmed.

## Library Workspace

The library workspace is a tree view window. It is used to administer the Natural system files in the current environment in a hierarchical manner.



Tabs are provided for different views: logical view, flat view and file view. See ***Environments and Views in the Library Workspace*** for further information.

In the different views, you can expand or collapse the nodes in the tree by clicking the plus or minus sign in front of a node. When using the keyboard, select the node and press RIGHT-ARROW or LEFT-ARROW.

When you expand or collapse a node in one view, this action is not applied to the other views. Thus, it is possible to view the contents of different nodes in different views.

Depending on the current selection, the nodes in the library workspace can be opened into editor windows or list view windows. See ***Workspace*** below.

In the library workspace, only one node can be selected at a time. If you want to select several nodes (for example, in order to delete several nodes at the same time) you have to open a list view window. See ***List View Windows*** for further information.

The library workspace provides access to two different types of environment: the local environment and a remote environment. The remote environment requires that Natural Development Server has been installed on z/OS, Linux, or Windows. This is also known as the Single Point of Development (SPoD) concept. See *Accessing a Remote Development Environment* in the *Remote Development Using SPoD* documentation for further information.

Using a filter, you can reduce the number of items that are shown in the tree. See *Filtering Libraries and Objects* for further information.

### To switch the library workspace display on and off

- From the **View** menu, choose **Library Workspace**.

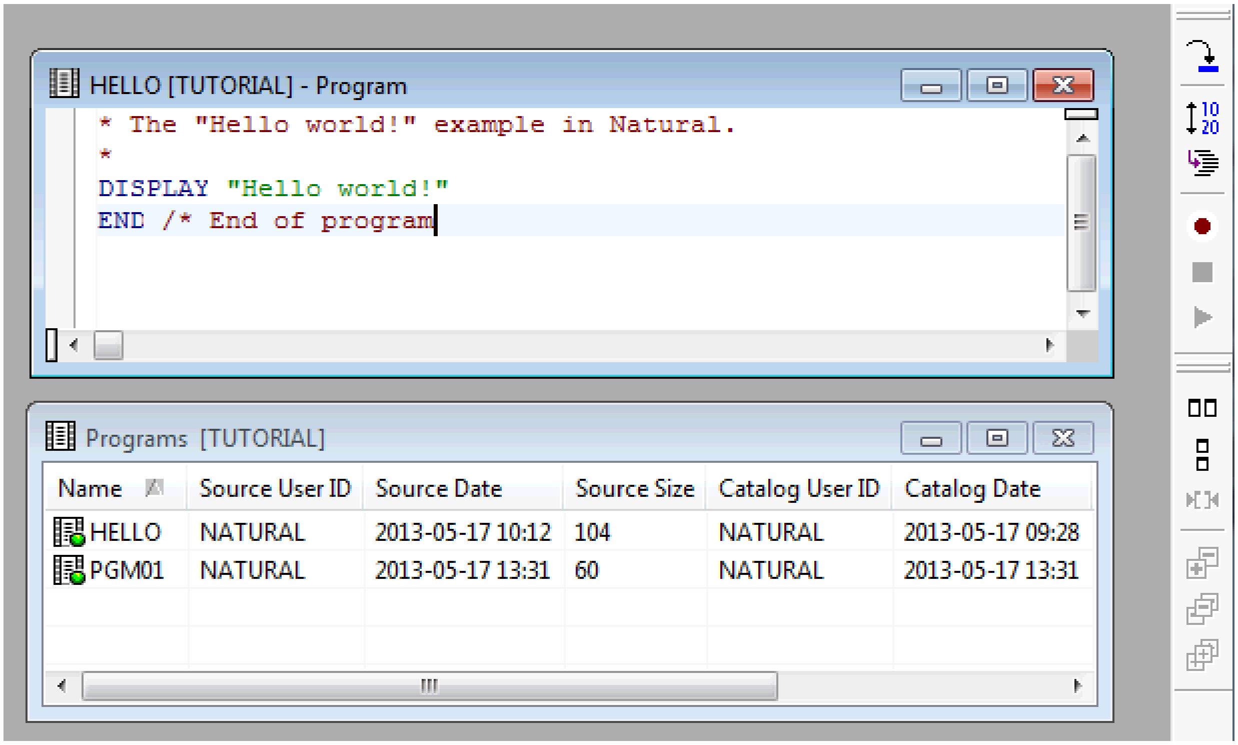
  Or:

  Press ALT+1.

  When the library workspace is displayed in the Natural Studio window, a check mark is shown next to this menu command.

## Workspace

Editor windows and list view windows are opened in the workspace.



```
* The "Hello world!" example in Natural.
*
DISPLAY "Hello world!"
END /* End of program
```

| Name | Source User ID | Source Date | Source Size | Catalog User ID | Catalog Date |
|---|---|---|---|---|---|
| HELLO | NATURAL | 2013-05-17 10:12 | 104 | NATURAL | 2013-05-17 09:28 |
| PGM01 | NATURAL | 2013-05-17 13:31 | 60 | NATURAL | 2013-05-17 13:31 |

When you create a new object (for example, a program) an editor window appears. An editor window also appears when you open an existing object from the library workspace. See *Invoking Editors* for further information.

When an editor window is active, an editor-specific toolbar is shown. By default, this type of toolbar appears to the right of the workspace. See the *Editors* documentation for information on the commands that can be chosen from such a toolbar.

For each group node in the library workspace (for example, the **Programs** node in the logical view), a list view window can be opened in order to display more detailed information on the selected node. See ***Opening a List View Window*** for further information.

A wallpaper can be used for the workspace. See ***Workspace Options*** for further information.

See also ***Managing the Different Types of Windows***.

## Application Workspace

Initially, the application workspace is not shown.

The application workspace can only be used when you are working with Natural Studio in a remote development environment (on a z/OS, Linux, or Windows computer) using Natural Single Point of Development (SPoD). It is used to administer Natural applications in a remote environment in a hierarchical manner.

Application Workspace
Applications
Logical View
Flat View
File View

Like the library workspace, the application workspace is also a tree view window and it also has tabs for different views: logical view, flat view and file view. See *Managing Applications* in the *Remote Development Using SPoD* documentation for detailed information.

### To switch the application workspace display on and off

- From the **View** menu, choose **Application Workspace**.

  Or:

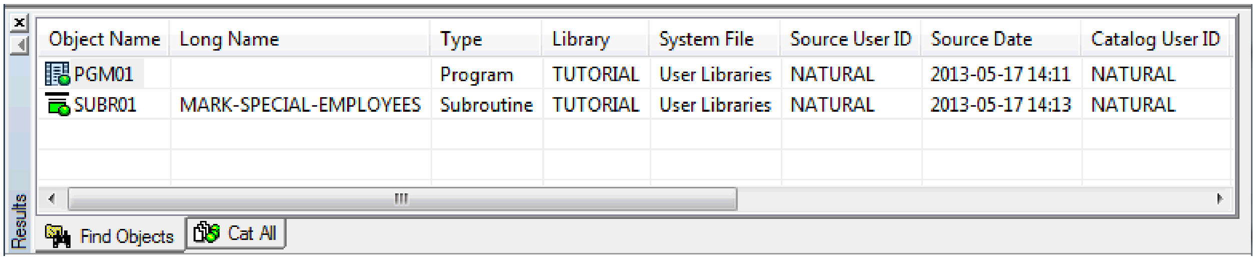
  Press ALT+2.

  When the application workspace is displayed in the Natural Studio window, a check mark is shown next to this menu command.

## Results Window

Initially, the results window is not shown.

By default, a tab for the results of the **Find Objects** command and another tab listing the objects that were not cataloged successfully with the **Cat All** command are shown (see *Finding Objects in a Library* and *Cataloging the Objects in a Library* for detailed information on these commands). Example:



| Object Name | Long Name | Type | Library | System File | Source User ID | Source Date | Catalog User ID |
|---|---|---|---|---|---|---|---|
| PGM01 | | Program | TUTORIAL | User Libraries | NATURAL | 2013-05-17 14:11 | NATURAL |
| SUBR01 | MARK-SPECIAL-EMPLOYEES | Subroutine | TUTORIAL | User Libraries | NATURAL | 2013-05-17 14:13 | NATURAL |

As a rule, when you select an entry on the **Find Objects** or **Cat All** tab, you can use most of the commands that can also be used in a **list view window** (menu bar, context menu or toolbar button).

When an entry is not selected on the **Find Objects** or **Cat All** tab, the following commands are available in the context menu:

| Command | Description |
|---|---|
| **View** | Define another view for the results window (for example, large icons). |
| **Copy View** | Copy the content of the results window, including the column headers, as normal text in column format. The individual columns are separated by tab characters. Thus, the copied content can easily be pasted into an application such as Microsoft Excel.<br>**Note:** You can also copy a single lines as normal text in column format (without the column header). When one or more lines are selected, the **Copy** command is available in the context menu. |

| Command | Description |
|---|---|
| **Clear View** | Remove all entries from a tab of the results window. This does not delete the objects. |
| **Select All** | Select all entries in the results window. |
| **Print View** | Print the content of the result window as normal text in column format. See also ***Printing Views***. |
| **Save View As** | Save the content of the result window as normal text in column format. The individual columns are separated by tab characters. A **Save As** dialog box appears in which you enter a file name and location. The resulting file can be opened, for example, with Microsoft Excel. |

Additional user-defined tabs may be shown that have been created with the Results Interface (see *Results Interface* in the *Programming Guide*) or for activated plug-ins (see ***Plug-in Manager*** later in this documentation). An additional tab is also shown when you try to unlock objects in a remote development environment (see *Unlocking Objects Manually* in the *Remote Development Using SPoD* documentation). These tabs may provide different commands in their context menus (not the commands listed above).

### To switch the results window display on and off

- From the **View** menu, choose **Results**.

  Or:

  Press ALT+5.

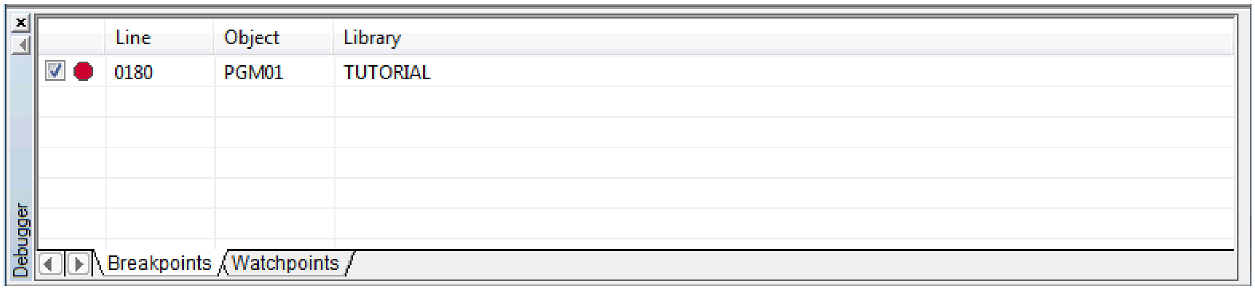
  When the results window is displayed in the Natural Studio window, a check mark is shown next to this menu command.

## Debugger Window

Initially, the debugger window is not shown.

When the debugger is not active, you can invoke the debugger window. It shows all breakpoints and watchpoints which have been set in the active environment. You can switch between the display of the breakpoints and watchpoints by choosing the corresponding tab at the bottom of the debugger window.

A question mark (?) in the first column indicates that the breakpoint or watchpoint has not been validated. Breakpoints and watchpoints can only be validated when the debugger is active and has reached the corresponding line in the editor.



| | Line | Object | Library |
|---|---|---|---|
| ☑ ● | 0180 | PGM01 | TUTORIAL |

It is possible to add or remove (toggle) breakpoints in the editor even if the debugger is not active. See *Adding and Removing a Breakpoint* in the *Debugger* documentation.

When you invoke a context menu in the debugger window, you can, for example, deactivate or delete breakpoints or watchpoints. For detailed information on the context menu commands, see *Setting Breakpoints and Watchpoints* in the *Debugger* documentation.

When the debugger is started, the debugger windows which are described in *Debugger* documentation are shown instead of the above debugger window. When the debugger is stopped, the above debugger window will be shown again - provided that its display has previously been switched on in the **View** menu.

### To switch the debugger window display on and off

- From the **View** menu, choose **Debugger**.

  Or:

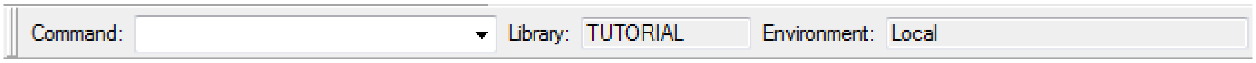
  Press ALT+6.

  When the debugger window is displayed in the Natural Studio window, a check mark is shown next to this menu command.

  **Note:** When the debugger is active, the above command is not available in the **View** menu.

## Command Line

Initially, the command line is not shown.



The command line consists of the following elements:

**Command**

You can execute system commands, utilities, generated programs and generated dialogs directly from the **Command** drop-down list box. See ***Issuing Commands in the Command Line*** for detailed information.

**Library**

The name of the current library is always indicated in the **Library** text box (read-only).

The current library is the library which is currently selected in the library workspace or in the application workspace. Or it is the library which contains the object that is currently selected in the library workspace or application workspace. When you select an editor window or a list view window, the corresponding library is considered as the current library. See ***Using Natural Libraries*** for further information.

**Environment**

The active environment is always indicated in the **Environment** text box (read-only).

There are two types of environment:

- local environment on a workstation (this is also the runtime environment of Natural Studio),
- remote environment on a development server (only available in a SPoD environment).

All commands that you issue are always applied to the active environment. When you edit a Natural object, the corresponding editor is invoked and the object is taken from the active environment. When you execute an object, it is executed in the active environment. Only one environment can be active at one point in time. See ***Environments and Views in the Library Workspace*** for further information.

### To switch the command line display on and off

- From the **View** menu, choose **Command Line**.

  Or:

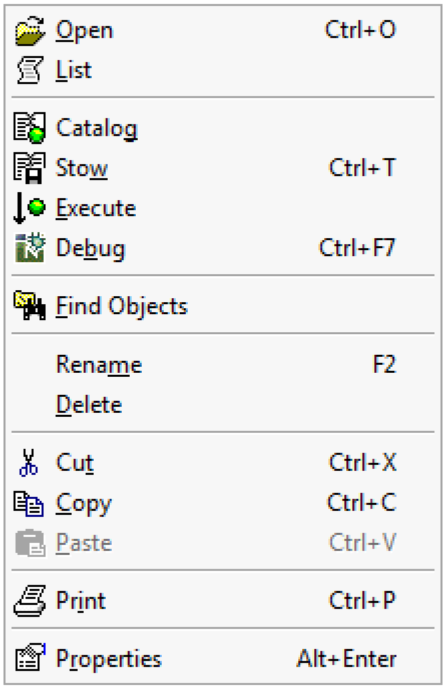
  Press ALT+3.

  When the command line is displayed in the Natural Studio window, a check mark is shown next to this menu command.

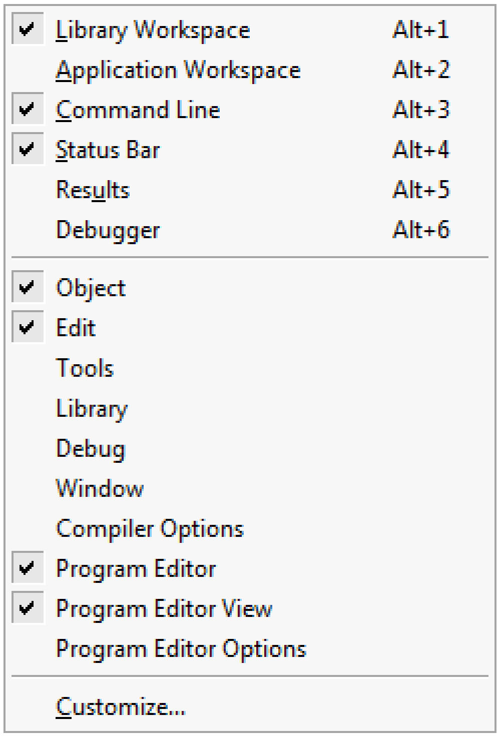
## Context Menus

Context menus provide for selection of commands that are valid in the current context. The content of a context menu depends on the currently selected object or the position within the Natural Studio window.

For example, the following context menu is available when you select a program in the library workspace:



Or when you invoke the context menu by clicking, for example, the workspace or toolbar with the right mouse button when the program editor is currently active, the following context menu appears:



Using the above context menu, you can switch the display of a specific element of the Natural Studio window on and off. The commands listed below the first divider apply to toolbars. Different toolbars will be listed when a different type of editor window is currently active.

You can also define different commands (including your own user commands) to be shown in a context menu. See *Customizing Natural Studio* for further information.

**Note:** When Natural Security is active, all commands which are not allowed to be used are dimmed.

### To invoke a context menu

1 Select the element for which you want to invoke a context menu.

2 Click the right mouse button.

Or:

Press SHIFT+F10.

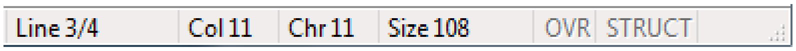
The context menu appears and you can now choose the required command.

## Status Bar

The status bar is the horizontal information line at the bottom of the Natural Studio window. It shows short help texts for the commands in the menu bar and context menu.

When a list view window is active, the name of the corresponding library is shown and the number of objects.

When an editor window is active, editor-specific information is shown at the right of the status bar. For example, the following information is shown for the program editor:



Line 3/4 | Col 11 | Chr 11 | Size 108 | OVR | STRUCT

### To switch the status bar display on and off

- From the **View** menu, choose **Status Bar**.

  Or:

  Press ALT+4.

  When the status bar is displayed in the Natural Studio window, a check mark is shown next to this menu command.

### Editor-specific Status Bar Information

The different editors show different information in the status bar.

When the corresponding option has been set in the **Options** dialog box, a status bar is also shown in an editor window.

**Note:** The information provided in the tables below applies to the status bars which are available in the different editor windows. In the status bar at the bottom of the application window, this information is in some cases shown in a different way and in a different sequence.

**Program Editor**

When the display of the status bar has been activated in the **program editor options**, the following information is shown in the program editor window:

| Information | Description |
|---|---|
| Line: *x* of *y* | *x* stands for the line in which the cursor is currently positioned. *y* stands for the total number of lines in the object. |
| Col | The column in which the cursor is currently positioned. |
| Size | The total number of characters in the source code. |
| Structured or Report | The programming mode of the active object. For further information, see *Natural Programming Modes* in the *Programming Guide*. |
| Modified | Indicates that the object has been modified since the last save. If "Modified" is not displayed, the object has not been modified since the last save. |
| INS or OVR | INS stands for insert mode: input does not overwrite existing text. OVR stands for overwrite mode: input overwrites existing text. |

### Data Area Editor

When the display of the status bar has been activated in the **data area editor options**, the following information is shown in the data area editor window:

| Information | Description |
|---|---|
| Line: *x* of *y* | *x* stands for the line which is currently selected. *y* stands for the total number of lines in the source area. |
| Size | The total number of bytes in the data area. |

### Map Editor

When the display of the status bar has been activated in the **map editor options**, the following information is shown in the map editor window:

| Information | Description |
|---|---|
| Name | The name of the active field. |
| Row | The top-most row in which the active field begins. |
| Col | The left-most column in which the active field begins. |
| Len | The length (characters) of the active field. |
| Format | The format of the active field. |
| Structured or Report | The programming mode of the active object. For further information, see *Natural Programming Modes* in the *Programming Guide*. |

### DDM Editor

When the display of the status bar has been activated in the **DDM editor options**, the following information is shown in the DDM editor window:

| Information | Description |
|---|---|
| Line: *x* of *y* | *x* stands for the line which is currently selected. *y* stands for the total number of lines in the DDM source. |
| DBID | The database ID from which the DDM is derived. |
| File | The number of the database file from which the DDM is derived. |
| Type | The type of database from which the DDM is derived. |

**Note:** To modify the database ID, file number and default sequence, see *DDM Header Information* in the *DDM Editor* section of the *Editors* documentation.

### Dialog Editor

When the display of the status bar has been activated in the **dialog editor options**, the following information is shown in the dialog editor window:

| Information | Description |
|---|---|
| Modified | Indicates that the dialog has been modified since the last save. If "Modified" is not displayed, the dialog has not been modified since the last save. |
| Selected | The handle name of the selected dialog element. You can select a different dialog element from this drop-down list box. |
| x: *n* y: *n* | Coordinates indicating the position of the selected dialog element. |
| w: *n* h: *n* | Width and height of the selected dialog element. |

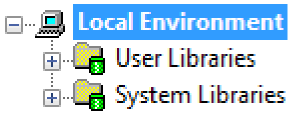
See *Dialog Editor Window* in the *Dialog Editor* section of the *Editors* documentation for more detailed information.

# 4 Environments and Views in the Library Workspace

The library workspace is used to administer the Natural system files in the current environment in a hierarchical manner as a tree view. In Natural terms, a system file is a collection of Natural libraries and a Natural library is a collection of Natural objects and resources.

## Local Environment

All nodes below the node **Local Environment** belong to the local Windows environment which is normally your own PC. This node is available in all views described below. Example for the logical view:



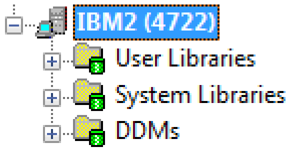
When the local environment is active, the node name **Local Environment** is shown in bold, and the **Environment** text box of the command line contains the string "Local".

Using the Configuration Utility, alias names can be defined for the system files. When an alias name has been defined, it is shown for the corresponding node in the library workspace. See *System Files* (global configuration file) in the *Configuration Utility* documentation.

## Remote Environment

A node for a remote environment can only be shown when you are using Natural Single Point of Development (SPoD) for remote development on a z/OS, Linux, or Windows computer. See *Accessing a Remote Development Environment* in the *Remote Development Using SPoD* documentation for further information.

The node name is either a user-defined description, or (if a description has not been specified) the name of the server and the port number. This node is available in all views described below. Example for the logical view:



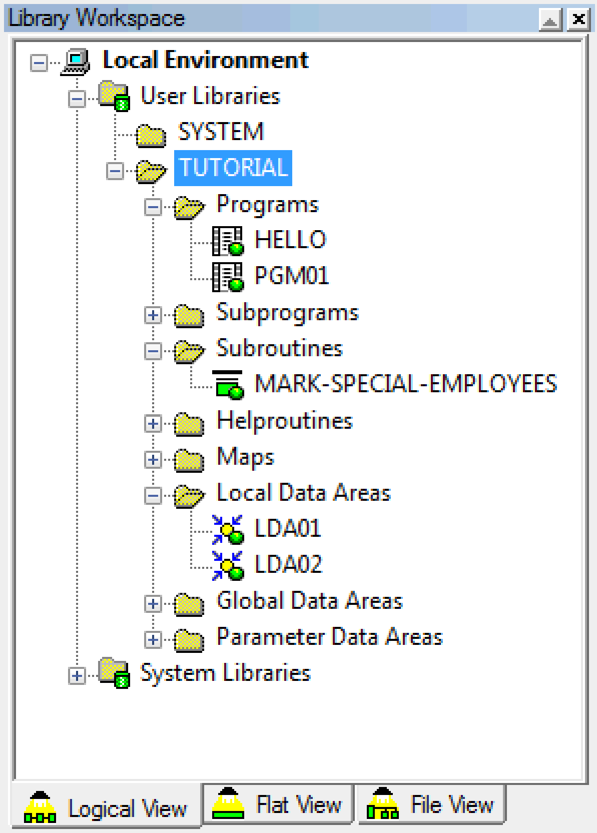
When the remote environment is active, the node name is shown in bold, and the **Environment** text box of the command line also contains the node name.

When the remote environment is currently not connected, the string "Disconnected" is shown to the right of the node name.

Alias names can also be shown for the system files in a remote Linux or Windows environment. The alias names are defined with the Configuration Utility which is available with Natural on these platforms.

## Logical View

In the logical view, different nodes are provided for user libraries and system libraries. The objects in a library are grouped into different nodes, according to their Natural object types.



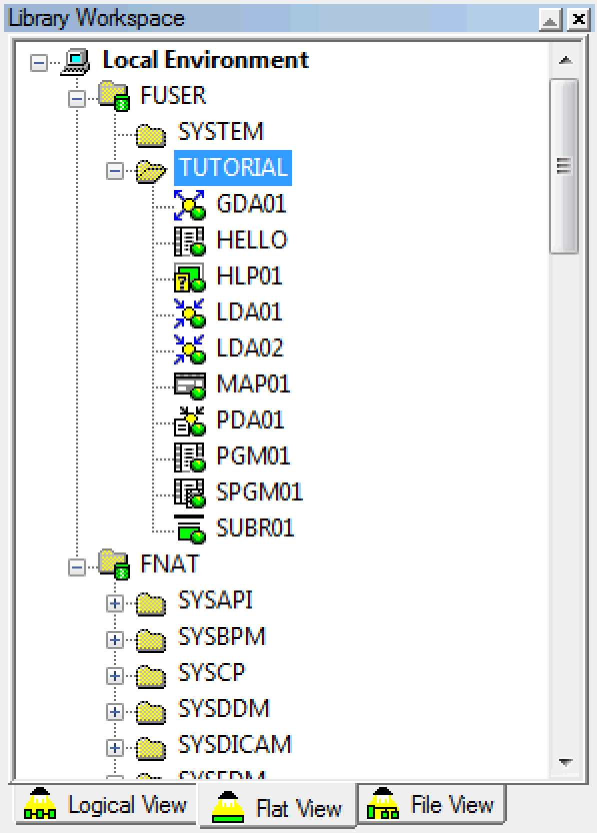
For example, all programs are shown in a group node called **Programs**. Thus, if you want to view the available programs in a library, you must expand the **Programs** node. This is not required in the flat view (see below).

For subroutines, functions, classes and DDMs, the long names can be shown in the logical view. The other views show the short names (max. 8 characters; see also *Object Naming Conventions*). For example, for a subroutine, the name that has been defined in the program is shown (which may exceed 8 characters). The flat view and file view show the name that has been defined when saving the subroutine.

## Flat View

In the flat view, different group nodes are provided for system files containing user-written programs (`FUSER`) and system programs (`FNAT`).

The objects in a library are displayed without any grouping. The object type is always indicated by the icon next to an object. See ***Object Nodes***.

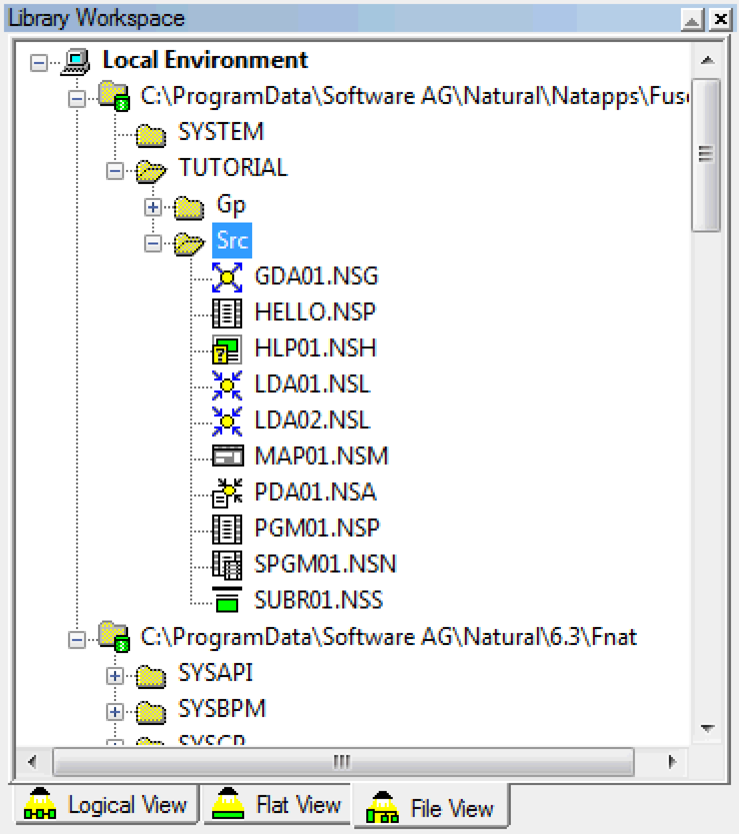


## File View

In the file view, the structure of the Natural environment is represented as available on the file system. A Natural system file is displayed as the path to the *Src* and *Gp* subdirectories.

The group node **Src** contains all Natural sources and the group node **Gp** contains all generated programs (Natural executables). Group nodes for other subdirectories (for example, **Res** for resources or **Err** for error codes) may also be available.

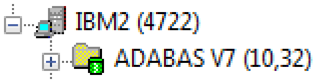
**Note:** Generated programs are also referred to as “cataloged objects”. They are created when objects are **stowed** or **cataloged**.



The object type is always indicated by the icon next to an object. See ***Object Nodes***.

The Natural objects are displayed with the corresponding file extensions (for example, ".NGP" for “Natural Generated Program” or ".NSP" for “Natural Source Program”). For further information on the possible file extensions, see *System File Structure* in the *Operations* documentation.

In a remote environment, the name of each mapped remote database is shown with database ID and file number. Example:



IBM2 (4722)

ADABAS V7 (10,32)

# Icons for the Nodes

This section describes the icons that are used in the tree view windows and list view windows for the different types of nodes.

- Group Nodes
- Object Nodes

## Group Nodes

The following icons are used for the different group nodes:

Local environment.

System file containing user libraries or system libraries. When working with SPoD in a remote environment, the system file also contains DDMs. See also ***Library Types***.

Inactive system file. See also ***Library Types***.

System file with an active filter. See ***Filtering Libraries and Objects*** for further information.

Nodes for libraries or groups of Natural objects. The content of a group depends on the view (logical view, flat view or file view).

Library with an active filter. See ***Filtering Libraries and Objects*** for further information.

Library with suppressed line numbers in the source code. See ***Suppressing Line Numbers in the Source Code*** for further information.

Library with an active filter and suppressed line numbers in the source code.

Remote environment.

## Object Nodes

For a detailed description of each object type that is available with Natural, see *Objects for Natural Application Management* in the *Programming Guide*.

The following icons are used for the different object types:

Program

Class (see also *Class Builder Nodes* in the *Editors* documentation)

Subprogram

Function

Subroutine

Copycode

Helproutine

Text

Dialog

Map

Adapter

Local data area (LDA)

Global data area (GDA)

Parameter data area (PDA)

Data definition module (DDM)

Natural objects can consist of the Natural source, the Natural generated object (for example, a generated program or a generated local data area) or both. The difference is reflected in the icon. The above icons can also be shown as follows (example for a program):

Without a green ball and not gray: only the source of the object is available.

With a green ball and not gray: source and generated object are available.

With a green ball and gray: only a generated object is available and no source.

See ***Invoking Editors*** for information on how to create or open such an object. Or see the *Editors* documentation for detailed information on how to use the different editors.

Icons are also shown for resources. See ***Managing Resources*** for further information.

## Properties for the Nodes

You can display the properties for each node shown in the library workspace or in a list view window. Different information is shown in the resulting **Properties** dialog box, depending on the selected item (see the examples below).

### To display the properties for a node

1. Select the node for which you want to display the properties.
2. From the **View** menu, choose **Properties**.

   Or:

   Invoke the context menu and choose **Properties**.

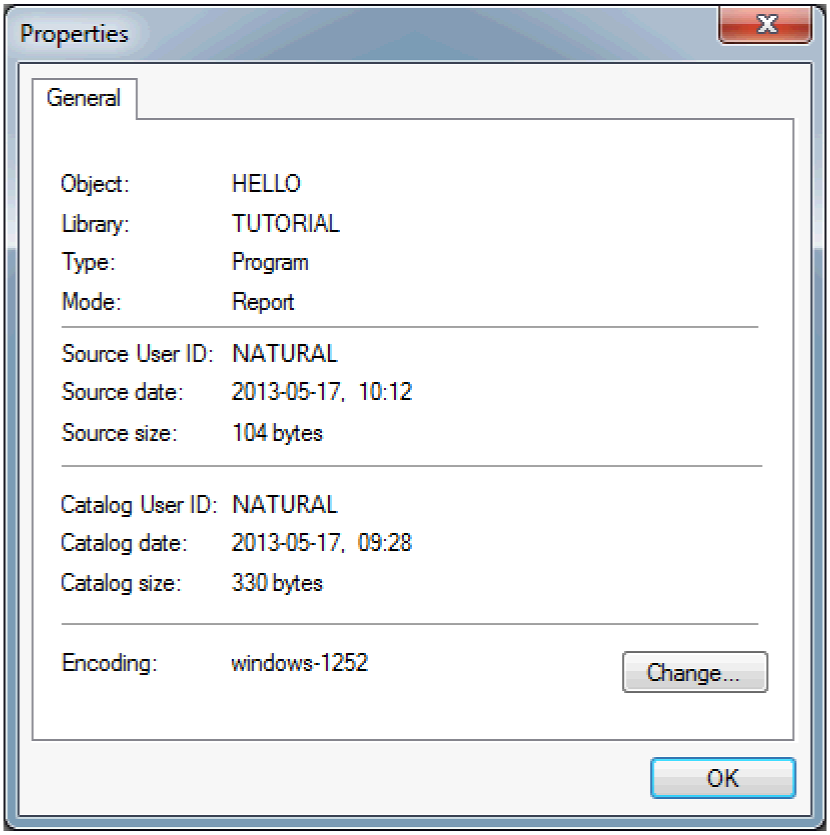
   Or:

   Press ALT+ENTER.

   The **Properties** dialog box appears.

**Note:** You can also display the properties for the nodes in the application workspace. Other than the **Properties** dialog boxes that are shown for the nodes in the library workspace, the **Properties** dialog boxes that can be invoked for the application server and for an application provide more than one page. It is possible to enter information on the additional pages. See *Managing Applications* in the *Remote Development Using SPoD* documentation for further information.

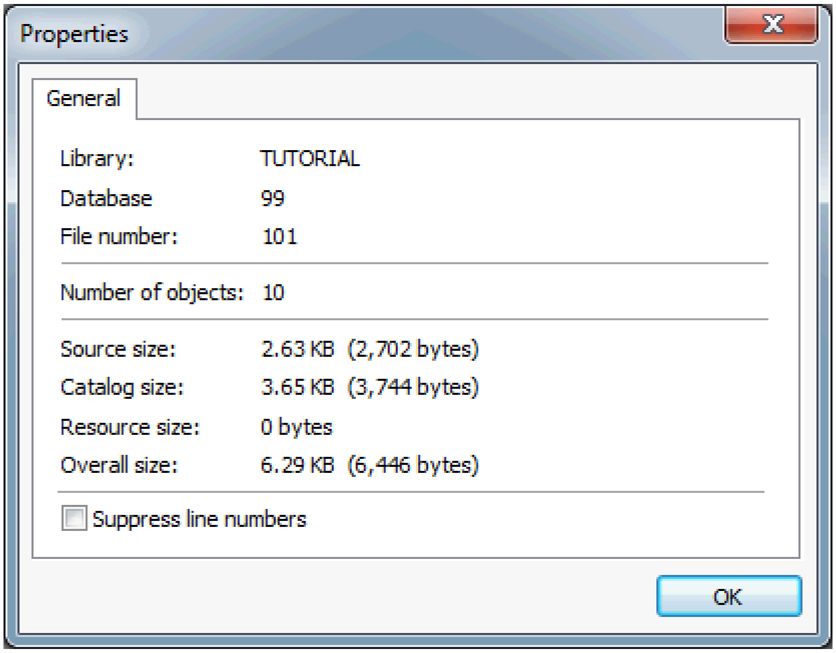
### Example for a Program



| | |
|---|---|
| Object: | HELLO |
| Library: | TUTORIAL |
| Type: | Program |
| Mode: | Report |
| Source User ID: | NATURAL |
| Source date: | 2013-05-17, 10:12 |
| Source size: | 104 bytes |
| Catalog User ID: | NATURAL |
| Catalog date: | 2013-05-17, 09:28 |
| Catalog size: | 330 bytes |
| Encoding: | windows-1252 |

If available for the selected object type, the **Change** button is provided. When you choose this button, the **Encoding** dialog box appears in which you can select a different code page. See *Changing the Code Page of a Natural Object* for further information.

### Example for a Library



Library: TUTORIAL

Database 99

File number: 101

Number of objects: 10

Source size: 2.63 KB (2,702 bytes)

Catalog size: 3.65 KB (3,744 bytes)

Resource size: 0 bytes

Overall size: 6.29 KB (6,446 bytes)

□ Suppress line numbers

When you display the properties for a library, the **Suppress line numbers** check box is available. When this check box is selected, new sources in this library will be saved without line numbers in the file system. The line numbers in existing sources will be suppressed the next time these sources are saved. See *Suppressing Line Numbers in the Source Code* for further information.

# 5 Using Natural Libraries

A library in Natural terms is the container for Natural objects. A Natural application can access objects in multiple libraries depending on how the environment is set up.

Libraries are managed in the library workspace or in a **list view window** that has been opened from the library workspace.

**Note:** The applications in the application workspace also contain libraries and objects. However, these are just links to the libraries and objects which are normally available in the library workspace. See *Managing Applications* in the *Remote Development Using SPoD* documentation for further information.

## Library Types

The following types of libraries are available.

| Library Type | Description |
|---|---|
| System library | The system libraries are also known as the `FNAT` system file which is version-dependent. A system library contains all Natural objects required to run Natural tools (or utilities). An example of a system library is the library `SYSERR` which represents a system tool that is used to maintain error messages. The name of a system library always starts with "SYS".<br>The system libraries are reserved for internal purposes only and are subject to change without notice.<br>**Important:** Do not store any user-specific objects in a Natural system library. User-specific objects can be overwritten when a new version of Natural is installed.<br>**Note:**<br>1. Natural provides example libraries containing example programs for demonstration purposes. The names of these example libraries start with "SYSEX".<br>2. The system libraries `SYSLIB` and `SYSLIBS` are reserved for internal use. |
| User library | The user libraries are also known as the `FUSER` system file. A user library contains all user-defined objects (for example, programs and data areas) which make up an application. |
| Inactive library | All system files that are defined in the global configuration file and which are not defined in the parameter file that was used to start the current Natural session are inactive system files. See also *System Files* in the *Overview of Configuration File Parameters* section of the *Configuration Utility* documentation.<br>It is not possible to execute a Natural object in an inactive library. However, it is possible to delete or rename objects in an inactive library and to copy or move objects from an inactive library to an active library, and vice versa. |

| Library Type | Description |
|---|---|
| | If you want to use an inactive library, the corresponding inactive system file must become an active system file (either FUSER or FNAT). To activate an inactive system file, you have to define it in the current parameter file. See also *System Files* in the *Overview of Profile Parameters* section of the *Configuration Utility* documentation. |

See also ***Icons for the Nodes*** and ***Library Naming Conventions***.

## Steplibs

A steplib is a Natural user library or system library that is concatenated with the current user or system library. This avoids redundant storage of identical objects and helps organize applications. In addition, if Natural Security is installed, a steplib can be used to restrict access to particular objects.

Natural searches in a steplib when an object is not found in the current library (see ***Search Sequence for Object Execution*** for further information). The standard steplibs are the libraries SYSTEM in the FUSER and the FNAT system files.

If Natural Security is active, you can define additional steplibs in the security profile of each library. The entries in a library security profile override any definitions made outside Natural Security.

If Natural Security is not active, you can specify additional steplibs in the Configuration Utility, with the profile parameter STEPLIB. In addition, you can define further steplibs by using one of the application programming interfaces (for example, USR1025N or USR3025N) that are supplied for this purpose in the Natural system library SYSEXT.

The additional steplibs are searched for an object before the standard steplibs SYSTEM (FUSER and FNAT).

The currently active steplibs are shown on the **Steplibs** page of the **TECH** dialog box. See ***Technical Information*** for further information.

## Search Sequence for Object Execution

This section describes the sequence in which Natural libraries and system files are searched for a requested object that is to be executed from either a user library or a system library.

**Note:** If the profile parameter BPSFI is set to "ON" (the default setting is "OFF"), objects are searched for in the buffer pool first. This is the proposed setting in a production environment.

**The search sequence for a user-written object to be executed from a user library is as follows:**

1. The current library in the FUSER system file as defined by the system variable *LIBRARY-ID.
2. The steplibs (in sequence) as specified in the Natural Security profile for the current library or in the steplib table.
3. The default steplib as defined by the system variable *STEPLIB.
4. The library SYSTEM in the FUSER system file.
5. The library SYSTEM in the FNAT system file.

**The search sequence for a Natural object to be executed from a system library is as follows:**

1. The current "SYS" library in the FNAT system file as defined by the system variable *LIBRARY-ID.
2. The steplibs (in sequence) as specified in the Natural Security profile for the current library or in the steplib table.
3. The library SYSLIBS in the FNAT system file, which contains objects shared by system commands and utilities.
4. The library SYSTEM in the FNAT system file.
5. The library SYSTEM in the FUSER system file.

   Since the FUSER system file is searched last, you must provide an object that is used in both the FUSER and the FNAT system files (for example, a user exit routine for a Natural utility) only in one location, namely in FUSER.

## Logging on to a Library

In order to work with objects in a Natural library or to start an application inside a specific library, you must first log on to this library.

With Natural Studio, you can log on to a library in two different ways: automatically and manually. The name of the library to which you are logged on is always shown in the **command line**.

**Note:** When Natural Security is active, all libraries that you are not allowed to use are not visible in the library workspace.

### To perform an automatic logon to a library

- Select the required library in the library workspace or in a list view window.

  Or:

  Select an object in the library workspace or in a list view window.

Or:

Activate an open editor window.

**Note:** When you expand or collapse the node for a library in the tree by clicking the plus or minus sign in front of a node, you do not automatically logon to this library. A logon is only performed when you select the library (for example, by clicking on its name).

### To perform a manual logon to a library

- Enter the following system command in the command line:

  ```
  LOGON library-ID
  ```

  where *library-ID* is the ID (name) of the library you want to access.

  See also the description of the system command `LOGON`.

## Creating a New Library

A new library can only be created in a system file (`FNAT` or `FUSER`) located inside the local or remote environment. Inactive libraries cannot be created; they are read-only.

Only one library can be created at a time.

### To create a new library

1. In the library workspace, select the node for the system file that is to contain the new library (either below the node for the local environment or below the node of the development server to which you have previously connected).

   The nodes for the system files have different names, depending on the current view. For example, if you want to create a user library in the logical view, you have to select the node named **User Libraries**.
2. From the **Library** menu, choose **New**.

   Or:

   Invoke the context menu and choose **New**.

   A new library with the default name "USRNEW" is now shown in the tree. The default name is selected so that you can immediately enter a new name. Any text you enter automatically deletes the selection.
3. Enter a name for the library.

See also *Library Naming Conventions*.

**Note:** When the name is in edit mode, you can invoke a context menu. In addition to the standard Windows editing commands such as **Cut** and **Paste**, the context menu may also contain Windows commands for Unicode. See your Windows documentation for further information.

4 Press ENTER.

Or:

Click any other position in the library workspace.

Your new library is now available in all views (sorted alphabetically). When a filter is currently active, it may happen that your new library is not shown since its name starts with different characters. See *Filtering Libraries and Objects* for further information.

## Filtering Libraries and Objects

Using a filter, you can reduce the number of items that are shown in the library workspace, in the application workspace and in the list view windows that are opened from these workspaces.

When a filter has been defined in the library workspace, the same filter applies for the application workspace, and vice versa. All views (logical view, file view and flat view) reflect this filter. In a remote environment, all views which belong to the environment specified by server name and port number reflect this filter. It may be necessary to **refresh** the display in the other workspace before the change can be seen there.

The type of filter that is applied depends on your selection. When you select a system file, a library filter will be applied. When you select a library, an object filter will be applied; this object filter also allows the definition of resources.

The following exception applies for DDMs: When FDDM mode has been activated for the local environment or for a remote Linux or Windows environment, you have to select the `FDDM` system file in order to apply an object filter for the DDMs. In a remote z/OS environment, the DDMs are always stored in the system file `FDIC`; therefore, you have to select the `FDIC` system file in order to apply an object filter for the DDMs.

When a filter has been applied, the icons that are shown in the tree contain an additional plus sign:

System file with an active filter.

Library with an active filter.

### To define a filter

1 To define a filter for libraries, select a system file.

Or:

To define a filter for the objects and resources in a library, select the library.

**Note:** In a **list view window**, it is possible to select several libraries at the same time. Thus, you can define the same filter for all selected libraries at once.

Or:

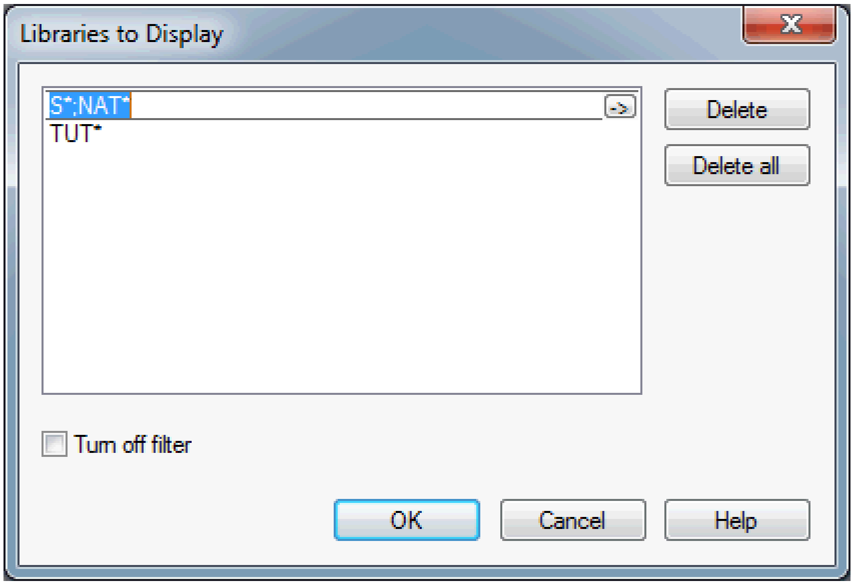
To define a filter for DDMs, select the system file `FDDM` (if FDDM mode is active in the local environment or in the remote Linux or Windows environment) or `FDIC` (remote z/OS environment).

2 From the **View** menu, choose **Display Filter**.

Or:

Invoke the context menu and choose **Display Filter**.

The following dialog box appears when you have selected a system file (except a system file for DDMs):



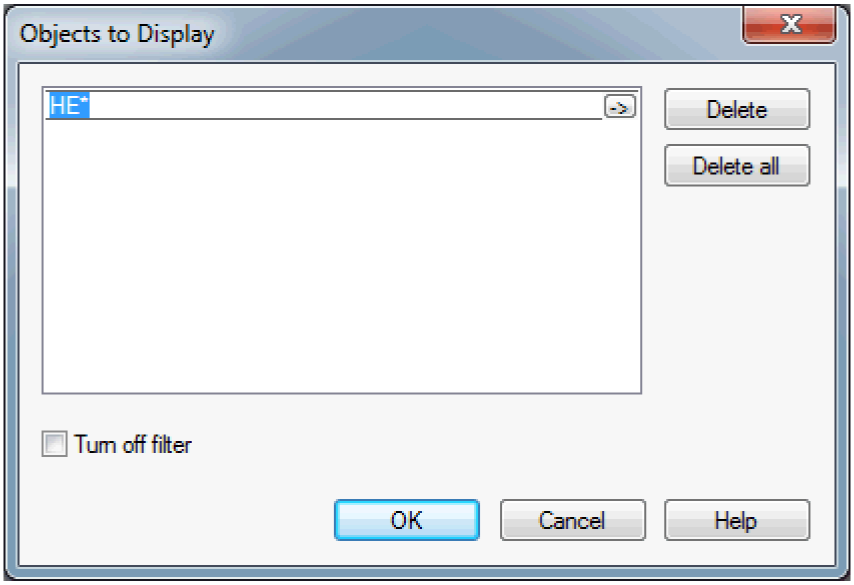
Libraries to Display

S*;NAT*

TUT*

□ Turn off filter

A similar dialog box appears when you have selected a library or a system file for DDMs. It has a different title:



When a filter has already been defined, it is shown in the dialog box (as in the above examples).

3 In the text box, enter the names of the libraries or objects/resources that are to be shown. All names must be separated by a semicolon.

The Natural-specific naming conventions for libraries and objects apply for the definition of a filter. See *Rules and Naming Conventions*.

Instead of entering the names in a single text box, you can also enter them using several text boxes. When you have defined at least one name, you can click the empty area below the text box to add an additional text box in which you can enter further names.

You can use wildcards (? or *) within the names, if you do not want to enter each name individually. The question mark (?) may be specified at any position inside the name. The asterisk (*) is only allowed at the end of a name. Instead of typing the wildcards, it also is possible to enter them by choosing the button to the right of the text box (the button contains an arrow pointing to the right); in this case, you can see short explanations for the wildcards.

You can also enter a range of names. Ranges must be entered as follows:

```
name1 - name2
```

When defining a range, it is important that you enter a space before and after the hyphen. The spaces are necessary since names may contain hyphens. Without the spaces, `name1-name2` would be interpreted as the name of a single library or object/resource.

Each name in a range definition may contain wildcards (see above). Example:

```
AL* - AM?TEST
```

Or:

Choose the button to the right of a text box (the button contains an arrow pointing to the right) and then choose **Select from list**. You can now select all required libraries or objects/resources from a selection list which appears in an additional dialog box. When you press CTRL in the selection list, you can select multiple names; when you press SHIFT, you can select a range of names. When you choose the **OK** button, the dialog box containing the selection list is closed and the selected names are shown in the text box.

4 Choose the **OK** button.

The content of the library workspace, application workspace and of the corresponding list view window changes. Now, just the libraries or objects/resources that match your filter criteria are shown.

### To disable or delete a filter

1 Invoke the dialog box in which the filter has been defined as described above.

2 To disable the filter (without deleting all of the entries in the dialog box), activate the **Turn off filter** check box.

Or:

To delete just the entries in the currently selected text box, choose the **Delete** button.

Or:

To delete all entries in all text boxes, choose the **Delete all** button.

3 Choose the **OK** button.

The content of the library workspace, application workspace and of the corresponding list view window changes. When you have deactivated the filter or deleted all entries, all libraries or objects/resources are shown again. However, when you have just deleted one of several text boxes containing entries, just the libraries or objects/resources that match the new filter criteria are shown.

## Finding Objects in a Library

You can search, for example, for objects containing specific text or for objects with a specific user ID. It is also possible to replace or delete found text.

The found objects are shown in the **results window**. An active **filter** is not considered.

See also the description of the system command `SCAN`.

### To find objects

1 Optional. Select a node in the library workspace or application workspace, or select one or more entries in a **list view window**. This selection will be considered in the **Find Objects** dialog box.

2 From the **Library** menu, choose **Find Objects**.

Or:

Invoke the context menu and choose **Find Objects**.

Or:

When the Library toolbar is shown, choose the following toolbar button.

The **Find Objects** dialog box appears. This dialog box provides several pages:

- **Location**
- **Contents**
- **Advanced**

When a library, object or a specific object type was selected before invoking the **Find Objects** command, this selection is considered in the **Find Objects** dialog box.

3 Specify your search criteria on the different pages as described below.

Make sure to fill all required edit controls. Problems may occur, for example, if you are searching for the objects of a specific user and the text boxes **Names** and **Libraries** are blank; in this case, no objects will be found. If you do not want to specify a specific object name or library, specify an asterisk (*) in the corresponding text boxes.

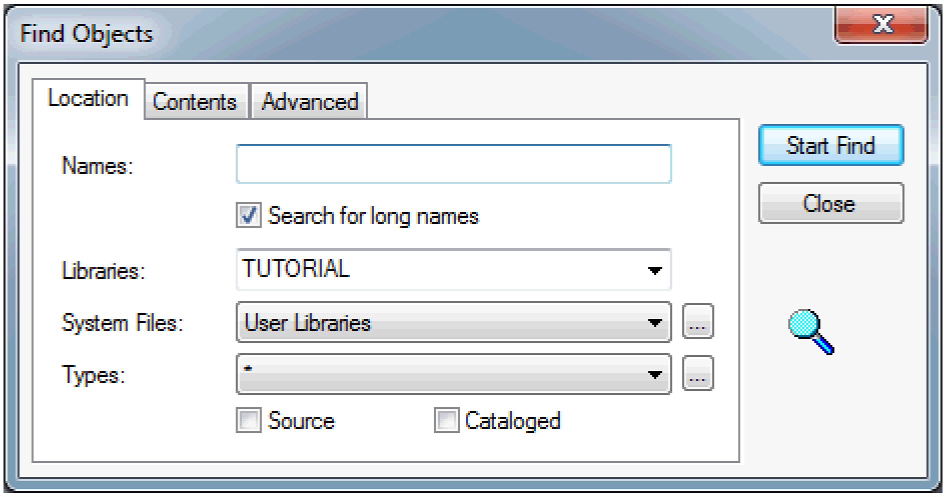
4 Choose the **Start Find** button.

The find process is started. Depending on the number of libraries and objects in the libraries, this may take a while.

A **Stop Find** button is available as long as the find process is running. You may choose this command button to abort the find process. All objects that have been found so far, are shown in the results window.

When the find process has completed, all found objects are shown in the results window. Any previous find results are overwritten. See ***Results Window*** for further information.

## Location



Names:

☑ Search for long names

Libraries: TUTORIAL

System Files: User Libraries

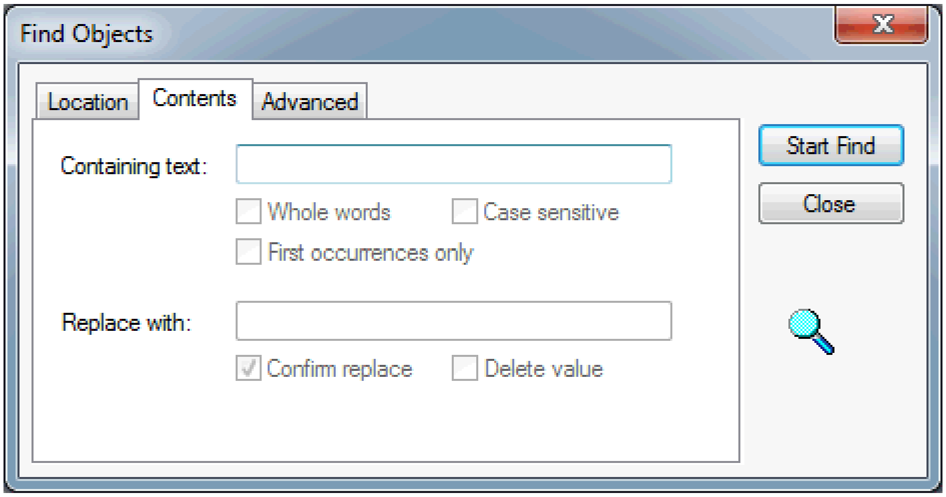
Types: *

☐ Source ☐ Cataloged

| Option | Description |
|---|---|
| **Names** | The names of the objects to be found. The content of this text box depends on the previous selection. If you specify more than one name, separate the names with semicolons. You can also use wildcards (* or ?). |
| **Search for long names** | If selected, the content of the **Names** text box is used to search for the long names of the objects. When the logical view was active before invoking the **Find Objects** dialog box, this check box is automatically selected; when the flat view or file view was active, this check box is not selected. |
| **Libraries** | The libraries in which to search for objects. The content of this drop-down list box depends on the previous selection. When a filter is active, only the libraries that match the filter criteria can be selected from the drop-down list box. However, you can specify the name of any other library. If you specify more than one library, separate the library names with semicolons. To specify all libraries, select the asterisk from the drop-down list box. You can also use wildcards with a library name (* or ?). |
| **System Files** | The system file to be used for the search. The content of this drop-down list box depends on the previous selection. In addition, the names in this drop-down list box depend on the current view (logical view, flat view or file view). Select the required system file from this drop-down list box. If you want to specify more than one system file, choose the button to the right of this drop-down list box and select the system files in the resulting dialog box. |
| **Types** | The types of objects (Natural objects and resources) to be included in the search. The content of this drop-down list box depends on the previous selection. Select the required type from this drop-down list box. If you want to specify more than one type, choose the button to the right of this drop-down list box and select the types in the resulting dialog box. To specify all types, select the asterisk from the drop-down list box. |
| **Source** | If selected, only Natural sources are found. |
| **Cataloged** | If selected, only Natural generated programs are found. |

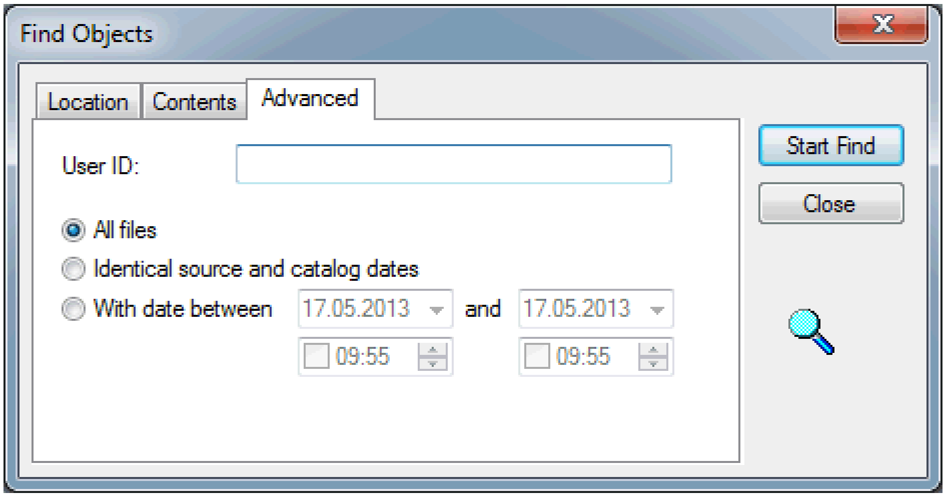
**Note:** If both the **Source** and the **Cataloged** check boxes are activated, both the source and the generated program of the object must exist in order for the object to be found.

### Contents



| Option | Description |
|---|---|
| **Containing text** | The text that is to be found. The edit controls on this page are only available when you have entered text in this text box. |
| **Whole words** | If activated, only whole words are taken into account. |
| **Case sensitive** | If activated, a distinction is made between uppercase and lowercase characters. |
| **First occurrences only** | If activated, the objects containing the specified text are only displayed once in the result window. |
| **Replace with** | If you want to replace the found text, specify the new text in this text box. |
| **Confirm replace** | Only available, if a replace text has been specified or if the **Delete value** check box has been activated. If activated, the **Confirm Replace** dialog box is shown when the specified text has been found. You can confirm each replace or delete operation separately, or you can confirm all replace or delete operations at once. |
| **Delete value** | Only available, when a replace text has not been specified. If you want to delete the found text, activate this check box. |

### Advanced



User ID:

● All files
○ Identical source and catalog dates
○ With date between 17.05.2013 and 17.05.2013
□ 09:55 □ 09:55

| Option | Description |
|---|---|
| **User ID** | The ID of the user who has created or modified an object. If specified, only objects with a matching user ID are found. If you specify more than one user ID, separate the IDs with semicolons. You can also use wildcards (* or ?). |
| **All files** | If selected, all specified Natural objects are searched. |
| **Identical source and catalog dates** | If selected, only those objects are found where the catalog date is identical with the source date. |
| **With date between** | If selected, only those objects are found where the last modification took place in the specified date and time range. You can specify the dates manually or select them from the drop-down list boxes. When you activate the check boxes for the dates, this information is also taken into account. You can specify a start time, an end time or both. You can specify the time either manually or with the spin boxes. |

## Suppressing Line Numbers in the Source Code

Normally, line numbers are written to the source code when you save an object to the file system. However, you can define that these line numbers are not written. This is important if you are storing your objects in a source control system. The suppression of line numbers is defined in the properties of a library (see also *Properties for the Nodes*).

When line numbers are suppressed in a library, the library icon that is shown in the tree looks as follows:

In addition to the local environment, you can also define the suppression of line numbers for the libraries in a remote Linux or Windows environment.

When line numbers are suppressed, any line number reference that is contained in the source code is replaced by a placeholder. The first line number reference is replaced by the placeholder "!0". For each further line number reference, the number following the exclamation mark (!) is incremented by 1. You cannot see these placeholders when you open a source in Natural Studio, you can only see them directly in the file system (remember that this is important for storing the objects in a source control system). When you open a source in Natural Studio, the line number references are always shown. It may happen that line number references have changed after reopening a source in Natural Studio, however, this will only happen when an irregular line number increment was previously used. When reopening a source for which line numbers are suppressed, the line numbers are always consistently incremented (just like after renumbering the lines).

**Notes:**

1. In the **program editor options**, you can specify whether the line numbers are to be shown or not.
2. The line number suppression state is stored in the file *FILEDIR.SAG*. The state can also be changed using the `FTOUCH` utility.

## Copying and Moving Libraries

Libraries are copied and moved in the same way as any other object in the library workspace (that is: with **Cut**, **Copy** and **Paste**, or with drag-and-drop). See ***General Information on Copying and Moving Objects*** for further information.

The target node of the copy or the move of a library can be any system file node (user libraries, system libraries or an inactive system file) in the local or remote environment. It can even be any other library node; in this case, all objects of the source library are copied.

When a library is copied or moved to a system file node for user libraries (`FUSER`), the new library must conform to the **library naming conventions**. Thus, when you copy or move a system library which starts with "SYS" to the user libraries, a dialog box appears and you have to specify another name. The default library name "USRLIB" that is provided in this dialog box can be overwritten.

The same applies when copying or moving a user library to the system libraries (`FNAT`) where the library names must start with "SYS". In this case, the dialog box mentioned above offers the default name "SYSLIB".

In all other cases, for example, when copying a system library to an inactive system file, the name of the source library is taken as the target library name.

## Renaming Libraries

A library is renamed in the same way as a Natural object. See ***Renaming Objects*** for further information.

When renaming a library, make sure to adhere to the **library naming conventions**. It is not possible to rename the library `SYSTEM`.

## Deleting Libraries

A library is deleted in the same way as a Natural object. See ***Deleting Objects*** for further information.

If you are working in a multiple-user environment, you should only delete a Natural library if you have exclusive access to the library involved. It is not possible to delete the library `SYSTEM`.

## Defining Your Own Startup Library

When you access Natural Studio for the first time, you log on to the library `SYSTEM` by default.

### To log on to a different library at startup

This description applies only when Natural Security is not active. Different rules apply if you logon to Natural under Natural Security; see *Logging On* in the *Natural Security* documentation.

1 Use the Configuration Utility to create your own Natural parameter file.

2 Select your new parameter file in the Configuration Utility.

3 Locate the parameter `INIT-LIB`.

> **Tip:** Locate this parameter by searching for "INIT-LIB". See *Finding a Parameter* in the *Configuration Utility* documentation for further information.

4 Specify the name of the startup library with the `INIT-LIB` parameter.

A startup library can be one of your own libraries or a library set up for you by your administrator.

5 Make sure that the parameter `AUTO` (which is defined on the same page as `INIT-LIB`) has been set to "ON". Otherwise `INIT-LIB` is not used.

6 Save your changes.

7 Invoke Natural Studio with your own parameter file. See *Invoking Natural with an Alternative Parameter File* in the *Configuration Utility* documentation.

## Example Library for New Features

The library `SYSEXV` contains several example programs which illustrate some of the new features of various Natural versions. See also *SYSEXV Utility* in the *Tools and Utilities* documentation.

# 6 Creating, Maintaining and Executing Natural Objects

The Natural system commands are used to create, maintain and execute Natural objects. In Natural Studio, menu commands are provided for several system commands. For example, the menu command **Check** corresponds to the Natural system command `CHECK`.

All system commands, including those for which a menu command is provided, can be entered in the command line. See ***Issuing Commands in the Command Line*** for further information.

This chapter describes the system commands that can be chosen from a menu.

Natural system commands can be executed on all platforms on which Natural is available. For detailed information on the available system commands, see the *System Commands* documentation.

**Note:** The system commands do not apply to resources. See ***Managing Resources***.

## Invoking Editors

An editor is invoked when you create a new Natural object or when you open an existing Natural object. Exception: an editor does not appear for classes (further information is provided in the topics below).

The following topics are provided:

- Creating a Natural Object
- Opening Existing Natural Objects

The corresponding system command for creating and opening a Natural object is `EDIT`.

For information on the programming language, see the *Programming Guide* and the *Statements* documentation.

See also *Object Locking* in the *Remote Development Using SPoD* documentation which contains information on the locking mechanism in the local and remote environment.

### Creating a Natural Object

A Natural object is always created in the library and environment that is currently indicated in command line.

#### To create a new Natural object

1 Make sure that the library in which you want to store the new object is selected.

2 From the **Object** menu, choose **New >** *`object-type`*. For a detailed description of each object type that is available with Natural, see *Objects for Natural Application Management* in the *Programming Guide*.

Instead of an object type, you can also choose **Dialog Wizard**.

Or:

Invoke the context menu and choose **New Source >** *`object-type`*.

Or:

When the Object toolbar is shown, choose the corresponding toolbar button (toolbar buttons are only available for some of the object types).

Or:

If you want to create a program, press CTRL+N (shortcut keys are not provided for creating other object types).

An empty editor window appears for the selected object type. Natural provides different editors for the different object types. For example, programs and subprogram are edited with the program editor, and the different types of data area (local, global and parameter data areas) are edited with the data area editor. See the *Editors* documentation for detailed information on how to use the different editors.

Exceptions:

- When you have chosen the **Dialog Wizard** command, a wizard appears. See *Dialog Wizard* in the *Dialog Editor* section of the *Editors* documentation for further information.
- When you create a DDM, a dialog appears first in which you have to select the database type. See *Creating DDMs* in the *DDM Editor* section of the *Editors* documentation for further information.
- Classes can only be created in the logical view and in the flat view. When you create a class, a new class with the default name "NEWCLS" is immediately shown in the tree. An editor is not invoked. Instead, you select the class and then choose the **New** command from the context menu to add further nodes (for example, an interface) to your class. As long as you do not save the class, it is not available in source format. See *Class Builder* in the *Editors* documentation for further information.

### To create a new Natural object by preselecting the object type

1. Select a group node in the logical view of the library workspace or in a list view window (for example, the **Programs** node).
2. Invoke the context menu and choose **New**.

   Since the type of object has already been determined by the previous selection, an empty editor window appears for this object type.

**Note:** When you select the **Classes** or **Resources** node or any group node within this node, and then choose **New** from the context menu, a new class or resource with a default name is added to the tree. An editor is not invoked in this case.

## Opening Existing Natural Objects

When you open a Natural object, it is opened in the corresponding Natural editor.

Exceptions:

- When you double-click a generated program or dialog for which a source does no longer exist, the program or dialog is executed. The **Open** command is not available in this case.
- When you open the node for a class, a list view window appears. To view the content of a class, you simply expand the nodes in the tree (only possible in the logical view and flat view).

The following description applies to all Natural objects except classes. It applies only to the source of an object (and not to the generated program which is visible in the file view).

### To open existing Natural objects

- Select the Natural object(s) that you want to open, invoke the context menu and choose **Open**.

  **Tip:** In a **list view window**, it is possible to select more than one object.

  Or:

  Double-click the Natural object.

  Or:

  When the Object toolbar is shown, select the Natural object(s) and choose the following toolbar button.

  📂

## Listing Objects

You can display the source code of an object in read-only mode. The object is displayed in the corresponding editor. "Object Listing" is shown in its title bar. You can copy its contents, but you cannot modify it. This is helpful, for example, if you want to view the content of an object in an inactive environment where it is not possible to open the object for editing.

**Note:** When you list a class, its code is shown in an editor window.

See also the description of the system command `LIST`.

### To list objects

1 Select the Natural object(s) that you want to list.

**Tip:** In a **list view window**, it is possible to select more than one object.

2 From the **Object** menu, choose **List**.

Or:

Invoke the context menu and choose **List**.

Or:

When the Object toolbar is shown, choose the following toolbar button.

## Checking Objects

You can check the source code of an object (except classes) for syntax errors.

**Note:** Syntax checking is also performed when you **run** or **stow** an object, or when you **catalog all** objects in a library.

See also the description of the system command `CHECK`.

### To check objects

1 Activate the editor containing the object that you want to check.

Or:

Select the object in the library workspace or list view window. When an editor window has not been opened, you can only use the toolbar button or shortcut key (see below).

**Tip:** In a **list view window**, it is possible to select more than one object.

2 From the **Object** menu, choose **Check**.

Or:

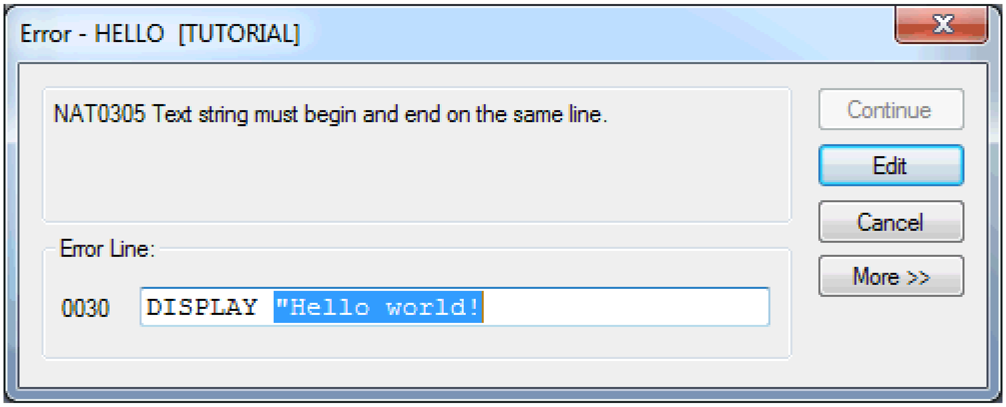
When the Object toolbar is shown, choose the following toolbar button.

Or:

Press CTRL+E.

If no errors are found, a message is shown indicating that the check was successful. The message is either shown in the status bar or in a dialog box. This depends on the setting for success messages in the **workspace options**.

If an error was found, a dialog box appears, providing information on the error. Example:



Error - HELLO [TUTORIAL]

NAT0305 Text string must begin and end on the same line.

Error Line:

0030 `DISPLAY "Hello world!`

3 Correct the error in the dialog box.

4 Choose the **Continue** button to find the next error.

When no more error is found, the dialog box is closed.

**Note:** Instead of choosing the **Continue** button, you can also choose the **Edit** button. The dialog box is then closed and you can correct the error directly in the program editor.

## Saving Objects

You can save the source form of a Natural object. The syntax is not checked.

**Note:** You can **run** a saved object. As long as you do not **stow** an object, you cannot **execute** it.

Natural Studio offers different commands for saving. They are described in the topics below:

- Saving an Object
- Saving an Object with a New Name
- Saving all Objects

When you close Natural Studio and your recent changes have not yet been saved, you are asked whether you want to save these changes.

See also the description of the system command `SAVE`.

### Saving an Object

You can save the source object in the active editor window.

Classes are handled differently since they are not edited in an editor window.

#### To save an object (except class)

1. Activate the editor containing the object that you want to save.
2. From the **Object** menu, choose **Save**.

   Or:

   When the Object toolbar is shown, choose the following toolbar button.

   💾

   Or:

   Press CTRL+S.

   **Note:** When you save an object for the first time, the **Save As** dialog box appears and you have to specify a name for the object. See below.

### To save a class

1 Select the top-level node of the class that want to save.

2 From the **Object** menu, choose **Save**.

Or:

Invoke the context menu and choose **Save**.

Or:

When the Object toolbar is shown, choose the following toolbar button.

💾

Or:

Press CTRL+S.

**Note:** When you save a class for the first time, the **Save As** dialog box appears and you have to specify a module name. See below.

## Saving an Object with a New Name

You can create a new object by copying the current contents of the editor window to a new object. The editor window for the original object is automatically closed.

For several object types it is also possible to select a different code page for the object. See the *Unicode and Code Page Support* documentation for further information.

**Caution:** If changes have been made since the last save, only the new object contains the changes. The original object is closed without saving the changes.

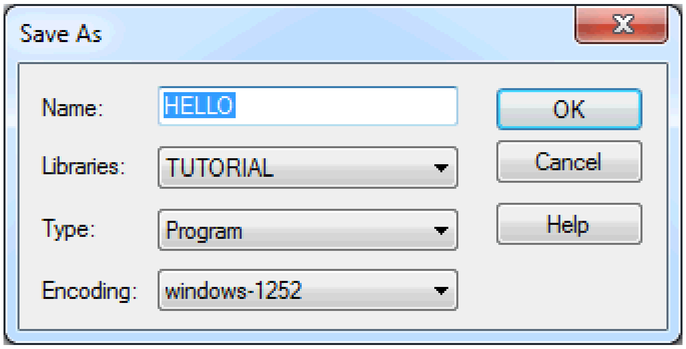
Classes are handled differently since they are not edited in an editor window. When a new class is created in the tree, it is not saved automatically. The class name which may be 32 characters long is shown in the logical view. The class module name, i.e. the name of the actual Natural object which may be up to 8 characters long, is assigned when the class is saved the first time; this name is not shown in the logical view. To find out the Natural object name for a class, you can display its **properties** or you can check it in the flat view or file view. See *Class Builder* in the *Editors* documentation for further information.

### To save an object (except class) with a different name

1 Activate the editor containing the object that you want to save under a different name.

2. From the **Object** menu, choose **Save As**.

   The **Save As** dialog box appears. Example for a program.



Name: HELLO

Libraries: TUTORIAL

Type: Program

Encoding: windows-1252

3. Specify a new name for the object.

   See also *Object Naming Conventions*.

4. Optional. Define a different library by selecting it from the **Library** drop-down list box.

5. Optional (if available for the current object type). Define a different object type by selecting it from the **Type** drop-down list box.

   The drop-down list box contains only those types that are allowed for the current object type. For example, you can change a program to a subprogram. See also *Changing the Type of a Natural Object*.

   **Note:** A type is not shown when you save a map, dialog or DDM.

6. Optional (if available for the current object type). Define a different code page by selecting it from the **Encoding** drop-down list box.

7. Choose the **OK** button.

   The new object is created. The name, library and object type of the new object are now shown in the title bar of the editor window.

   The new object is available in all views (sorted alphabetically). When a filter is currently active, it may happen that the new object is not shown since its name starts with different characters. See *Filtering Libraries and Objects* for further information.

### To save a class with a new module name

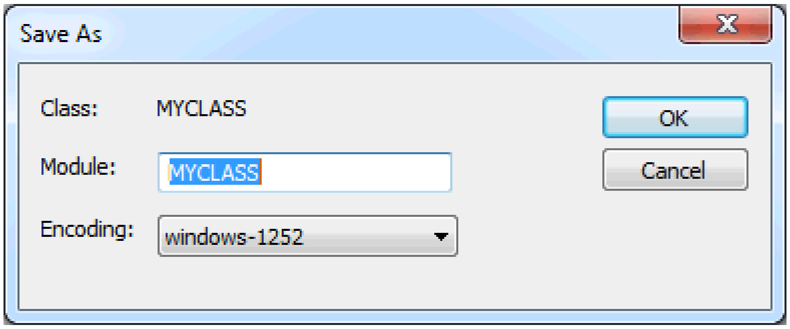
1. Select the top-level node of the class that you want to save under a different name.

2. From the **Object** menu, choose **Save As**.

   Or:

Invoke the context menu and choose **Save As**.

The **Save As** dialog box appears.



Save As

Class: MYCLASS

Module: MYCLASS

Encoding: windows-1252

3 Specify the new module name.

The dialog box displays the current module name for existing classes and the default module name for new classes. The module name may be up to 8 characters long.

4 Optional. Define a different code page by selecting it from the **Encoding** drop-down list box.

5 Choose the **OK** button.

## Saving all Objects

You can save all objects for which an editor window is open. You can also save all classes.

### To save all objects

- From the **Object** menu, choose **Save All**.

  Or:

  When the Object toolbar is shown, choose the following toolbar button.

  Or:

  Press CTRL+SHIFT+A.

  The objects are saved. When an object has not yet been saved previously, a **Save As** dialog box appears.

## Stowing Objects

When you stow an object, it is **checked** and compiled. If no errors are found, the source form of the object is saved and the resulting generated program is stored (in addition to the source).

**Note:** When you have stowed an object, you can **execute** it.

See also the description of the system command `STOW`.

### To stow objects

1 Activate the editor containing the object that you want to stow.

Or:

Select the object in the library workspace or list view window. When an editor window has not been opened, you can only use the context menu, toolbar button or shortcut key (see below).

**Tip:** In a **list view window**, it is possible to select more than one object.

2 From the **Object** menu, choose **Stow**.

Or:

Invoke the context menu for the selected object and choose **Stow**.

Or:

When the Object toolbar is shown, choose the following toolbar button.

Or:

Press CTRL+T.

When an object has not yet been saved previously, a **Stow As** dialog box appears. In this dialog box, you have to specify the same information as in the **Save As** dialog box.

## Cataloging Objects

When you catalog an object, it is **checked** and compiled. If no errors are found, the resulting generated program is stored.

In contrast to the **Stow** command, the source code is not saved when you catalog an object, for example, when the editor is open and the latest changes have not yet been saved. However, the latest changes in the editor are always considered during the compilation process even if they have not yet been saved.

See also the description of the system command `CATALOG`.

### To catalog objects

1 Select the object in the library workspace or list view window.

**Tip:** In a **list view window**, it is possible to select more than one object.

2 Invoke the context menu and choose **Catalog**.

Or:

When the Object toolbar is shown, choose the following toolbar button.

## Cataloging the Objects in a Library

You can catalog all objects or selected objects in a library. Instead of cataloging, you can also use this function to check, save or stow objects.

### To catalog the objects in a library

1 Select the library in which the objects are to be cataloged.

**Tip:** In a **list view window**, it is possible to select and catalog several libraries at the same time.

2 From the **Library** menu, choose **Cat All**.

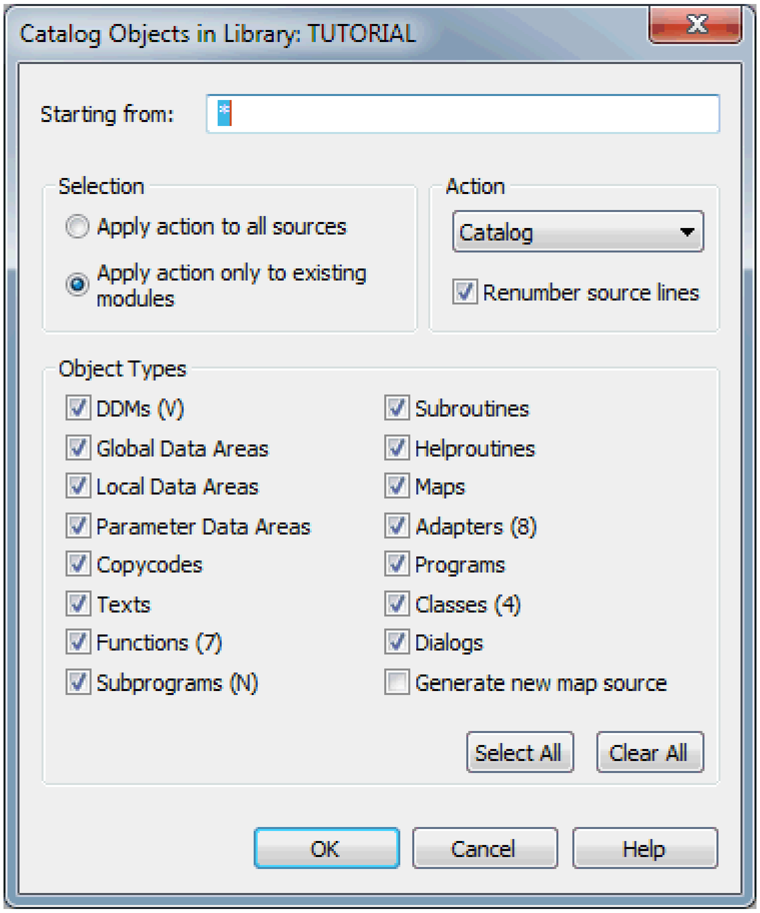
Or:

Invoke the context menu and choose **Cat All**.

Or:

When the Library toolbar is shown, choose the following toolbar button.

The following dialog box appears:



Catalog Objects in Library: TUTORIAL

Starting from: *

Selection

○ Apply action to all sources

● Apply action only to existing modules

Action

Catalog

☑ Renumber source lines

Object Types

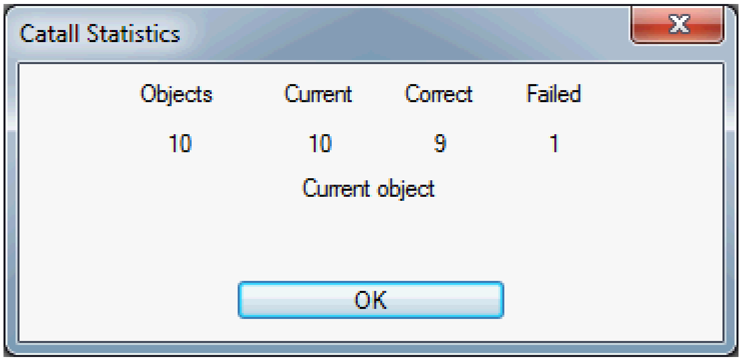
| | |
|---|---|
| ☑ DDMs (V) | ☑ Subroutines |
| ☑ Global Data Areas | ☑ Helproutines |
| ☑ Local Data Areas | ☑ Maps |
| ☑ Parameter Data Areas | ☑ Adapters (8) |
| ☑ Copycodes | ☑ Programs |
| ☑ Texts | ☑ Classes (4) |
| ☑ Functions (7) | ☑ Dialogs |
| ☑ Subprograms (N) | ☐ Generate new map source |

**Note:** When you are working in a remote development environment using SPoD, the options which cannot be used for this environment are dimmed.

3 Specify the types of objects and the action that is to be applied. See the description of the system command `CATALL` for detailed information on the options in this dialog box.

4 Choose the **OK** button.

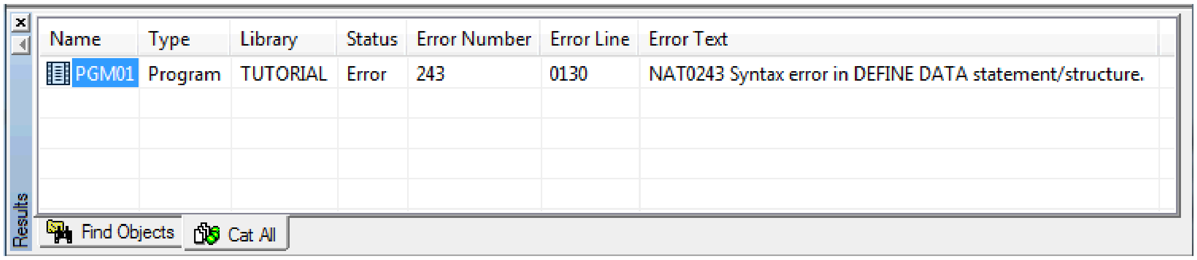
A dialog box appears providing statistics about the performed command. Example:



| Objects | Current | Correct | Failed |
|---|---|---|---|
| 10 | 10 | 9 | 1 |

Current object

If an object could not be cataloged, it is listed in the **results window**. Example:



| Name | Type | Library | Status | Error Number | Error Line | Error Text |
|---|---|---|---|---|---|---|
| PGM01 | Program | TUTORIAL | Error | 243 | 0130 | NAT0243 Syntax error in DEFINE DATA statement/structure. |

The objects are listed in the order in which they were processed. Associated with each object name is an error number (reason why the object could not be cataloged) and an error line (the line position in source code where the error can be found).

When you open an object from the results window, the line in which the error occurs is automatically selected in the editor.

**Note:** By default, only errors are listed in the results window. It is also possible to show information on all cataloged objects regardless if an error occurred or not. See *Workspace Options* for further information.

## Running Objects

You can run programs and dialogs. When you run such an object, it is **checked** and compiled on the fly. If no errors are found, the object is executed. A generated program is not stored.

The latest changes in the editor are always considered during the compilation process even if they have not yet been saved.

See also the description of the system command `RUN`.

### To run objects

1 Activate the editor containing the object that you want to run.

2 From the **Object** menu, choose **Run**.

Or:

When the Object toolbar is shown, choose the following toolbar button.

Or:

Press CTRL+R.

In the local environment, an output window appears when a Natural program writes output to the screen. See *Output Window* in the *Operations* documentation for further information.

In a remote development environment, either the Natural Web I/O Interface client or a terminal emulation window appears instead of an output window. See *Natural Web I/O Interface Client* and *Terminal Emulation* in the *Remote Development Using SPoD* documentation for further information.

## Executing Objects

You can execute programs and dialogs. When you execute such an object, it is **checked** and compiled. If no errors are found, the resulting generated program is stored and executed.

In contrast to the **Run** command which always uses your latest changes to the object, the **Execute** command always uses the stowed version of your object. The latest changes in the editor are not considered during the compilation process.

See also the description of the system command `EXECUTE`.

### To execute objects

1 Select the object in the library workspace or list view window.

**Tip:** In a **list view window**, it is possible to select more than one object. When you select several objects in a list view window, they are executed one after the other. The first object in the list view window is executed first.

2 From the **Object** menu, choose **Execute**.

Or:

Invoke the context menu and choose **Execute**.

Or:

When the Object toolbar is shown, choose the following toolbar button.

In the local environment, an output window appears when a Natural program writes output to the screen. See *Output Window* in the *Operations* documentation for further information.

In a remote development environment, either the Natural Web I/O Interface client or a terminal emulation window appears instead of an output window. See *Natural Web I/O Interface Client* and *Terminal Emulation* in the *Remote Development Using SPoD* documentation for further information.

## Clearing Editor Windows

When you clear an editor window, it is closed and a new empty editor window is opened at the same position. The title bar of the editor window shows that a name and library are not yet defined. The object type remains the same.

See also the description of the system command `CLEAR`.

### To clear an editor window

1 Activate the editor window that you want to clear.

2 From the **Object** menu, choose **Clear**.

Or:

When the Object toolbar is shown, choose the following toolbar button.

**Note:** When your latest changes have not yet been saved, a dialog box appears asking whether you want to discard these changes. If you do not want to discard the changes (**No** button), the clear process is stopped. You can then save the object and issue the **Clear** command once more.

# 7 Managing Natural Objects

## General Information on Copying and Moving Objects

This general information applies to the commands **Cut**, **Copy** and **Paste** and to drag-and-drop. Detailed information on how to **copy** and **move** objects is provided later in this section.

**Note:** The **Edit** menu which contains the above commands is only shown when an editor window is currently active in the workspace. However, these commands can always be chosen (with the restrictions described below) from the context menu, from the toolbar or using a shortcut key.

You can copy and move nodes from the local environment to the remote environment, and vice versa.

You can copy and move nearly all of the nodes in the library workspace and in the various list views. This can be either a group node or a node for a single object. For example, you can copy all programs of a library by copying the **Programs** node which is available in the logical view of the library workspace. Or you can move all generated programs of a library by moving the **Gp** node which is available in the file view of the library workspace.

The following restrictions apply:

- It is not possible to copy or move an object as long as the editor for this object is open.
- It is not possible to copy or move a system file node.
- The target node of a copy or move operation will only accept the objects of the selected source node when *all* objects can be copied or moved to the target node.

  Example: a list view window has been opened for a library in the flat view and you copy some of the objects. When you have also copied a Natural generated program, it is not possible to paste the objects on a **Src** node in the file view since the target node only accepts Natural source files. The **Paste** command is dimmed in this case.
- A group node cannot be copied *into* another group node. It can only be copied to a library node.

  For example, it is not possible to copy the **Programs** node of a library *into* the **Programs** node of another library. However, it is possible to copy the **Programs** node *onto* another **Programs** node or *onto* the library node itself.

When you copy, for a example, a program, it is not possible to paste it in the same library. If you want to create a copy, you have to save the object with a new name (see ***Saving an Object with a New Name***). However, it is possible to copy the contents of an entire library to the same system file. In this case, a dialog box appears prompting you to enter a new name for the library (see also *Copying and Moving Libraries*).

When you copy or move an object to another library in which an object of the type and with the same name exists already, and when you have activated the corresponding **workspace option**, a dialog box appears asking whether you really want to replace the existing object.

Since cut and copied data is stored on the Windows clipboard, you can also copy or move nodes to the Windows environment (**export**). Or you can copy or move, for example, an image from the Windows environment to a Natural library node; the image will then be available in a corresponding resource node (**import**).

**Caution:** It is recommended that you copy and move objects only from within Natural Studio. When you copy and move Natural files in the file system (for example, when using the Windows Explorer to move a file from one directory into another directory), the library information in the file *FILEDIR.SAG* is not updated and the files are therefore not shown in Natural Studio. To update *FILEDIR.SAG* in such a case, you have to use the `FTOUCH` utility.

## Copying Objects

You can copy the selected nodes by using the commands **Copy** and **Paste** or by using drag-and-drop.

When you are dragging nodes, the mouse pointer always indicates whether the copied objects can be dropped or not.

**Tip:** In a **list view window**, it is possible to select more than one node.

### To copy objects using menu commands

1 Select one or more nodes.

2 Invoke the context menu and choose **Copy**.

Or:

When the Edit toolbar is shown, choose the following toolbar button.

Or:

Press CTRL+C.

3 Select the target node.

4 Invoke the context menu and choose **Paste**.

Or:

When the Edit toolbar is shown, choose the following toolbar button.

Or:

Press CTRL+V.

### To copy objects using drag-and-drop - left mouse button

1. Select one or more nodes.
2. Click and hold down the left mouse button.
3. Drag the mouse to the node to which you want to copy the objects.
4. Hold down CTRL.
5. Release the mouse button and then CTRL.

**Caution:** If you do not hold down CTRL before releasing the mouse button, the objects are moved.

**Note:** The above description applies when the **workspace option** **Context menu for drag-and-drop with left mouse button** is not enabled (default). However, when you enable this option, you need not press CTRL. A context menu will then appear from which you can choose whether to copy or move the objects.

### To copy objects using drag-and-drop - right mouse button

1. Select one or more nodes.
2. Click and hold down the right mouse button.
3. Drag the mouse to the node to which you want to copy the objects.
4. Release the mouse button.
5. From the resulting context menu, choose **Copy Here**.

## Moving Objects

You can move the selected nodes by using the commands **Cut** and **Paste** or by using drag-and-drop.

When you are dragging nodes, the mouse pointer always indicates whether the cut objects can be dropped or not.

The selected nodes are deleted at their original position and copied to the new position.

**Tip:** In a **list view window**, it is possible to select more than one node.

### To move objects using menu commands

1 Select one or more nodes.

2 Invoke the context menu and choose **Cut**.

Or:

When the Edit toolbar is shown, choose the following toolbar button.

✂

Or:

Press CTRL+X.

3 Select the target node.

4 Invoke the context menu and choose **Paste**.

Or:

When the Edit toolbar is shown, choose the following toolbar button.

📋

Or:

Press CTRL+V.

### To move objects using drag-and-drop - left mouse button

1 Select one or more nodes.

2 Click and hold down the left mouse button.

3 Drag the mouse to the node to which you want to move the objects.

4 Release the mouse button.

**Notes:**

1. When the workspace option **Context menu for drag-and-drop with left mouse button** is enabled, a context menu will appear from which you can choose whether to copy or move the objects.
2. When you are working with SPoD and drag an object to a different environment using the left mouse button (for example, from the local environment to a remote Linux environment, or from a remote Linux environment to a remote z/OS environment), the object is copied and not moved. The object is not deleted at its original position.

### To move objects using drag-and-drop - right mouse button

1 Select one or more nodes.

2 Click and hold down the right mouse button.

3 Drag the mouse to the node to which you want to move the objects.

4 Release the mouse button.

5 From the resulting context menu, choose **Move Here**.

## Exporting Objects

You can export a node from the Natural environment to the Windows Explorer. This is helpful, for example, when you want to send one of your Natural source objects to an other user via e-mail.

To export a node, you use the **Copy** or **Cut** command of Natural Studio as described above and the **Paste** command of the Windows Explorer. You can also use drag-and-drop.

When you drag a node to the Windows environment using the *left* mouse button, it is always copied (it is not necessary to press CTRL in this case). When you use the *right* mouse button for drag-and-drop, you can select the appropriate command for copying or moving from a context menu.

When you export a group node (for example the **Programs** node which is available in the logical view), all sources and generated programs are copied or moved. When both source and generated program are available for an object, two files will then be available in the Windows environment. The extension indicates the type of file. Example: "NSP" stands for Natural source program and "NGP" stands for Natural generated program.

## Importing Objects

You can import files from the Windows Explorer to the Natural environment. This is helpful, for example, when you have received a Natural source object via e-mail.

To import files, you use the **Copy** or **Cut** command of the Windows Explorer and the **Paste** command of Natural Studio as described above. You can also use drag-and-drop.

When you drag a file to the Natural environment using the left mouse button, the file is always copied (it is not necessary to press CTRL in this case).

When you select a folder name in the Windows Explorer and then import this folder, only the content of the folder is imported. Subfolders are not imported. To import the content of a subfolder, either select the subfolder name or display its contents and select the required objects. When an object is imported which has a file extension unknown to Natural, it is stored as a resource.

**Important:** Before the import is started, it is important to set the proper programming mode (structured mode or reporting mode) in which the Natural objects to be imported were developed, otherwise some of them might not be compiled. To set the programming mode, you can either set the compiler option `SM` as described in ***Using Session Parameters***, or you can enter the system command `GLOBALS` with the following values in the command line: `GLOBALS SM=ON` (structured mode) or `GLOBALS SM=OFF` (reporting mode). See also *Natural Programming Modes* in the *Programming Guide*.

## Renaming Objects

A subroutine cannot be renamed in the logical view. This is because the logical view always shows the subroutine name that has been defined in the program (which may exceed 8 characters). The flat view and file view show the name that has been defined when saving the subroutine; this can be renamed.

**Tip:** It is recommended that you rename objects in the logical view. When you rename, for example, a source in the file view for which a generated object exists, only the source is renamed; a new generated program with the new name is not created. The existing generated program is not renamed.

**Note:** You can also rename an object by entering the system command `RENAME` in the command line. In this case, a dialog box appears in which it is also possible to change the type.

See also ***Renaming Libraries***.

### To rename an object

1 Select the node that is to be renamed.

**Note:** Only one object can be renamed at a time. If several objects are selected in a **list view window**, the **Rename** command is only applied to the object which has the focus.

2 From the **Object** menu, choose **Rename**.

Or:

Invoke the context menu and choose **Rename**.

Or:

Press F2.

The name is selected so that you can immediately enter a new name. Any text you enter automatically deletes the selection.

**Note:** If you want to cancel your modifications, press ESC.

3 Enter a new name for the object.

See also ***Object Naming Conventions***.

**Note:** When the name is in edit mode, you can invoke a context menu. In addition to the standard Windows editing commands such as **Cut** and **Paste**, the context menu may also contain Windows commands for Unicode. See your Windows documentation for further information.

4 Press ENTER.

Or:

Click any other position in the library workspace or list view window.

When a filter is currently active, it may happen that the renamed object is not shown since its name now starts with different characters. See ***Filtering Libraries and Objects*** for further information.

## Changing the Type of a Natural Object

You can change the type of any Natural object except a DDM, map, or dialog. For the remaining objects, the following restrictions apply:

- A data area can be converted only to another type of data area: local, global or parameter.
- All other Natural objects can be converted only to another type of Natural object: program, subprogram, subroutine, helproutine, function, copycode or text.

**Note:** You can also change the object type by entering the system command `RENAME` in the command line. In this case, a dialog box appears in which it is also possible to rename the object.

### To change the object type

1. Open the object so that it is shown in an editor window. See *Opening an Existing Natural Object*.
2. Make sure that the editor window is the active window.
3. From the **Object** menu, choose **Object Type >** *`object-type`*.

   Only the allowed object types are available from the menu.

   The new type is indicated in the title bar of the editor window.
4. **Save** the object to make the conversion complete.

## Changing the Code Page of a Natural Object

The code page is defined in the properties of an object. If available for the selected object type, you can invoke an **Encoding** dialog box from the **Properties** dialog box. See *Properties for the Nodes* for further information.

Changing the code page of an object means that the encoding information which is stored in the file *FILEDIR.SAG* is changed; it does not mean that the contents of the object is converted in any way. If you want to convert an object to a different code page, you have to open the object and save it with the **Save As** command; you can then select the desired encoding in the **Save As** dialog box (see *Saving an Object with a New Name*).

For further information on code pages, see the *Unicode and Code Page Support* documentation.

## Deleting Objects

When you delete an object in the logical view or flat view, both source and generated program are deleted.

You can delete, for example, all programs in a library by deleting the **Programs** node in the logical view. Or you can delete only the generated programs in a library by deleting the **Gp** node in the file view.

It is not possible to delete an object when the editor window for this object is currently open.

See also ***Deleting Libraries***.

**Note:** You can also delete objects by entering the system command `SCRATCH`, `PURGE` or `UNCATALOG` in the command line. A dialog box appears for each of these system commands and you can select the objects to be deleted.

### To delete objects

1 Select one or more nodes.

**Tip:** In a **list view window**, it is possible to select more than one node.

2 From the **Object** menu, choose **Delete**.

Or:

Invoke the context menu and choose **Delete**.

Or:

Press DEL.

When the corresponding **workspace option** has been set, you are prompted to confirm the deletion.

## Printing Objects and Views

There are two different types of print command. You can either print an object or the content of a view. The object or view is printed on a printer that is defined in Windows.

The layout of the printed page is defined in the **Options** dialog box. See ***Printing Options*** for further information.

The following topics are covered below:

- Printing Objects
- Printing Views

### Printing Objects

When you print an object, its source code is printed. In the case of an object which has a graphical representation (such as a dialog or map), the source code as generated by Natural is printed.

If the source is partially collapsed in the program editor (see *Showing and Hiding Source Code* in the *Program Editor* section of the *Editors* documentation), the hidden lines are not printed.

**Note:** The description below does not apply to resources. See ***Managing Resources***.

#### To print objects

1 Activate the editor containing the object that you want to print.

Or:

Select the object in the library workspace or application workspace.

Or:

Select one or more objects in a **list view window**.

2 From the **Object** menu, choose **Print**.

Or:

Invoke the context menu and choose **Print** (not possible in the editor window).

Or:

When the Object toolbar is shown, choose the following toolbar button.

🖨

Or:

Press CTRL+P.

The **Print** dialog box appears.

3 Specify all required information (for example, a different printer or text orientation) and choose the **Print** button.

## Printing Views

You can print the contents of tree view windows and list view windows.

- **Tree Views**
  A tree view is printed graphically. The printout is a copy of the tree view. The font will appear as on the screen; it cannot be changed.

  You can print, for example, the current layout of the library workspace. When a node is expanded in the tree view, it is also expanded on the printout; when a node is not expanded in the tree view, it is not expanded in the printout.

  When a node is currently selected in the tree view, this node is taken as the starting point for the printout; the nodes above are not contained in the printout.
- **List Views**
  A list view is printed as normal text in column format. Therefore, a different font can be defined in the **printing options**.

  When one or more group nodes are selected in the list view window, the list views for these nodes are printed (that is, the list views that are normally displayed when you open such a node). A separate printout is then available for each selected group node. When a node is not selected, the current list view is printed.

  You can suppress printing of specific columns in the current list view. To do so, resize the column in the list view window so that it is no longer visible. See also ***List View Windows***.

  The current width of a column is not considered. The printout always contains the entire text in a column.

### To print a tree view or list view

1 Activate the window for which you want to print a view (either a tree view window or a list view window).

2 Optional. Select a node if the printout is to contain the output described above.

3 From the **View** menu, choose **Print View**.

In a tree view window, it is also possible to choose command from the context menu, but only when a node is not selected.

The **Print** dialog box appears.

4 Specify all required information (for example, a different printer or text orientation) and choose the **Print** button.

## Refreshing the Display

Usually when something changes in Natural Studio, the display is automatically refreshed. This happens, for example, when objects are created, renamed or deleted. This automatic refresh requires that the corresponding option has been set in the **workspace options**.

There are, however, situations where an automatic refresh does not take place, since Natural is not aware of a modification. For example, two Natural processes are currently active and both are working on the same system file. When one Natural is applying a change to the system file (such as creating a new object), the second Natural is not aware of this modification. In this case, you have to refresh the display manually.

A manual refresh can only be performed on a selected system file node or library node (in the library workspace, application workspace or in a list view window). It can be performed in both the local and remote environment.

### To refresh the display manually

1 Select the system file node or library node to be refreshed.

2 From the **View** menu, choose **Refresh**.

Or:

From the context menu, choose **Refresh**.

Or:

Press F5.

# 8 Managing Resources

Resources are used with dialogs. A resource is a non-Natural file (for example, an image file or a Word document) that has been created in the Windows environment. For more information, see *Resource* in the *Object Types* section of the *Programming Guide*.

**Note:** The information in this section does not apply to the so-called “private” resources which are created when using ActiveX controls in a Natural dialog.

Resources can be created and edited in Natural Studio. Using copy-and-paste or drag-and-drop, they can also be imported/exported from/to the Windows Explorer. When you invoke the context menu for a resource, you will notice that Natural Studio provides commands that are also used for this type of file in the Windows Explorer. Using these Windows commands, you can manage (open, copy, print etc.) the resources in Natural Studio.

For example, when a print command is available for an image file, a Windows command is executed (not a Natural Studio command which may have the same name). The associated program is opened, the print command is automatically executed and the program is automatically closed. The resource is printed on the printer that has been defined in Windows.

The **Properties** command is an exception. Other than the Windows command with the same name, it shows only Natural-specific information: name, library, type, size, creation date and date of last access. See also ***Properties for the Nodes***.

### To create a resource in Natural Studio

1 Make sure that the library in which you want to store the new resource is selected.

2 Invoke the context menu and choose **New Resource >** `resource-type`.

The resource types that are listed in the cascading menu are those which are registered in your Windows environment.

A new resource with a default name (for example, "New Bitmap Image" when you have chosen the corresponding command from the context menu) is now shown in the tree.

The icon shown for a resource is the same as that used in Windows.

3 Enter a name for the resource. Do not specify an extension. The correct extension will be used automatically when this resource is opened.

4 Press ENTER.

Or:

Click any other position in the library workspace.

Your new resource is now available in all views (sorted alphabetically). As long as you do not edit it as described below, it is empty.

### ≫ To open and edit a resource in Natural Studio

1 Select the resource that you want to open, invoke the context menu and choose the Windows command that is used to invoke the corresponding program (**Open** or **Edit** in many cases).

Or:

Double-click the resource.

The resource is opened in the associated program.

**Note:** When an association for this program has not yet been defined in Windows, a dialog box appears and you have to specify the program with which this file is to be opened. This is standard Windows functionality.

2 Edit the resource (for example, draw a bitmap), save your changes and close the program.

If the buffer pool synchronization is enabled for the associated buffer pool, all of your changes will be visible the next time you open this resource from Natural Studio.

**Note:** The buffer pool synchronization is enabled/disabled using the Configuration Utility. See *Buffer Pool Assignments* in the *Configuration Utility* documentation. The modified buffer pool assignments are only available after restarting either the buffer pool service or the entire system.

# 9 Managing the Different Types of Windows

**Note:** For information on the output window, which appears in the local environment when a Natural program writes output to the screen, see *Output Window* in the *Operations* documentation.

## List View Windows

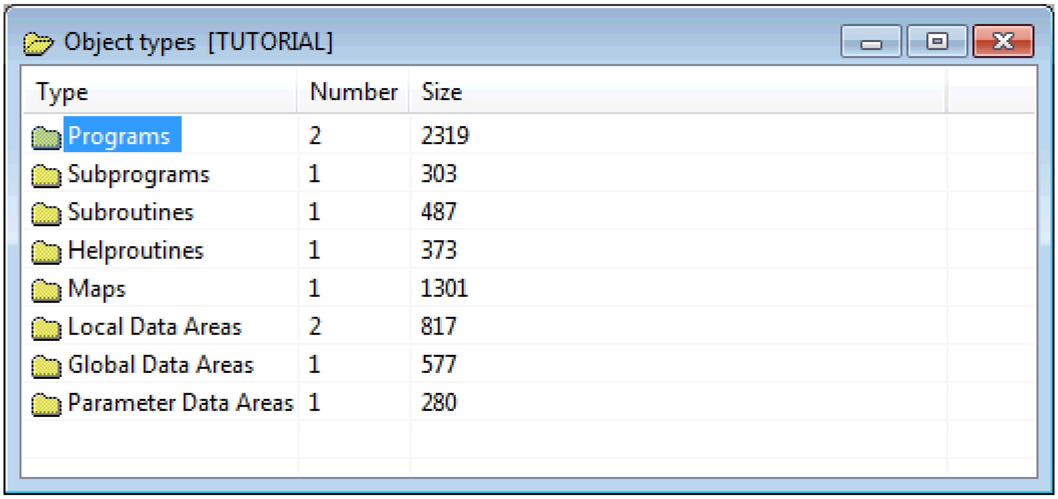
The following topics are covered below:

- Opening a List View Window
- Selecting More than One Node
- Defining a Different View
- Resizing a Column
- Repositioning a Column
- Sorting a Column
- Defining a Different Layout for the Date

### Opening a List View Window

You can open a list view window for each group node in a tree view window.

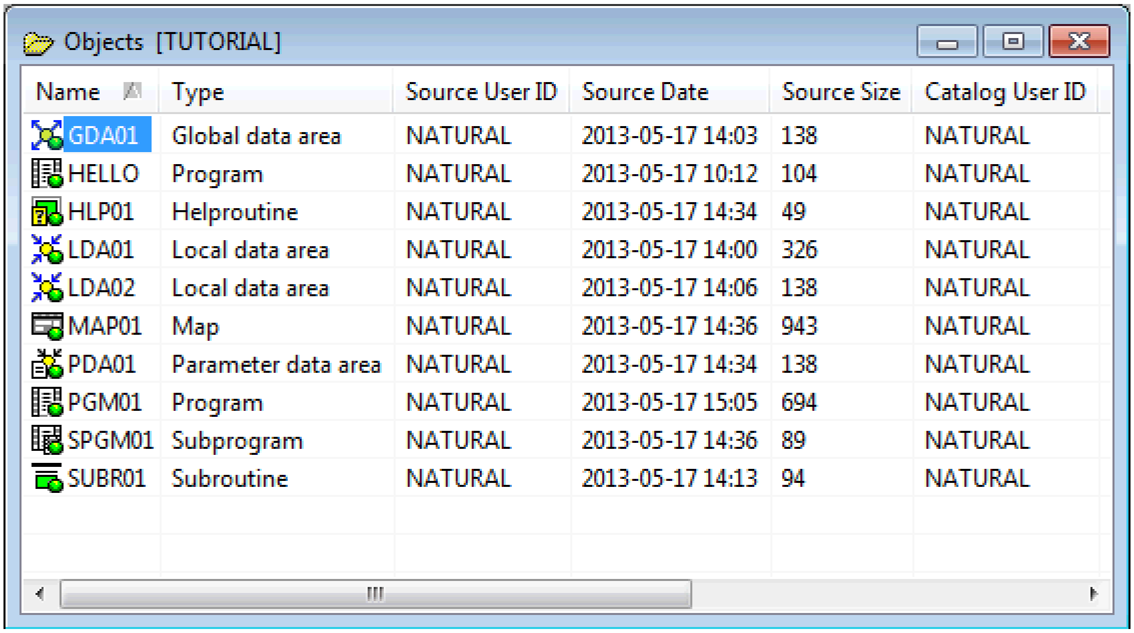
A list view window always reflects the view from which it was opened. For example, when you open a library node in the logical view, the following information is shown:



Object types [TUTORIAL]

| Type | Number | Size |
|---|---|---|
| Programs | 2 | 2319 |
| Subprograms | 1 | 303 |
| Subroutines | 1 | 487 |
| Helproutines | 1 | 373 |
| Maps | 1 | 1301 |
| Local Data Areas | 2 | 817 |
| Global Data Areas | 1 | 577 |
| Parameter Data Areas | 1 | 280 |

When you open the same node in the flat view, the following is shown:



Objects [TUTORIAL]

| Name | Type | Source User ID | Source Date | Source Size | Catalog User ID |
|---|---|---|---|---|---|
| GDA01 | Global data area | NATURAL | 2013-05-17 14:03 | 138 | NATURAL |
| HELLO | Program | NATURAL | 2013-05-17 10:12 | 104 | NATURAL |
| HLP01 | Helproutine | NATURAL | 2013-05-17 14:34 | 49 | NATURAL |
| LDA01 | Local data area | NATURAL | 2013-05-17 14:00 | 326 | NATURAL |
| LDA02 | Local data area | NATURAL | 2013-05-17 14:06 | 138 | NATURAL |
| MAP01 | Map | NATURAL | 2013-05-17 14:36 | 943 | NATURAL |
| PDA01 | Parameter data area | NATURAL | 2013-05-17 14:34 | 138 | NATURAL |
| PGM01 | Program | NATURAL | 2013-05-17 15:05 | 694 | NATURAL |
| SPGM01 | Subprogram | NATURAL | 2013-05-17 14:36 | 89 | NATURAL |
| SUBR01 | Subroutine | NATURAL | 2013-05-17 14:13 | 94 | NATURAL |

When a list view window does not show all defined objects, a filter has been applied. See *Filtering Libraries and Objects* for further information.

### To open a list view window

1 Select a group node in the tree view window.

Or:

In an already open list view window, select a node for which a folder icon is shown.

2 From the **Object** menu, choose **Open**.

Or:

Invoke the context menu and choose **Open**.

Or:

Press CTRL+O.

Or:

Press ENTER.

### Selecting More than One Node

In a list view window, it is possible to select several nodes at the same time (which is not possible in a tree view window). This is helpful, for example, if you want to delete several objects at the same time.

#### ≫ To select several nodes in a row

1. Select the first node.
2. Press SHIFT.
3. Use the mouse to select the last node that is to be part of the selection.

   Or:

   Use the UP-ARROW or DOWN-ARROW key to select all nodes that are to be part of the selection.

#### ≫ To select individual nodes

1. Select the first node.
2. Press CTRL.
3. Use the mouse to select each node that is to be part of the selection.

   Or:

   Use the UP-ARROW or DOWN-ARROW key to move to the node that is to be part of the selection and then press the SPACEBAR. Repeat this until all required nodes are selected.

#### ≫ To select all nodes

- Invoke the context menu and choose **Select All**.

  Make sure that the mouse pointer is not positioned on the name of an entry in the list. Otherwise, a different context menu appears.

  Or:

  Press CTRL+A.

  Or:

  When the Edit toolbar is shown, choose the following toolbar button.

### Defining a Different View

By default, the content of a list view window is shown in details view. You can also define another view (for example, large icons).

#### To define another view

1 Activate the list view window.

2 Invoke the context menu and choose one of the following commands from the **View** menu: **Large Icons**, **Small Icons**, **List** or **Details**.

Make sure that the mouse pointer is not positioned on the name of an entry in the list. Otherwise, a different context menu appears.

The current type of view is always indicated by a check mark.

### Resizing a Column

In details view, you can use the mouse to change the width of each column. This change is only valid as long as the list view window is open.

You can also resize a column in such a way that it is not longer visible. If you want to redisplay the column, close the list view window and then open it once more. Or if you still know at which position the column is hidden, move the mouse pointer to this position in the header and resize it as described below.

#### To resize or hide a column

1 Move the mouse pointer to the line in the header which separates two columns.

The pointer changes, showing two arrows pointing into opposite directions.

2 Use the mouse to drag the column until it has the desired width or until it is no longer visible.

### Repositioning a Column

In details view, you can use the mouse to move a column to a different position in the list view window. This change is only valid as long as the list view window is open.

**To reposition a column**

- Drag the header of a specific column and drop it onto a different header.

### Sorting a Column

In details view, you can use the mouse to sort the columns in ascending or descending order.

It is, for example, possible to click on the **Extension** column header to sort the column in ascending order and then hold the CTRL key down and click on the **Name** column to sort it in descending order (inside the sort sequence of the column first sorted).

**To sort a column**

- Click the header of a column.

  An arrow in the header indicates the sort direction. With each click on the same header, the column in sorted in the opposite direction.

### Defining a Different Layout for the Date

Using the Configuration Utility, you can change the Natural profile parameter `DTFORM` which determines the format in which the dates are represented in details view.

## Arranging and Activating List View Windows and Editor Windows

When at least one list view window or editor window is open, the **Windows** menu is available.

**Note:** How to open an editor window is explained in the section ***Invoking Editors***.

Using the commands in the **Windows** menu, you can arrange or activate the list view windows and editor windows in the workspace. The following commands are available:

| Command | Shortcut key | Description |
|---|---|---|
| **Next** | CTRL+TAB | Activate the next inactive window. |
| **Previous** | CTRL+SHIFT+TAB | Activate the previous inactive window. |
| **Close All** | | Close all open windows. When the changes in an editor window have not yet been saved, a dialog box appears, asking whether you want to save the changes. |
| **Cascade** | | Arrange all windows so that they overlap. |
| **Tile Horizontally** | | Arrange all windows horizontally so that they do not overlap. |
| **Tile Vertically** | | Arrange all windows vertically so that they do not overlap. |
| **Arrange Icons** | | Arrange the icons of minimized windows in a row at the bottom of the visible workspace. |
| *Name of open window* | | Commands for the open windows are shown at the bottom of the menu. The window which is currently active is indicated by a check mark. To activate a window, choose the corresponding command. |

You can also use the following shortcut keys:

| Shortcut key | Description |
|---|---|
| CTRL+D | Toggle between the command line and the active list view or editor window. |
| CTRL+L | Toggle between the application workspace and the active list view or editor window. |
| CTRL+U | Toggle between the result window and the active list view or editor window. |
| CTRL+W | Toggle between the library workspace and the active list view or editor window. |

In addition to the standard Windows techniques for closing a window, you can also close the active list view window or editor window as described below.

### To close the active window

- From the **Object** menu, choose **Close**.

  Or:

  Press CTRL+F4.

  Or:

  When the Object toolbar is shown, choose the following toolbar button.

## Dockable Windows

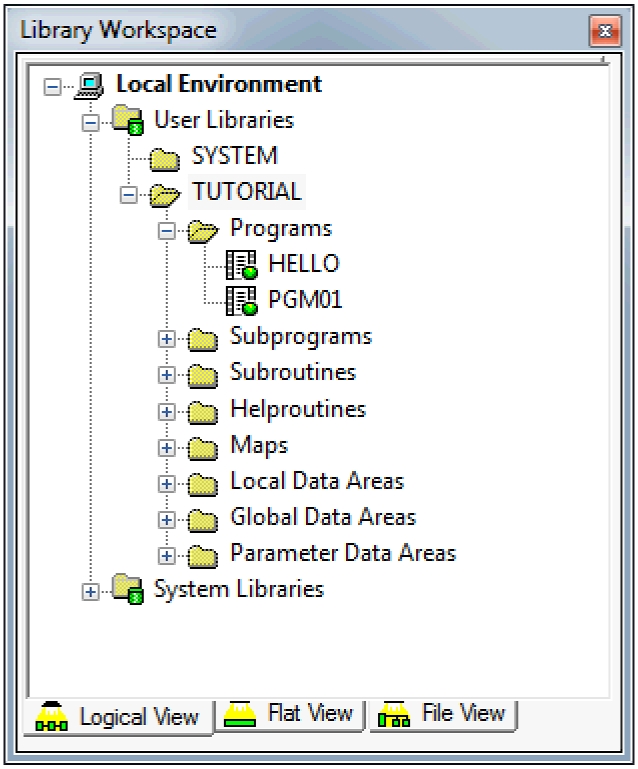
The following elements of the Natural Studio window are dockable windows:

- menu bar
- toolbars
- library workspace
- application workspace
- results window
- command line
- debugger windows

**Note:** The commands in the **Windows** menu do not apply to dockable windows.

Using the mouse, you can drag a dockable window to another position so that it is shown

- at another position within the Natural Studio window, or
- in a window of its own. For example:

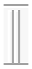


You can move the window freely on your screen. You can move it back to the Natural Studio window (for example, back to its original position) so that it is no longer shown in a window. This process is called "docking".

### To enable/disable docking for a window

1 When the library workspace, application workspace or results window is currently docked, move the mouse pointer to title bar of the window.

When the command line, menu bar or a toolbar is currently docked, move the mouse pointer to the two lines which are shown with the docked window. For example, when docked horizontally, the two lines are shown as follows to the left of the docked window:



Or:

When the window is currently undocked, move the mouse pointer to the title bar of the window.

2 Invoke the context menu.

When a check mark is shown next to **Enable docking**, the window can be docked. When a check mark is not shown, the window cannot be docked.

3 If you want to toggle the docking status, choose **Enable docking**.

> **Note:** When you disable docking for a window which is currently docked, the window is undocked.

### To undock/dock a window using the mouse

This requires that docking is enabled (see above).

- Double-click the title bar of a docked window, or double-click the two lines which are shown for specific types of docked windows (see above).

  Or:

  Double-click the title bar of an undocked window.

  When the window is currently docked, it is undocked. When the window is currently undocked, it is docked at its previous position.

### To drag a docked window to another position

This requires that docking is enabled (see above).

1 Move the mouse pointer to the title bar of a docked window or to the two lines which are shown for specific types of docked windows (see above).

2 Press the mouse button and drag the window to another position.

An outline of the window is shown. The outline indicates the position at which the window can be docked.

3 If you want to prevent docking (that is, if you want to undock the window), press CTRL while dragging the window.

4 Release the mouse button.

### To drag an undocked window back to the Natural Studio window

This requires that docking is enabled (see above).

1 Move the mouse pointer to title bar of an undocked window.

2 Press the mouse button and drag the window to the position at which you want to dock it.

An outline of the window is shown. The outline indicates the position at which the window can be docked.

3 Release the mouse button.

### To expand a docked window

Applies only to the library workspace, application workspace, results window and debugger window. This feature is only functional when two of these elements are docked next to each other, either horizontally or vertically.

1. In the title bar of the docked window, choose the button which contains an arrow.

   The arrow on the button points either upwards or to the left, depending on the type of window. When you move the mouse over this button, the tooltip "Expand docked window" appears.

   The docked window is expanded so that it hides the other window. It now occupies the space which both windows together have previously occupied. You can see that a window is hidden because the title bar of the hidden window is still visible.

   The arrow which is shown on the button now points to the opposite direction. When you move the mouse over this button, the tooltip "Contract docked window" appears now.

2. To contract the window so that both windows are shown again, choose the above button once more.

# 10 Issuing Commands in the Command Line

Initially, the command line is not shown. For information on how to display it, see ***Command Line*** in the section *Elements of the Natural Studio Window*.

## What Can be Entered in the Command Line

You can execute the following directly from the **Command** drop-down list box which is part of the command line.

- **Natural System Commands**
  In most cases, a system command can be issued using the menu bar, a context menu or a toolbar button. However, there are scenarios where applying commands with the command line is useful. This is especially the case for commands which do not depend on the current context. An example of such a command is `GLOBALS XREF=ON` which is used to set global settings (generation of XRef data for Natural). An example of a system command which does depend on the context is `STOW` which acts on the current object (this can be an active editor window or the current selection in the library workspace).

  See also ***Asterisk Notation*** below.

- **Natural Utilities**
  You can invoke Natural utilities which are available for the Windows environment but for which a menu command is not provided in Natural Studio. For example, when you enter "SYSMAIN" in the command line, the dialog box for this utility appears (see the *Tools and Utilities* documentation for further information).

  When working in a remote environment on a development server (only available in a SPoD environment), you can invoke utilities that can only be used in this environment and for which a graphical user interface is not provided in Natural Studio. `SYSBPM` is an example of such a utility (see the Natural for z/OS documentation for further information). When you invoke this utility, either the Natural Web I/O Interface client or a terminal emulation window is used to show its character-based screen. See *Natural Web I/O Interface Client* and *Terminal Emulation* in the *Remote Development Using SPoD* documentation for further information.

- **Generated Programs and Dialogs**
  You can enter the name of a generated program or dialog in the command line. When it can be found in the current library, it is executed.

## Command Execution

Natural Studio saves each character string you enter in the command line for the current session. The drop-down list box contains your last entries (history).

### To issue a command in the command line

1 Type the command in the text box of the **Command** drop-down list box.

Or:

When the mouse pointer is positioned on the **Command** drop-down list box, you can also use the right mouse button to invoke a context menu. Using the commands from this context menu, you can, for example, paste a text string into the command line.

**Note:** In addition to the standard Windows editing commands such as **Cut** and **Paste**, the context menu may also contain Windows commands for Unicode. See your Windows documentation for further information.

2 Press ENTER to execute the command.

### To issue a command using the command line history

1 Open the **Command** drop-down list box and select an entry.

Or:

Enter the first character of a command that you have previously entered in the text box of the **Command** drop-down list box. The corresponding command automatically appears in the text box. When you have previously entered several commands which start with the same letter(s), the command that has been entered last appears in the text box.

2 Press ENTER to execute the command once more.

## Asterisk Notation

Many Natural functions display lists of objects. Usually these lists contain all objects available (for example, all objects of a given type, all objects in a given library, etc.). If you do not wish all objects to be listed, but only a certain range of objects, you may specify that range by using asterisk notation.

By specifying a parameter value followed by an asterisk (*) you can get a list of only those objects whose names (or IDs or whatever the parameter is) begin with that value. This option to enter a value followed by an asterisk is referred to as asterisk notation.

**Example 1**

If you enter the system command `SCRATCH` without any parameters:

```
SCRATCH
```

you will get a list of all objects in the current library. You can then select the objects which are to be deleted.

**Example 2**

If you enter the system command `SCRATCH` as follows:

```
SCRATCH BOC*
```

you will get a list of only those objects in the current library whose names begin with "BOC". You can then select the objects which are to be deleted.

# 11 Customizing Natural Studio

This chapter explains how to change Natural Studio's command, toolbar and shortcut key assignments. For example, you can add commands to Natural Studio's menu bar or toolbar (including user-defined commands) or you can define different or additional shortcut keys. You can thus adapt Natural Studio's user interface to your requirements.

## Invoking the Customize Dialog Box

Many changes in the **Customize** dialog box are immediately reflected in the Natural Studio window. For example, when you switch the display of a specific toolbar on, this toolbar is immediately shown in the Natural Studio window; it is not necessary to apply this change. It is also possible, for example, to drag a command from the **Customize** dialog box to a menu in the Natural Studio window.

### To invoke the Customize dialog box

1 From the **Tools** menu, choose **Customize**.

The **Customize** dialog box appears. Different pages are provided for different types of customization.

2 Select one of the pages in this dialog box and proceed as described below under the following headings.

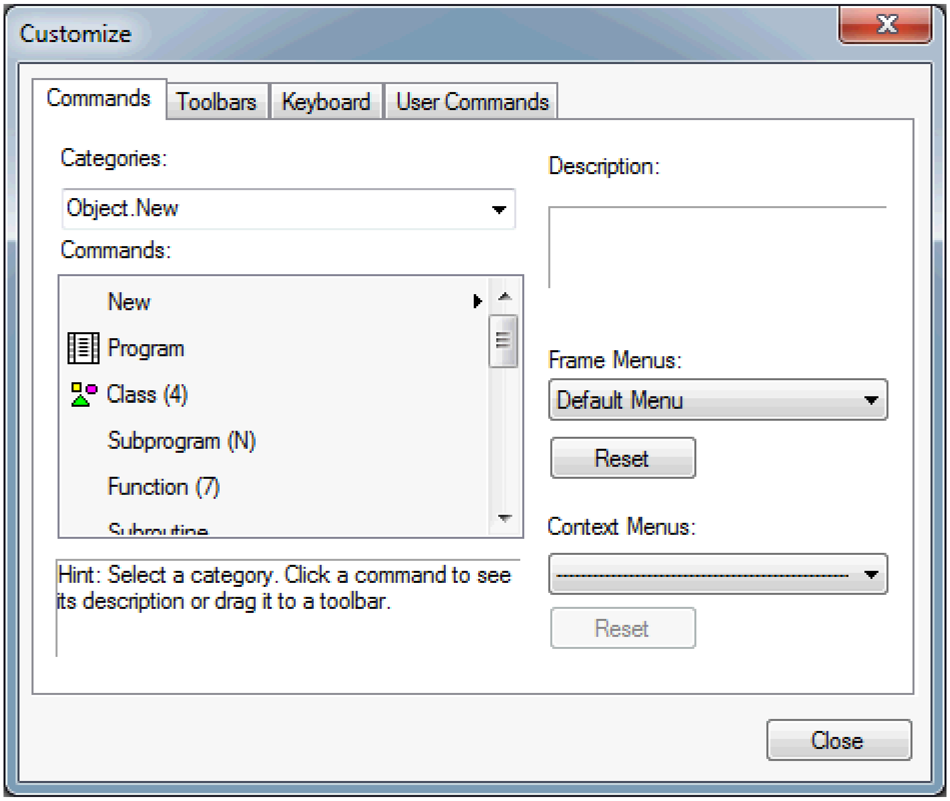
- Commands
- Toolbars
- Keyboard
- User Commands

All of your changes are automatically saved. You can simply close the dialog box after you have made your changes.

3 Choose the **Close** button to close the dialog box.

## Commands

When you invoke the **Commands** page of the **Customize** dialog box, you can add, move and delete commands (in the menu bar and in a context menu) and toolbar buttons.



Categories: Object.New

Frame Menus: Default Menu

Hint: Select a category. Click a command to see its description or drag it to a toolbar.

The following controls are available:

| Control | Description |
|---|---|
| **Categories** / **Commands** / **Description** | The **Categories** drop-down list box lists all menus which are available in Natural Studio. When you select such a menu from the drop-down list box, all commands in this menu are shown in the **Commands** list box; when a toolbar button is available for a command, the corresponding image is also shown. When you select a command, a description for this command is shown in the **Description** text box (if available). This is the same text that is also shown in the status bar, when this command is selected. |
| **Frame Menus** / **Reset** | When you invoke the **Customize** dialog box, the name of the currently active frame menu is automatically selected in the **Frame Menus** drop-down list box. For example, when the program editor was active before the **Customize** dialog box was invoked, **Program Editor** is selected. When you select another frame menu, for example, **Map Editor**, this change is immediately reflected in the Natural Studio window which is shown in the background: the menu bar of the Natural Studio window changes to show all menus which are normally shown when the map editor is active. This is helpful, if you want to customize the menu for a specific editor without having to invoke this editor first. For example, you can then drag a command from the **Customize** dialog box to a menu. See the descriptions below for further information.<br>When you choose the **Reset** button, your own definitions for the currently selected frame menu are reset to the default settings of Natural Studio. You are asked whether you really want to reset the settings. |

| Control | Description |
|---|---|
| **Context Menus** / **Reset** | The **Context Menus** drop-down list box lists all context menus which are available in Natural Studio. When you select, for example, **Node.Lib.Object.ProgramDialog** from this drop-down list box, a context menu appears containing the commands which are normally available when you invoke the context menu for a program or dialog from the library workspace. You can then drag, for example, a command from the **Customize** dialog box to the context menu. See the descriptions below for further information.<br>When you choose the **Reset** button, your own definitions for the currently selected context menu are reset to the default settings of Natural Studio. You are asked whether you really want to reset the settings. |

**Notes:**

1. The descriptions below require that the **Customize** dialog box has been opened.
2. When a toolbar that you want to customize is not shown in the Natural Studio window, go to the **Toolbars** page of the **Customize** dialog box and switch the display for this toolbar on. When you activate one or more toolbars on the **Toolbars** page, for example, the toolbars for the dialog editor, this change is immediately reflected in the Natural Studio window which is shown in the background: it then shows these toolbars. This is helpful, for example, if you want to add or delete toolbar buttons for a specific editor without having to invoke this editor first.

### To add a command to the menu bar

1 Select the required entry in the **Frame Menus** drop-down list box.

The corresponding menus are shown in the menu bar of the Natural Studio window.

2 Select the required menu in the **Categories** drop-down list box.

The commands for this menu are shown in the **Customize** dialog box.

3 Select a command and drag it to the menu bar.

While dragging the command along the menu bar, the menu below the mouse pointer is opened.

4 Drop the command at the desired position.

### To add a command to a context menu

1 Select the required entry in the **Context Menus** drop-down list box.

The corresponding context menu appears in a dialog box.

2 Select the required menu in the **Categories** drop-down list box.

The commands for this menu are shown in the **Customize** dialog box.

3 Select a command and drag it to the dialog box containing the context menu.

4 Drop the command at the desired position.

### To add a command to a toolbar

1 Select the required menu in the **Categories** drop-down list box.

The commands for this menu are shown.

2 Select a command and drag it to a toolbar.

3 Drop the command at the desired position.

When an image has already been defined for the command, a button with this image appears in the toolbar.

When an image has not yet been defined, the **Button Appearance** dialog box appears. See **below**.

**Note:** For the **Find** command (which is used in the **Edit** menu), an edit control (drop-down list box) is shown in the **Commands** list box. When you drag this command to a vertical toolbar, the toolbar button is shown (the control is not shown). When you move this toolbar to a horizontal position, the control is then shown instead of the toolbar button.

### To move a command or toolbar button

1 In the Natural Studio window, select the command in the menu bar or the toolbar button that you want to move.

2 Drag the command or toolbar button to another position.

It is possible to drag a command from a menu to a toolbar, and vice versa.

### To delete a command or toolbar button

1 In the Natural Studio window, select the command in the menu bar or the toolbar button that you want to delete.

Or:

If you want to delete a command from a context menu, select the required context menu in the **Context Menus** drop-down list box of the **Customize** dialog box and then select the command in the resulting dialog box.

2 Invoke the context menu and choose the **Delete** command.

Or:

Simply drag the command or toolbar button out of the menu or toolbar. Make sure not to drag it into another menu or toolbar; this would be a move.

### To change the appearance of a toolbar button

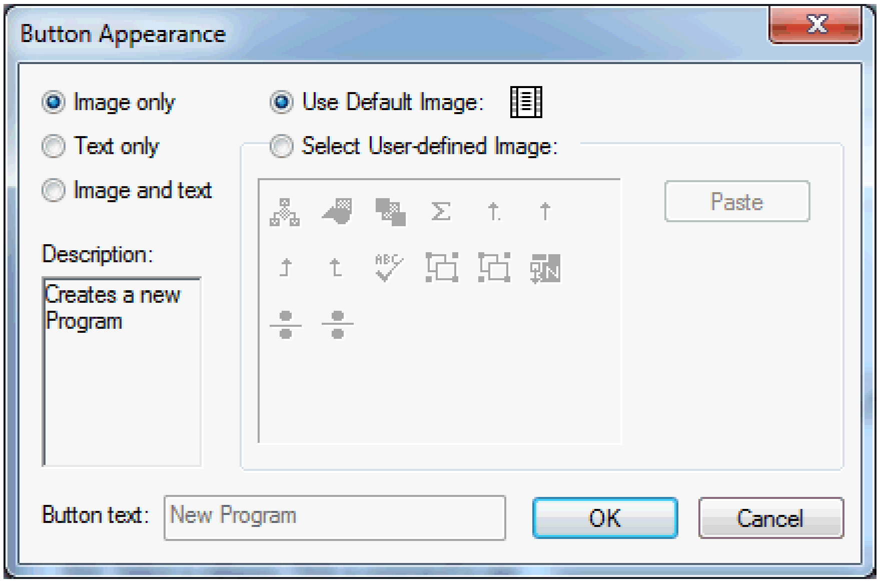
1 In the Natural Studio window, select the command in the menu bar or the toolbar button for which you want to change the appearance.

Or:

If you want to change the appearance of a command in a context menu, select the required context menu in the **Context Menus** drop-down list box of the **Customize** dialog box and then select the command in the resulting dialog box.

2 Invoke the context menu and choose the **Button Appearance** command.

The **Button Appearance** dialog box appears. Example:



● Image only
○ Text only
○ Image and text
● Use Default Image:
○ Select User-defined Image:
Description: Creates a new Program
Button text: New Program

The option buttons for displaying image and/or text are only available when you have selected a toolbar button. They do not apply to the menu commands.

**Note:** The options for displaying image and/or text are also available as commands when you invoke the context menu for a toolbar button. A check mark next to a command indicates that this option has been set.

3 If applicable, specify whether an image and/or text is to be shown in the toolbar by selecting the corresponding option button.

4 If an image other than the default image is to be shown, select the option button **Select user-defined image** (only available when one of the option buttons for displaying the image is selected) and then select one of the images provided in this dialog box.

If you have created your own image, go to the Windows application in which you are currently editing your image and copy it to the Windows clipboard. Then paste it into this dialog box using the **Paste** button. It is recommended that your image has a size of 16x16 pixels.

> **Note:** The images are stored in the file *nideres.bmp* which is located in Natural's *bin* folder. If you have prepared a file containing different images, you can overwrite the file *nideres.bmp* with your own image file. Your own images will then be available for selection in the **Button Appearance** dialog box.

5 When one of the text options is selected, you can specify a different text in the **Button text** text box. This text is then shown in the toolbar. It has no influence on the tooltip.

This text box is only available when you have selected a toolbar button. It does not apply to the menu commands.

6 Choose the **OK** button.

### To start a new group in a menu or toolbar

1 In the Natural Studio window, select the command in the menu bar or the toolbar button before which a separator bar is to be added.

Or:

If you want to add a separator bar to a context menu, select the required context menu in the **Context Menus** drop-down list box of the **Customize** dialog box and then select the command in the resulting dialog box.

2 Invoke the context menu and choose the **Start Group** command.

This is a toggle command which switches the display of the separator bar on and off. A check mark next to the **Start Group** command indicates that a separator bar appears before the currently selected command or toolbar button.

### To reset the assignments for a command or toolbar button to the default

1 In the Natural Studio window, select the command in the menu bar or the toolbar button that is to be reset.

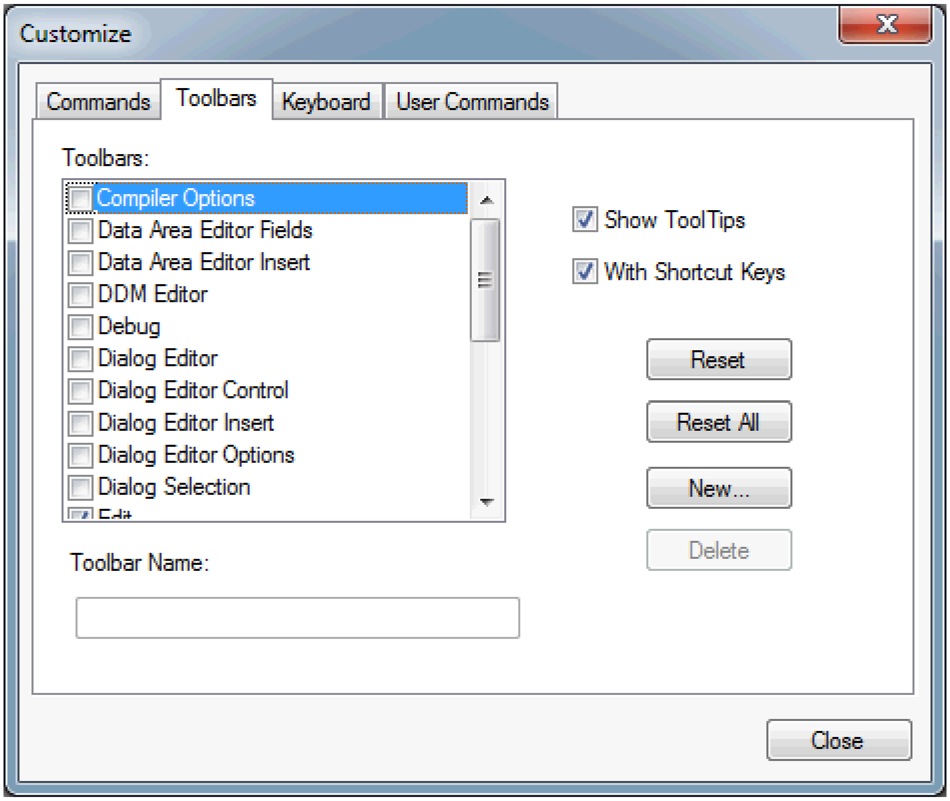
Or:

If you want to reset a command in a context menu, select the required context menu in the **Context Menus** drop-down list box of the **Customize** dialog box and then select the command in the resulting dialog box.

2 Invoke the context menu and choose the **Reset to Default** command.

## Toolbars

When you invoke the **Toolbars** page of the **Customize** dialog box, you can switch the display of specific toolbars on and off. You can also create your own toolbars.



Toolbars:

□ Compiler Options
□ Data Area Editor Fields
□ Data Area Editor Insert
□ DDM Editor
□ Debug
□ Dialog Editor
□ Dialog Editor Control
□ Dialog Editor Insert
□ Dialog Editor Options
□ Dialog Selection
☑ Edit

☑ Show ToolTips

☑ With Shortcut Keys

Toolbar Name:

The following controls are available:

| Control | Description |
|---|---|
| **Toolbars** | This list box contains all toolbars that are currently defined. When a check mark is shown next to a toolbar name, this toolbar is shown in the Natural Studio window. |
| **Toolbar Name** | You can specify a different name for the currently selected user-defined toolbar. Toolbars that are provided with Natural Studio cannot be renamed. |
| **Show Tooltips** | When this check box is selected, tooltips are shown for the toolbar buttons. |

| Control | Description |
|---|---|
| **With Shortcut Keys** | Can only be selected when **Show Tooltips** is also selected. When this check box is selected, the shortcut keys are also shown in the tooltips.<br>**Note:** This option also applies to the menu commands. When deselected, the shortcut keys are not shown next to a menu command. |
| **Reset** | Resets the currently selected toolbar to the default. User-defined toolbars cannot be reset. |
| **Reset All** | Resets all toolbars (except user-defined toolbars) to the default. |
| **New** | Creates a user-defined toolbar. See below. |
| **Delete** | Deletes the currently selected user-defined toolbar. Toolbars that are provided with Natural Studio cannot be deleted. |

### To switch the display of a toolbar on and off

- To display a toolbar in the Natural Studio window, activate the check box next to the toolbar name.

  Or:

  If a toolbar is no longer to be shown in the Natural Studio window, deactivate the check box next to the toolbar name.

  This change is immediately reflected in the Natural Studio window which is shown in the background.

### To add a user-defined toolbar

1. Choose the **New** button.
2. In the resulting dialog box, specify a name for the toolbar and choose the **OK** button.

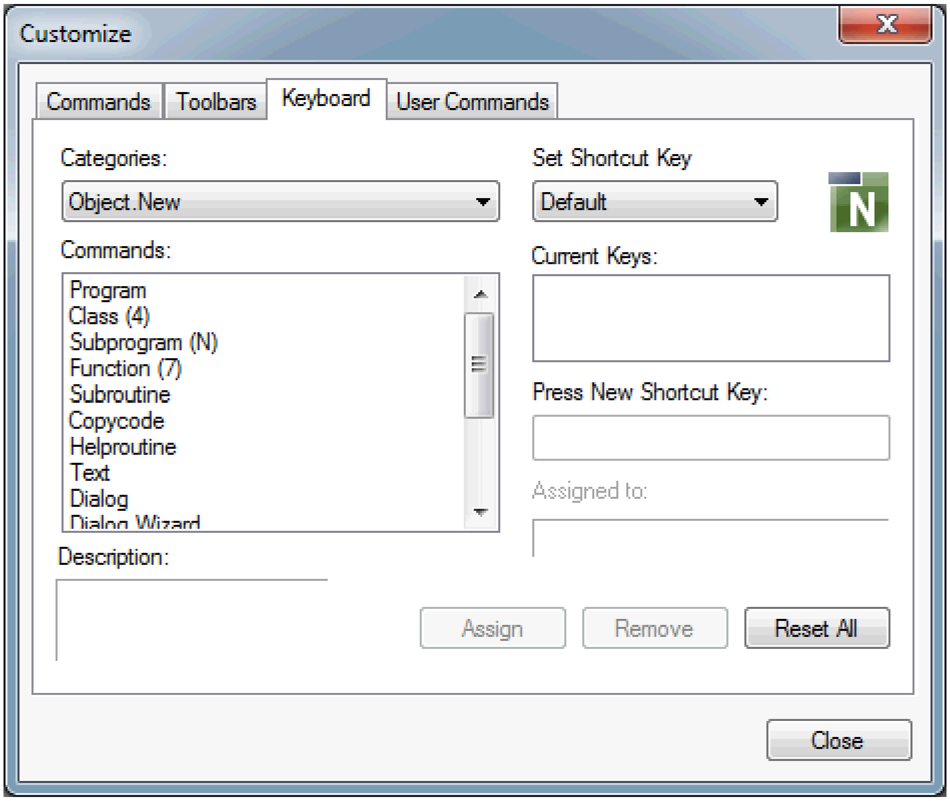
   An empty toolbar is shown as an undocked window. The name you have just provided is shown in its title bar. An entry for the new toolbar is shown in the **Toolbars** list box.
3. Go to **Commands** page of the **Customize** dialog box, select a category and drag all required commands to the new toolbar.

   When an image has not yet been defined for a command, the **Button Appearance** dialog box appears. Proceed as described previously for the **Button Appearance** dialog box.
4. Optional. You can dock the toolbar. See *Dockable Windows* for further information.

**Note:** For information on how to manage the buttons in a toolbar (for example, move and delete), see the description of the **Commands** page.

## Keyboard

When you invoke the **Keyboard** page of the **Customize** dialog box, you can change the assignments for the shortcut keys. See also *Shortcut Keys* which lists the default assignments of the shortcut keys which can be changed by the user.



Categories: Object.New

Set Shortcut Key: Default

### To add a new shortcut key

1 Select the required menu in the **Categories** drop-down list box.

The commands for this menu are shown.

2 From the **Set Shortcut Key** drop-down list box, select the scope in which the shortcut key is to be used.

For example, when you select **Program Editor**, the shortcut key can only be used when the program editor is active. When you select **Default**, the shortcut is always available.

3 In the **Commands** list box, select the command to which you want to assign a shortcut key.

If one or more shortcut keys have already been defined, they are shown in the **Current Keys** list box.

When "(Default)" is shown behind a shortcut key, this key has been defined when **Default** was active in the **Set Shortcut Key** drop-down list box.

The description which is shown at the bottom of the dialog box is shown in the status bar when this command is selected.

4 Place the mouse pointer in the **Press New Shortcut Key** text box.

5 Press the key combination that you want to use as the new shortcut key.

The **Assigned to** text box informs whether this shortcut key can be used:

- When the shortcut key has not yet been defined, it can be used. In this case, a message appears informing you that the key is available.
- When the shortcut key has already been defined, you are informed for which command it has been defined. You have to define another shortcut key.
- When you are informed that this is a reserved keyword, you have to define another shortcut key.

6 When your shortcut key can be used, choose the **Assign** button.

The new shortcut key is added to the **Current Keys** list box. It can then be used in addition to any previously defined shortcut keys.

### To remove a shortcut key

1 Select the required menu in the **Categories** drop-down list box.

The commands for this menu are shown.

2 From the **Set Shortcut Key** drop-down list box, select the scope in which the shortcut key is currently used.

3 In the **Commands** list box, select the command for which you want to remove the shortcut key.

If one or more shortcut keys have been defined for this command, they are shown in the **Current Keys** list box.

When "(Default)" is shown behind a shortcut key, this key can only be removed when **Default** is selected in the **Set Shortcut Key** drop-down list box.

4 In the **Current Keys** list box, select the shortcut key that you want to remove.

**Caution:** Keep in mind that you cannot use Natural Studio's default assignments, which are mentioned in this documentation, when you remove them here.

5 Choose the **Remove** button.

### To reset all shortcut key assignments

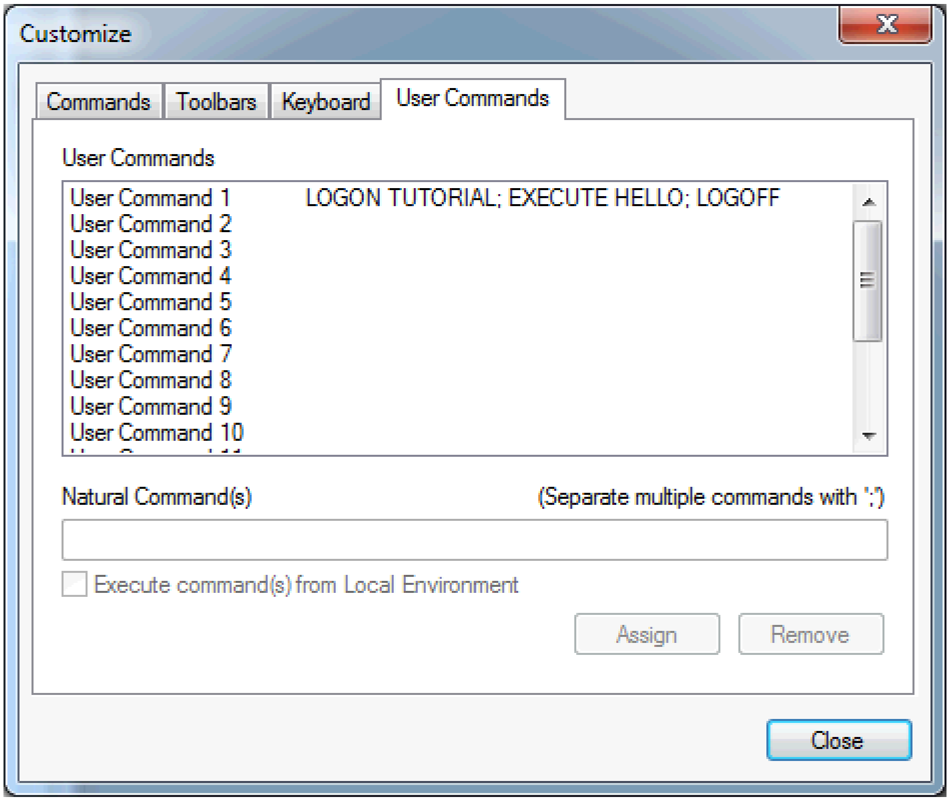
- Choose the **Reset All** button.

  All changed and deleted assignments are restored to the defaults of Natural Studio. All of your own assignments are deleted.

## User Commands

When you invoke the **User Commands** page of the **Customize** dialog box, you can define your own commands. Each command may contain a series of Natural commands. You can use all commands that can also be entered in the command line.

You can define up to 16 user commands. By default, empty entries for these user commands are provided in the **User Commands** list box. When a user command has already been defined, the Natural commands you have specified are shown in the list box. For example:



```
User Command 1     LOGON TUTORIAL; EXECUTE HELLO; LOGOFF
User Command 2
User Command 3
User Command 4
User Command 5
User Command 6
User Command 7
User Command 8
User Command 9
User Command 10
```

### To define a user command

1 In the **User Commands** list box, select the entry for which you want to define Natural commands.

When Natural commands have already been defined for the selected entry, they are shown in the **Natural Command(s)** text box. When Natural commands have not yet been defined, this text box is empty.

2 Specify the required Natural command(s). When you specify more than one command, separate the commands with semicolons. For example:

```
LOGON TUTORIAL; EXECUTE HELLO; LOGOFF
```

The syntax is not checked. Make sure to specify valid commands.

3 Optional. If you are working in a SPoD environment, activate the **Execute command(s) from Local Environment** check box to make sure that your user command is always executed in the local environment.

When this check box is not activated and you invoke the user command while a library in the remote environment is selected, Natural tries to execute your command in the remote environment.

4 Choose the **Assign** button.

Your (new) definition is saved. It appears in the list box.

So that the user command can be invoked, you have to add it to a menu and/or toolbar as described below.

### To add a user command to a menu or toolbar

1 If you want to add the user command to a toolbar (this can also be a user-defined toolbar), make sure that this toolbar has been activated on **Toolbars** page of the **Customize** dialog box.

2 Go to the **Commands** page of the **Customize** dialog box.

3 Select the required frame menu or context menu.

4 From the **Categories** drop-down list box, select **User Commands**.

The user commands are shown.

5 Drag the required user command to the menu bar, context menu or toolbar as described previously in the *Commands* section.

The user command is shown with the default name and/or image.

**Note:** When a Natural command has not been assigned to a user command, the corresponding menu command or toolbar button is dimmed.

### To define another name and image for a user command in a menu or toolbar

1 Select the user command in the menu bar, context menu or toolbar.

2 Invoke the context menu and choose the **Button Appearance** command.

3 Proceed as described previously for the **Button Appearance** dialog box.

### To define a shortcut key for a user command

1 Go to the **Keyboard** page of the **Customize** dialog box.

2 From the **Categories** drop-down list box, select **User Commands**.

3 Select the user command for which you want to assign a shortcut key.

4 Assign a shortcut key as described previously in the ***Keyboard*** section.

### To remove the Natural commands from a user command

1 Select the user command on the **User Commands** page.

2 Choose the **Remove** button.

The defined Natural commands are deleted. The entry for the user command, however, is still shown in the list box.

When you remove the Natural commands, the user command is still available in all menus or toolbars to which it has been added. As long as a Natural command is not assigned, the corresponding menu command or toolbar button is dimmed.

# 12 Setting the Options

This chapter provides information on the options that can be set in Natural Studio.

## Invoking the Options Dialog Box

The settings in the **Options** dialog box control the behavior of certain features of Natural Studio.

### To invoke the Options dialog box

1. From the **Tools** menu, choose **Options**.

   The **Options** dialog box appears. Different pages are provided for setting different types of options.

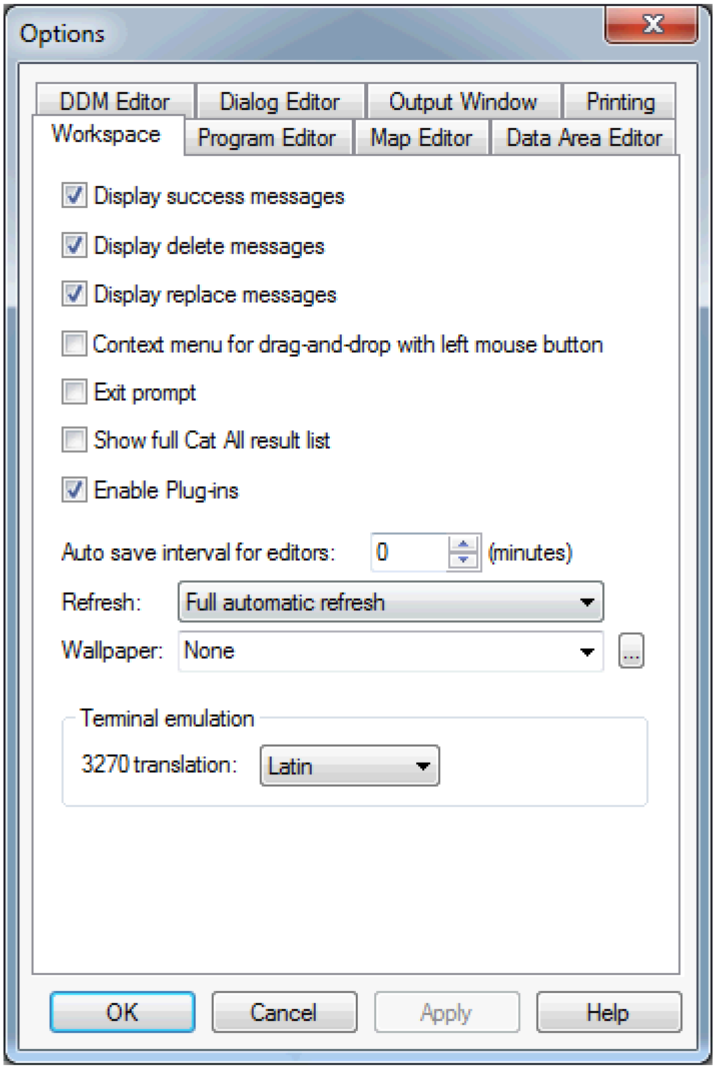
2. Select one of the pages in this dialog box and set the required options as described below under the following headings.

   - Workspace Options
   - Program Editor Options
   - Map Editor Options
   - Data Area Editor Options
   - DDM Editor Options
   - Dialog Editor Options
   - Output Window Options
   - Printing Options

   **Note:** Additional pages may appear in the **Options** dialog box. For example, when XRef Evaluation is active, a page for setting the options for this plug-in is also available. In such a case, you have to refer to the corresponding product documentation for further information.

3. Choose the **OK** button to save your changes.

## Workspace Options



☑ Display success messages

☑ Display delete messages

☑ Display replace messages

□ Context menu for drag-and-drop with left mouse button

□ Exit prompt

□ Show full Cat All result list

☑ Enable Plug-ins

Auto save interval for editors: 0 (minutes)

Refresh: Full automatic refresh

Wallpaper: None

Terminal emulation

3270 translation: Latin

You can set the following options:

**Display success messages**
When this check box is selected, a message box appears each time a command has been performed successfully. For example, you are informed when the stow operation was successful.

**Note:** Error messages are not affected by the setting of this option. They appear each time an error occurs.

**Display delete messages**
When this check box is selected, a message box appears each time you are deleting an object. You are then asked whether you really want to delete the selected object(s).

**Display replace messages**

When this check box is selected, a message box appears each time an object is about to be copied over an already existing object with the same name. You are then asked whether you really want to replace the object.

**Context menu for drag-and-drop with left mouse button**

When this check box is selected, a context menu appears when you drag and drop objects with the left mouse button. The context menu appears when you release the mouse button at the end of the drag-and-drop operation. You can then choose whether to copy or move the objects.

When this check box is not selected (default), a context menu does not appear.

See also *Copying Objects* and *Moving Objects*.

**Exit prompt**

When this check box is selected, a message box appears each time you are about to leave Natural Studio, informing you that this will end your session. The **Cancel** button in this message box can still be used to cancel the exit request.

**Show full Cat All result list**

When this check box is selected, the **Cat All** tab of the results window displays all objects processed with the **Cat All** command, regardless if an error has occurred or not. See *Cataloging the Objects in a Library*.

**Enable plug-ins**

When this check box is selected, the Plug-in Manager and all other plug-ins that are marked for automatic activation are activated when Natural Studio is started.

When this check box is not selected, neither the Plug-in Manager nor any other plug-in is activated when Natural Studio is started.

See also *Plug-in Manager*.

**Note:** This option cannot be selected if the profile parameter `USEREP` is set to "ON".

**Autosave interval for editors**

Use this spin box to specify a time interval in minutes after which the content of all open Natural editors is saved.

0 (zero) means that autosave does not take place.

**Refresh**

Use this drop-down list box to specify how the display is to be refreshed. The following options are available:

- **Full automatic refresh**

  When this option is selected, the display is refreshed automatically for every change. This is the default setting. It is recommended to use this setting as long as it does not cause any performance problems.

- **Optimized automatic refresh**
  When this option is selected, the display is refreshed automatically when a Natural object is changed. This refresh mode is performance-optimized for the case that only few objects of a huge library are changed. If a larger number of objects is changed, the behavior of **Full automatic refresh** is adopted and the increase of performance gained by **Optimized automatic refresh** is lost.

  It is recommended to use **Optimized automatic refresh** only if the performance of **Full automatic refresh** is not acceptable, especially when working in a remote development environment or when working with huge libraries.

  **Optimized automatic refresh** cannot be used in the application workspace.

  With **Optimized automatic refresh** the following differences apply, as compared to **Full automatic refresh**:

  - New library nodes or group nodes are not created. For example, if a library does not contain subprograms up to now, the group node **Subprograms** is not created when the user creates a new Natural subprogram.
  - Library nodes or group nodes are not deleted. For example, if all programs of a library are deleted, the **Programs** node is not deleted.
  - In a list view window, new objects are always inserted at the end of the list.
  - Changes from other users are not reflected in a library.
  - Classes and error messages are not refreshed at all.
- **No automatic refresh**
  When this option is selected, the display is not refreshed at all. You have to refresh the display manually. See ***Refreshing the Display***.

**Wallpaper**

You can select an image to replace the default wallpaper in Natural Studio.

Choose the button with the three dots which is located next to this drop-down list box to select an image file from your file system. The supported image formats are indicated in the resulting dialog box.

When you choose **None** from this drop-down list box, a wallpaper is not used.

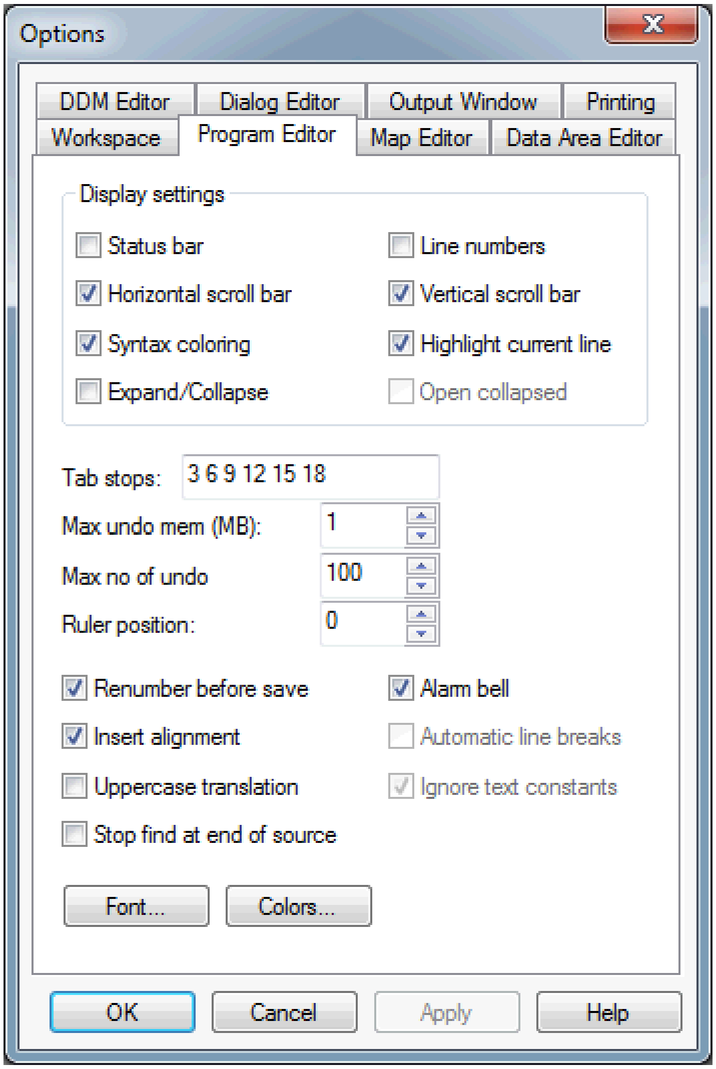
**3270 translation**

From this drop-down list box, you can select another character set to be used for terminal emulation.

Terminal emulation is used for remote development only. See *Terminal Emulation* and its subsection *Translation Tables* in the *Remote Development Using SPoD* documentation.

## Program Editor Options

You can set preferences for various program editor options. These settings are taken as default values each time you start the program editor. See *Program Editor* in the *Editors* documentation for further information.



Display settings

□ Status bar
□ Line numbers
☑ Horizontal scroll bar
☑ Vertical scroll bar
☑ Syntax coloring
☑ Highlight current line
□ Expand/Collapse
□ Open collapsed

Tab stops: 3 6 9 12 15 18

Max undo mem (MB): 1

Max no of undo 100

Ruler position: 0

☑ Renumber before save
☑ Alarm bell
☑ Insert alignment
□ Automatic line breaks
□ Uppercase translation
☑ Ignore text constants
□ Stop find at end of source

You can set the following options:

**Status bar**

When this check box is selected, a status bar is displayed at the top of the program editor window. This status bar provides the same information as the **status bar** which can be displayed at the bottom of the Natural Studio window.

**Line numbers**

When this check box is selected, line numbers are shown in the program editor.

**Horizontal scroll bar**

When this check box is selected, a horizontal scroll bar is shown in the program editor window.

**Vertical scroll bar**

When this check box is selected, a vertical scroll bar is shown in the program editor window.

**Syntax coloring**

When this check box is selected, syntax coloring is used in the program editor. See also *Defining Different Colors for the Syntax*.

**Highlight current line**

When this check box is selected, the program editor line in which the cursor is currently located is highlighted. See also *Defining Different Colors for the Syntax*.

**Expand/Collapse**

When this check box is selected, the source code in the program editor is displayed in expandable/collapsible format. See *Showing and Hiding Source Code* in the *Program Editor* section of the *Editors* documentation for detailed information.

**Open collapsed**

Only available when the check box **Expand/Collapse** is selected.

When this check box is selected, the source code is initially displayed in collapsed format.

When this check box is not selected, the source code is initially displayed in expanded (normal) format.

**Tab stops**

This text box is used to specify the column numbers for tabulator stops in the program editor. You can add or modify existing tabulator stops in the text box. All tabulator stops must be separated by blanks.

**Max undo mem (MB)**

Use this spin box to select the amount of buffer space available for storing text operations made in the program editor. You can specify a value between 0 and 64. Zero signifies the maximum space available.

**Max no of undo**

Use this spin box to select the number of text operations which can be recalled/reinstated using the undo/redo functions. You can specify a value between 0 and 9999. Zero signifies the maximum limit dependent only on the specified maximum memory size (see above).

For more information on the undo/redo functions available in the program editor, see *Undoing or Redoing Modifications* in the *Program Editor* section of the *Editors* documentation.

**Ruler position**

Use this spin box to select the column in which the ruler is to appear. You can specify a value between 0 and 245. 0 (zero) means that the ruler is not shown.

The ruler is a vertical bar which appears in the defined column. It is intended as a visual help when editing a program. For example, when a line of source code should not exceed 80 characters, you can define that the ruler is to appear in column 80.

A long line which extends beyond the ruler line is marked with a vertical bar that appears at the beginning of this line.

**Renumber before save**
When this check box is selected, the lines in a program are renumbered and the line number references are updated before every save.

**Alarm bell**
When this check box is selected, a beep sound is played each time an invalid key or key combination is pressed.

A prerequisites for this is that Natural is installed on your local PC. When you access Natural on a terminal server, you cannot hear the beep sound from your local PC.

**Insert alignment**
When this check box is selected, the cursor is aligned with the first non-blank character of the previous line of text when you press ENTER or RETURN. If there are no non-blank characters before the inserted line, the cursor is aligned with column 1.

**Automatic line breaks**
Only available if a ruler position has been specified (that is: if a value other than 0 is shown in the **Ruler position** spin box; see above).

When this check box is selected, the automatic line break feature is active. See *Using Automatic Line Breaks* in the *Program Editor* section of the *Editors* documentation for detailed information.

**Uppercase translation**
When this check box is selected, the source code is translated to uppercase before it is saved, cataloged or stowed. This is convenient if working with a z/OS source in a SPoD environment, where the z/OS compiler expects Natural sources to be written in uppercase.

**Ignore text constants**
Only available if the check box **Uppercase translation** is selected.

When this check box is selected, text constants placed in single quotation marks are not translated to uppercase. This option has no influence on text constants placed in double quotation marks.

**Stop find at end of source**
When this check box is selected, the **Find** command stops searching at the end of the source code if the specified string is not found. The search is performed in the specified direction (either up or down).

When this check box is not selected and the string is not found, the **Find** command wraps around to the other end of the source code and continues the search.

**Font**

When you choose this button, the **Font** dialog box appears. This is a standard Windows dialog in which you can define another font, font style and font size to be used in the program editor. Only monospaced fonts are available.

**Colors**

When you choose this button, the **Color Definition** dialog box appears. See *Defining Different Colors for the Syntax* below.

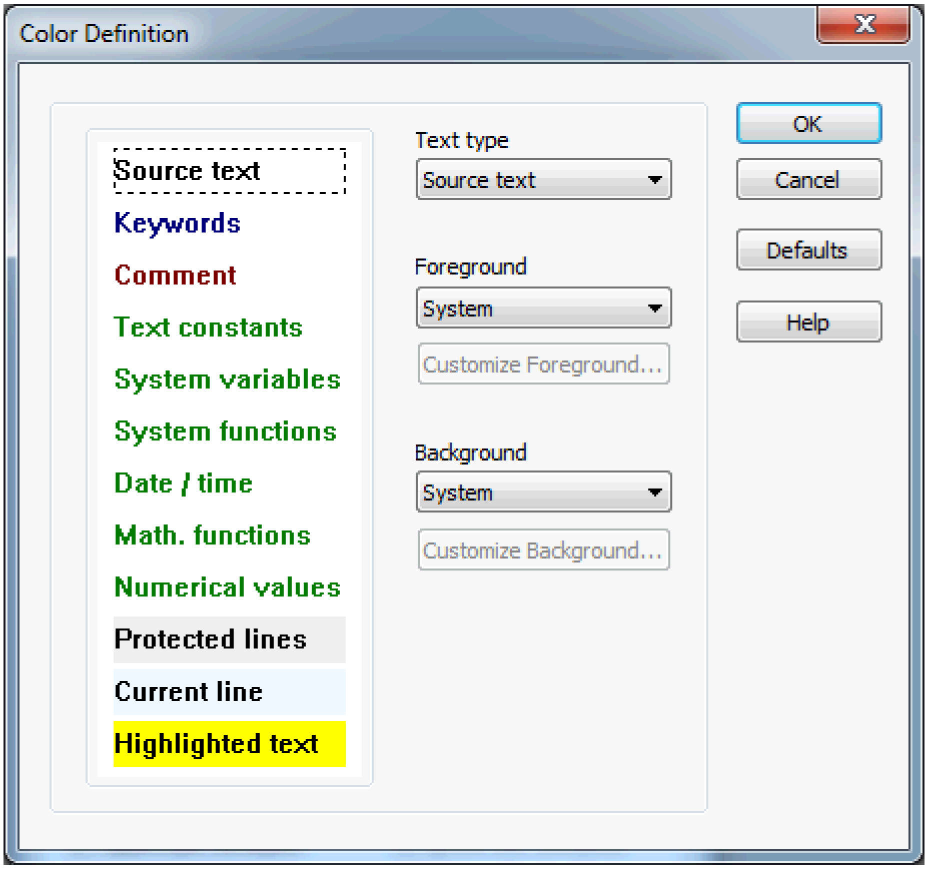
### Defining Different Colors for the Syntax

You can define different colors to be used with the syntax coloring feature of the program editor. Colors are only used in the program editor, when the **Syntax coloring** check box is selected (see above).

#### To define different colors for the syntax

1 On the **Program Editor** page of the **Options** dialog box, choose the **Colors** button.

The **Color Definition** dialog box appears in which you can define different colors for the different text types.



Text type: Source text

Foreground: System

Background: System

The current color for each text type is shown in the sample area at the left of the dialog box. The default foreground color assignments are:

| Default color | Used for the following text types |
|---|---|
| Blue | Natural keywords. |
| Red | Comments. |
| Green | Text constants, system variables, system functions, date/time, mathematical functions, and numerical values. |
| Black | User-defined variables, and all other remaining syntax elements. This color is defined in Windows (as defined for the item "Window"). |

For several text types (protected lines, current line, highlighted text), you can only define a different background color.

**Notes:**

1. The text type “Numerical values” also defines the color for the constants `TRUE` and `FALSE`.
2. For more information on the text type “protected lines”, see *Protecting Source-Code Lines* in the *Program Editor* section of the *Editors* documentation.
3. The text type “highlighted text” applies to the **Find** command. See *Finding Text* in the *Program Editor* section of the *Editors* documentation.

2 Select the text type for which you want to change the color from the **Text Type** drop-down list box.

Or:

Select the corresponding text type in the sample area.

**Note:** To select the text type “edit window” from the sample area, click any position that does not contain text. Only a background color can be defined for the edit window.

The drop-down list boxes **Foreground** and **Background** show the currently defined colors for the selected text type. The color name **System** means that the color as defined in Windows in used.

3 From the **Foreground** drop-down list box, choose the color that is to be used for text.

When a color other than **System** is selected, you can also choose the **Customize Foreground** button to invoke the standard Windows **Color** dialog box in which you can define your own custom colors. The custom color can then be selected by choosing the color name **Custom** from the drop-down list box. Note that this always uses the color which was last selected in the **Color** dialog box.

4 From the **Background** drop-down list box, choose the color that is to be used as the background color for text.

When a color other than **System** is selected, you can also choose the **Customize Background** button to invoke the standard Windows **Color** dialog box in which you can define your own custom colors. The custom color can then be selected by choosing the color name **Custom** from the drop-down list box. Note that this always uses the color which was last selected in the **Color** dialog box.

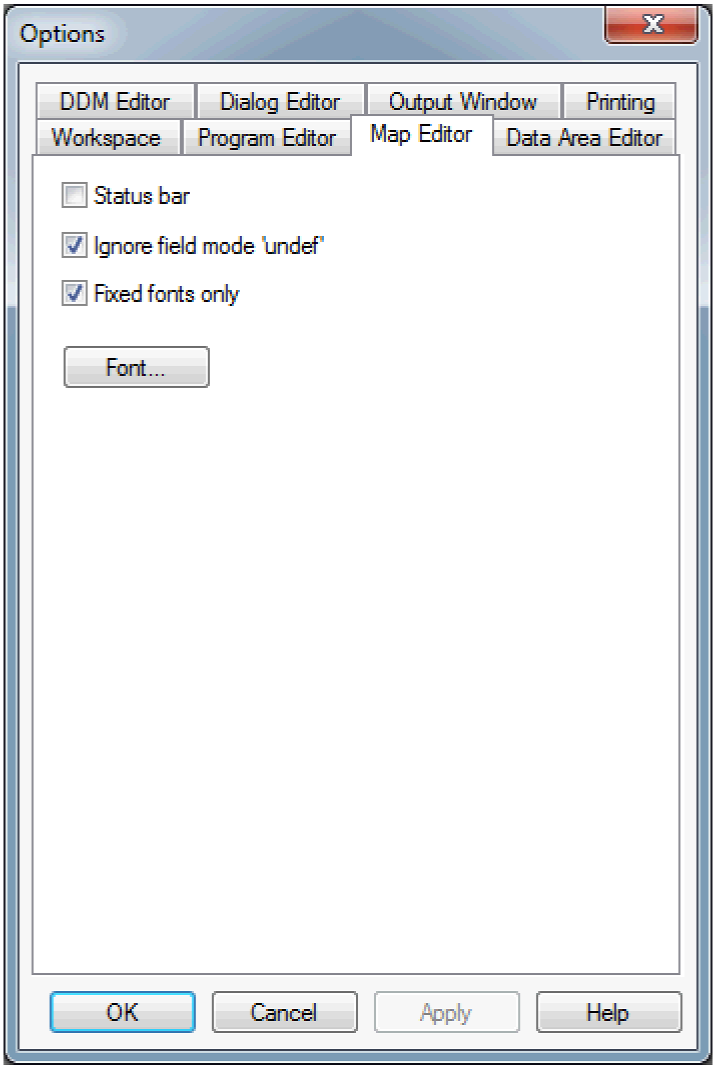
5 Optional. If you want to return to the default syntax colors, choose the **Default** button.

6 Optional. To save your settings without closing the dialog box, choose the **Apply** button.

7 To save your changes and close the dialog box, choose the **OK** button.

## Map Editor Options

You can set preferences for various map editor options. These settings are taken as default values each time you start the map editor. See *Map Editor* in the *Editors* documentation for further information.



□ Status bar

☑ Ignore field mode 'undef'

☑ Fixed fonts only

You can set the following options:

**Status bar**

When this check box is selected, the status bar is displayed at the top of the map editor window. This status bar provides the same information as the **status bar** which can be displayed at the bottom of the Natural Studio window.

**Ignore field mode 'undef'**

When this check box is selected, all fields that have been created on the map but have not yet been named or defined (that is, they have only the values generated by the map editor) are ignored when you **check** or **stow** the map.

**Fixed fonts only**

When this check box is selected, only fixed-width fonts are shown in the **Font** dialog box (see below).

When this check box is not selected, you can select proportional-width fonts in the **Font** dialog box, in addition to the fixed-width fonts.

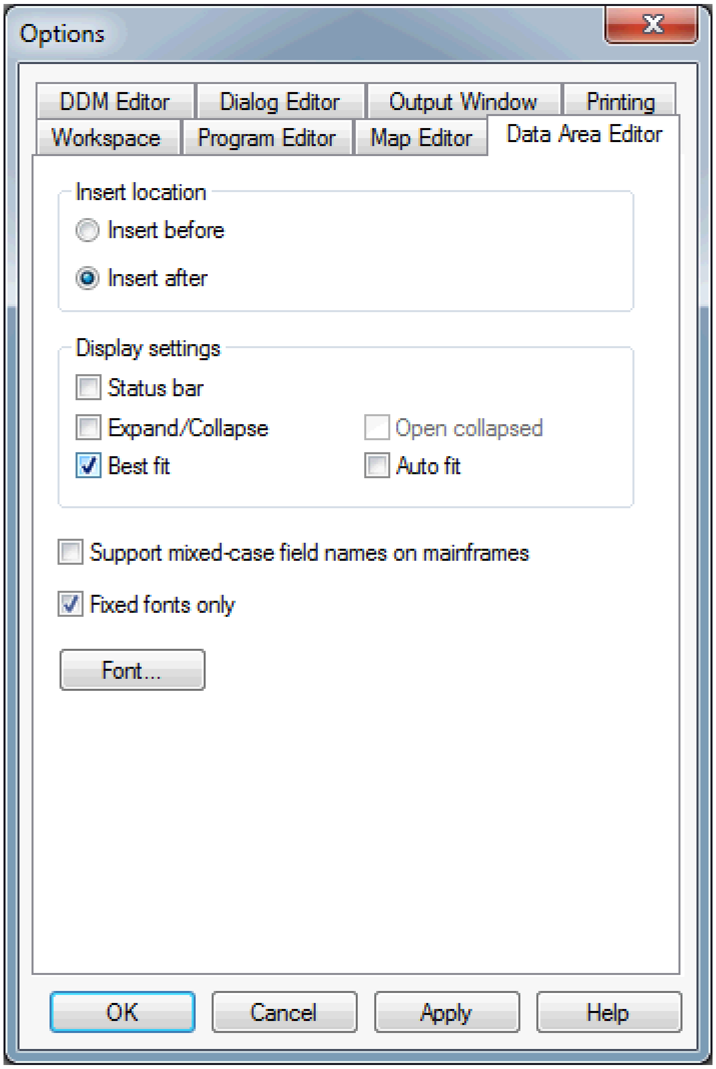
**Font**

When you choose this button, the **Font** dialog box appears. This is a standard Windows dialog in which you can define another font, font style and font size to be used in the map editor.

⚠ **Important:** The font that you define here is only used when you have defined the same font in the **output window options**. Otherwise the map is output with the default font.

## Data Area Editor Options

You can set preferences for various data area editor options. These settings are taken as default values each time you start the data area editor. See *Data Area Editor* in the *Editors* documentation for further information.



Insert location

○ Insert before

● Insert after

Display settings

□ Status bar

□ Expand/Collapse □ Open collapsed

☑ Best fit □ Auto fit

□ Support mixed-case field names on mainframes

☑ Fixed fonts only

You can set the following options:

**Insert location**

You can specify where fields are to be inserted. Select one of the following option buttons:

- **Insert before**
  When this option button is selected, a field is always inserted before the currently selected field.
- **Insert after**
  When this option button is selected, a field is always inserted after the currently selected field.

**Status bar**

When this check box is selected, the status bar is displayed at the top of the data area editor window. This status bar provides the same information as the **status bar** which can be displayed at the bottom of the Natural Studio window.

**Expand/Collapse**

When this check box is selected, the fields in the data area editor are displayed in expandable/collapsible format. See *Showing or Hiding Fields* in the *Data Area Editor* section of the *Editors* documentation for detailed information.

**Open collapsed**

Only available when the check box **Expand/Collapse** is selected.

When this check box is selected, the fields are initially displayed in collapsed format.

When this check box is not selected, the fields are initially displayed in expanded (normal) format.

**Best fit**

When this check box is selected, the width of each visible column is adjusted so that the column header and the content of a column are always completely visible. See *Rearranging Columns* in the *Data Area Editor* section of the *Editors* documentation for further information.

**Auto fit**

Only available when the check box **Best fit** is selected.

When this check box is selected, each edited column is automatically readjusted to the optimum width (as described above) when you leave the column.

When this check box is not selected, the width of the edited column is not readjusted when you leave the column.

**Support mixed-case field names on mainframes**

When this check box is not selected, the field names are translated to uppercase during editing.

When this check box is selected, the field names are not changed. This is convenient if working with a mainframe source in a SPoD environment where the names can be stored in a combination of uppercase and lowercase characters.

**Fixed fonts only**

When this check box is selected, only fixed-width fonts are shown in the **Font** dialog box (see below).

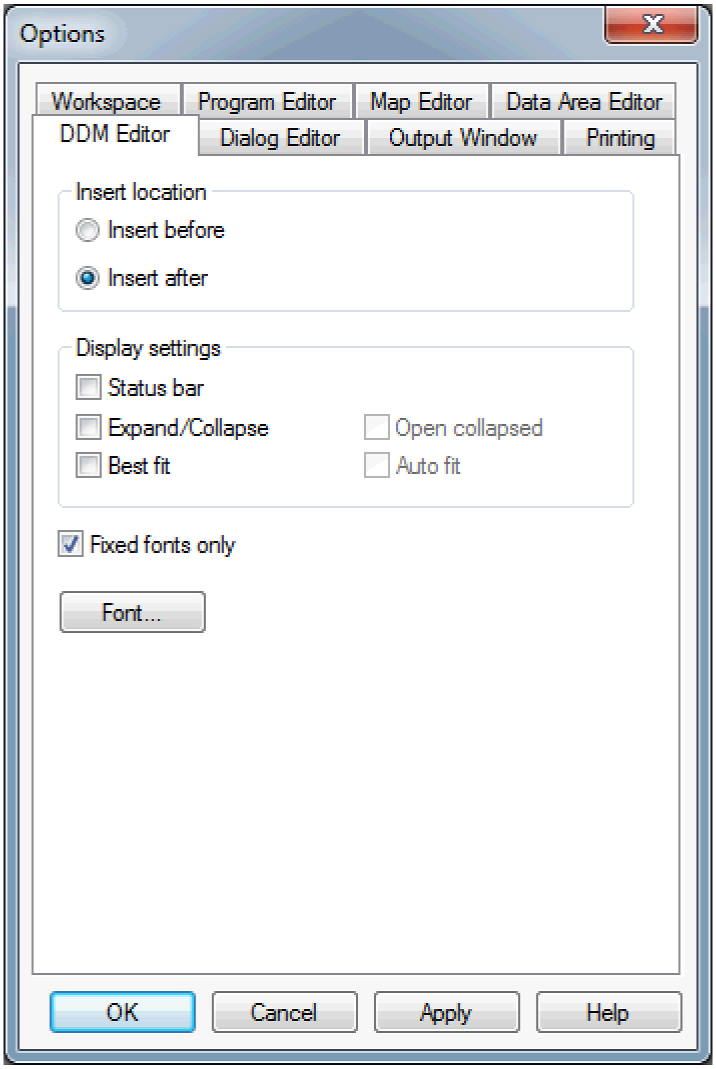
When this check box is not selected, you can select proportional-width fonts in the **Font** dialog box, in addition to the fixed-width fonts.

**Font**

When you choose this button, the **Font** dialog box appears. This is a standard Windows dialog in which you can define another font, font style and font size to be used for the edit area.

## DDM Editor Options

You can set preferences for various DDM editor options. These settings are taken as default values each time you start the DDM editor. See *DDM Editor* in the *Editors* documentation for further information.



Insert location

○ Insert before

● Insert after

Display settings

□ Status bar

□ Expand/Collapse □ Open collapsed

□ Best fit □ Auto fit

☑ Fixed fonts only

You can set the following options:

**Insert location**

You can specify where fields are to be inserted. Select one of the following option buttons:

- **Insert before**
  When this option button is selected, a field is always inserted before the currently selected field.
- **Insert after**
  When this option button is selected, a field is always inserted after the currently selected field.

**Status bar**

When this check box is selected, the status bar is displayed at the top of the DDM editor window. This status bar provides the same information as the **status bar** which can be displayed at the bottom of the Natural Studio window.

**Expand/Collapse**

When this check box is selected, the fields in the DDM editor are displayed in expandable/collapsible format. See *Showing or Hiding Fields* in the *DDM Editor* section of the *Editors* documentation for detailed information.

**Open collapsed**

Only available when the check box **Expand/Collapse** is selected.

When this check box is selected, the fields are initially displayed in collapsed format.

When this check box is not selected, the fields are initially displayed in expanded (normal) format.

**Best fit**

When this check box is selected, the width of each visible column is adjusted so that the column header and the content of a column are always completely visible. See *Rearranging Columns* in the *DDM Editor* section of the *Editors* documentation for further information.

**Auto fit**

Only available when the check box **Best fit** is selected.

When this check box is selected, each edited column is automatically readjusted to the optimum width (as described above) when you leave the column.

When this check box is not selected, the width of the edited column is not readjusted when you leave the column.

**Fixed fonts only**

When this check box is selected, only fixed-width fonts are shown in the **Font** dialog box (see below).

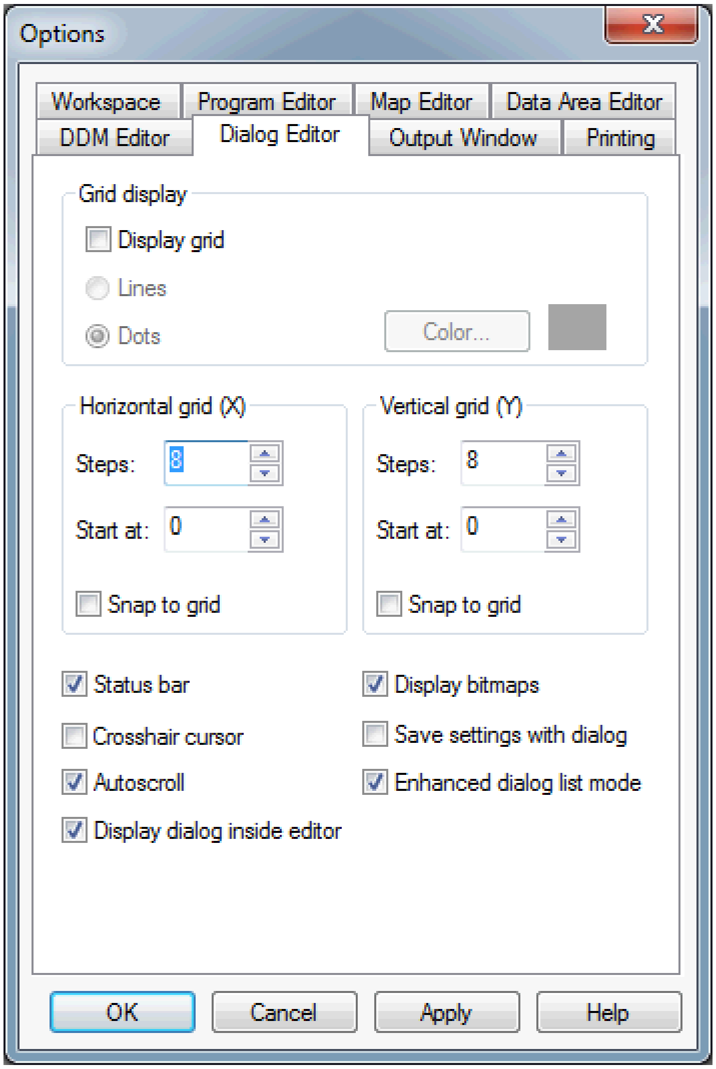
When this check box is not selected, you can select proportional-width fonts in the **Font** dialog box, in addition to the fixed-width fonts.

**Font**

When you choose this button, the **Font** dialog box appears. This is a standard Windows dialog in which you can define another font, font style and font size to be used for the DDM source code in the editing area.

## Dialog Editor Options

You can set preferences for various dialog editor options. These settings are taken as default values each time you start the dialog editor. See *Dialog Editor* in the *Editors* documentation for further information.



Grid display

□ Display grid

○ Lines

● Dots

Horizontal grid (X)

Steps: 8

Start at: 0

□ Snap to grid

Vertical grid (Y)

Steps: 8

Start at: 0

□ Snap to grid

☑ Status bar

☑ Display bitmaps

□ Crosshair cursor

□ Save settings with dialog

☑ Autoscroll

☑ Enhanced dialog list mode

☑ Display dialog inside editor

You can set the following options:

**Grid display**

When the **Display grid** check box is selected, a grid is displayed in the edit area of the dialog editor. The grid is helpful when aligning dialog elements.

The following options are only available when the **Display grid** check box is selected:

- **Lines**
  When this option button is selected, a classical grid is shown.

- **Dots**
  When this option button is selected, a dots pattern is used for the grid.

- **Color**
  When you choose this button, the standard Windows **Color** dialog box appears in which you can select or define another color to be used for the grid. The currently selected color is then shown in the **Options** dialog box, next to the **Color** button.

**Horizontal grid (X) / Vertical grid (Y)**
The following options are available in these two group boxes:

- **Steps**
  Use this spin box to select the number of pixels (steps) between two lines or dots.
- **Start at**
  Use this spin box to select starting value for the grid. For example, when you specify the value "20" for the horizontal grid and "10" for the vertical grid, the grid is moved 20 pixels to the right and 10 pixels to the bottom.
- **Snap to grid**
  When this check box is selected, the dialog elements snap to the grid.

**Status bar**
When this check box is selected, the status bar is displayed at the top of the dialog editor window. This status bar provides the same information as the **status bar** which can be displayed at the bottom of the Natural Studio window.

**Display bitmaps**
When this check box is selected, the dialog editor loads and displays the bitmaps in the dialog elements to show how they appear at runtime.

When this check box is not selected, the dialog element displays the bitmap name only, which improves editing performance.

**Crosshair cursor**
When this check box is selected, the crosshair cursor is used when you create or move dialog elements. The crosshair cursor consists of lines that extend horizontally and vertically to the windows borders.

When this check box is not selected, the system cursor is used.

**Save settings with dialog**
When this check box is selected, the option settings are always saved with a dialog.

Dialogs saved while this option is disabled continue to use the current option settings.

**Autoscroll**
When this check box is selected, the dialog editor window is automatically scrolled so that the dialog is always visible in the dialog editor window. This is also helpful when you move the dialog within the dialog editor window.

When this check box is not selected, and the dialog has been resized so that the defined dialog is currently not visible, you have to use the scroll bars to find the dialog in the dialog editor window.

It is recommended that you activate this option when the dialog is displayed inside the dialog editor window (see the corresponding option below).

**Enhanced dialog list mode**

Starting with Natural for Windows Version 4.1, dialog sources are always saved in enhanced format. Therefore, the line number inconsistency mentioned below does no longer exist. The older (22C) format is only seen if old dialogs in this format are listed or printed outside of the dialog editor. Enabling this option is only meaningful if you **list** or print non-enhanced dialog sources; it does not apply to the new dialog source format.

When this check box is selected, the dialog source is shown in a format that is more readable than the source code of the old Natural versions. If you choose to display the code listing after receiving a compiler error, you will get the code generated for the compiler.

If you have saved your dialog sources in non-enhanced format and the enhanced listing option is enabled, any Natural runtime error message will contain an incorrect line number. To ensure that you get the correct line number, disable this option.

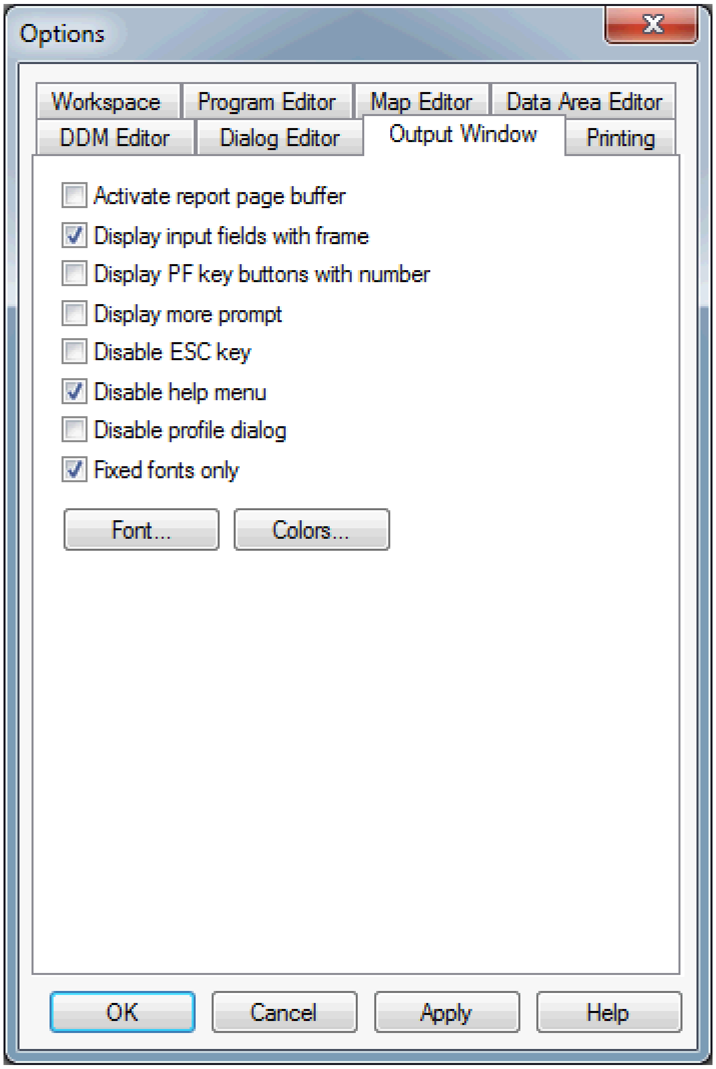
**Display dialog inside editor**

When this check box is selected, the current dialog is displayed inside the dialog editor window.

When this check box is not selected, the current dialog is displayed outside the dialog editor window. This provides more space for editing. Be careful in this case. Your dialog may disappear from view if its `RECTANGLE-X` and `RECTANGLE-Y` values denote a position outside the screen (see the *Dialog Component Reference* for further information on these values). Another consequence is that the dialog will overlap any code listing window that appears.

## Output Window Options

You can set preferences for various output window options. These settings are used each time a Natural program writes output to the screen. This output is shown in an output window. See *Output Window* in the *Operations* documentation for further information.



□ Activate report page buffer
☑ Display input fields with frame
□ Display PF key buttons with number
□ Display more prompt
□ Disable ESC key
☑ Disable help menu
□ Disable profile dialog
☑ Fixed fonts only

You can set the following options:

**Activate report page buffer**

When this check box is selected, buffering is active. Buffering accommodates approximately 250 lines of Natural output. Input empties the report page buffer.

**Display input fields with frame**

When this check box is selected, all input fields are shown with a border (frame).

**Display PF key buttons with number**

When this check box is selected, the PF key buttons contain the numbers of the associated PF key. The name of each PF key is displayed below the corresponding PF key button.

When this check box is not selected, the PF key buttons contain the names of the PF keys (instead of the numbers).

**Display more prompt**
When this check box is selected, the MORE prompt is shown for output generated by the Natural statements DISPLAY, WRITE or PRINT.

**Disable ESC key**
When this check box is selected, the ESC key cannot be used to quit the current Natural program.

**Disable help menu**
When this check box is selected, the Natural output window does not show a **Help** menu. When no other menus have been defined, the Natural output window does not show a menu bar.

**Disable profile dialog**
When this check box is selected, the output window options can only be set in Natural Studio. In this case, the **Profile** command is not available in the control menu of the output window. See also *Changing the Output Window Profile* in the *Operations* documentation.

**Fixed fonts only**
When this check box is selected, only fixed-width fonts are shown in the **Font** dialog box (see below).

When this check box is not selected, you can select proportional-width fonts in the **Font** dialog box, in addition to the fixed-width fonts.

**Font**
When you choose this button, the **Font** dialog box appears. This is a standard Windows dialog in which you can define another font, font style and font size to be used in the output window.

**Colors**
When you choose this button, the **Color Profiling** dialog box appears. See *Defining Different Colors for the Output* below.

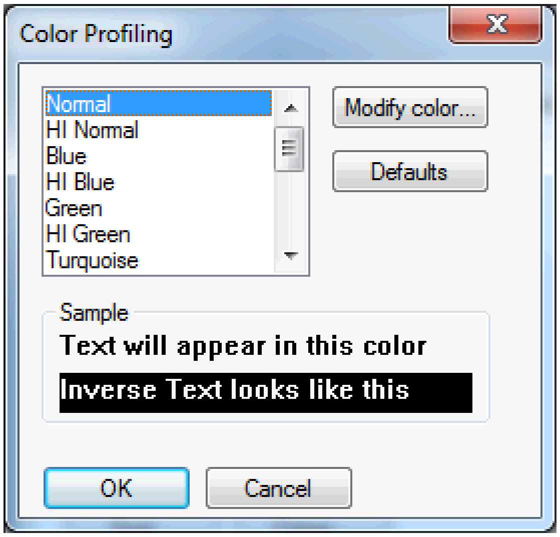
## Defining Different Colors for the Output

You can define different colors to be used for all Natural output in the output window.

### To define different colors for the output

1 On the **Output Window** page of the **Options** dialog box, choose the **Colors** button.

The **Color Profiling** dialog box appears in which you can define different colors for the different color attributes.



Sample

Text will appear in this color

Inverse Text looks like this

2 Select the attribute for which you want to change the color in the list box.

The current color for the selected attribute is shown in the sample area at the bottom of the dialog box.

3 Choose the **Modify Color** button to invoke the standard Windows **Color** dialog box.

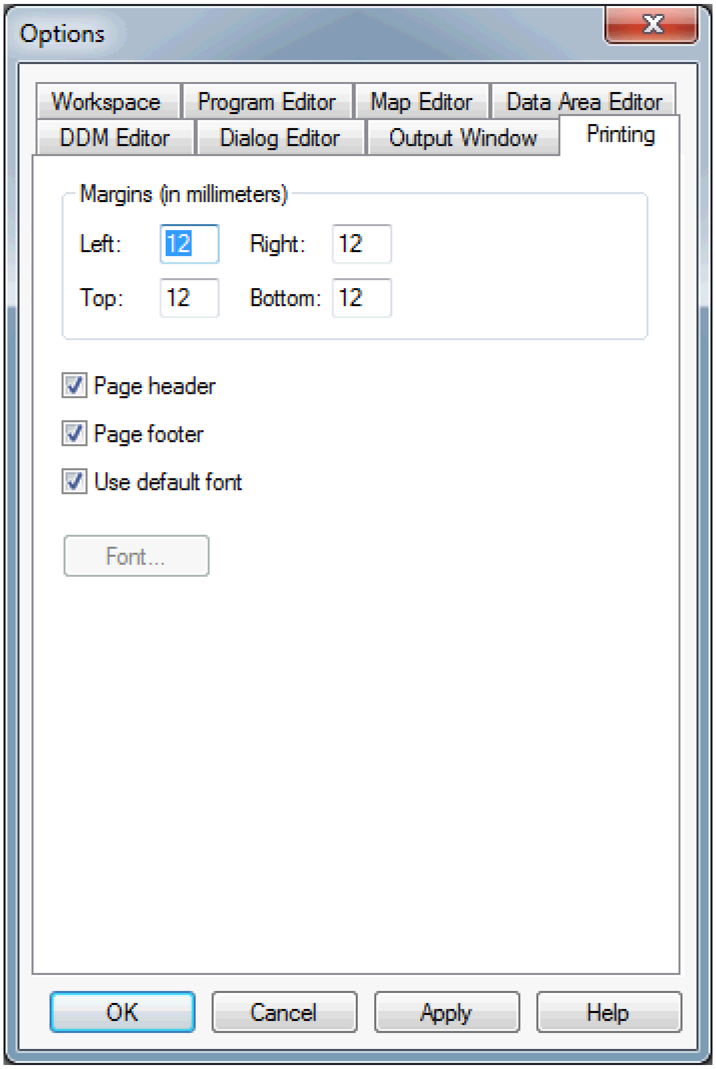
4 Select a color or define a custom color in the **Color** dialog box and choose the **OK** button to return to the **Color Profiling** dialog box.

5 Optional. If you want to return to the default colors, choose the **Default** button.

6 To save your changes and close the dialog box, choose the **OK** button.

## Printing Options

You can set preferences for various printing options. These settings are used each time you print a Natural object or a list view. See *Printing Objects and Views* for further information.



Margins (in millimeters)

Left: 12 Right: 12

Top: 12 Bottom: 12

☑ Page header

☑ Page footer

☑ Use default font

You can set the following options:

**Margins**

The values in these text boxes determine the margins, in millimeters, to be set on each side of the printed page. If any of these values is less than the minimum margin supported by the printer, this minimum value is used instead of the value entered.

**Page header**

When this check box is selected, a header is printed at the top of each page, containing the date and time, and the relevant object identification information (such as object name, type and library).

**Page footer**

When this check box is selected, a footer is printed at the bottom of each page containing the current page number.

**Use default font**

When this check box is selected, the default font is used for printing.

The default font for a Natural object is defined on the corresponding page of the **Options** dialog box. For example, when you print an object (either from an editor window or list view window) which is handled by the program editor, the font which is currently defined for the program editor is used.

**Font**

Only available when the **Use default font** check box is not selected.

You can define a font that is to be used instead of the default font. When you choose this button, the **Font** dialog box appears. This is a standard Windows dialog in which you can define the font, font style and font size to be used for printing. The font you define here is used whenever you print a Natural object or list view. Only monospaced fonts are available.

# 13 Using Session Parameters

When you start Natural Studio, the Natural parameter file (which contains the so-called “profile parameters”) is read to determine the settings of several parameters which customize Natural to your specific environment and requirements. Some of these parameters can be updated after the Natural session is up and running; these are the so-called “session parameters”.

**Note:** The profile parameters are defined with the Configuration Utility. It is possible to start Natural Studio with your own parameter file. See *Invoking Natural with an Alternative Parameter File* in the *Configuration Utility* documentation.

Modifications to the session parameters are only valid for the current session. When you exit Natural, your modifications are discarded. The next time you start Natural Studio, the settings in the Natural parameter file (which are defined with the Configuration Utility) are used again.

**Note:** This default behavior can be changed by setting the profile parameter `PERSIST` in the Configuration Utility.

This chapter describes parameters that can be updated during a running session.

For a full description of all available session parameters, see the *Parameter Reference*.

When you are working in a remote development environment using SPoD, the **Natural Session Parameters** window may contain parameters which do not apply to the Windows version of Natural and which are not listed in this chapter. See the Natural documentation for the corresponding platform for further information on these parameters.

## Invoking the Natural Session Parameters Window

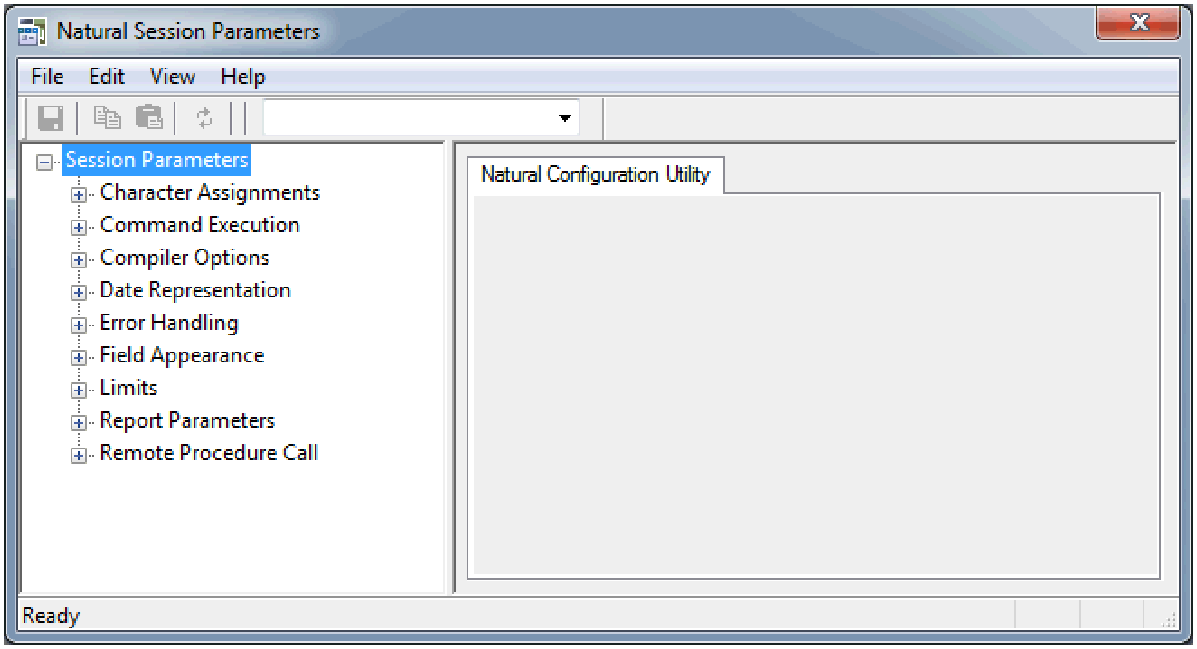
A separate application is available for setting the Natural session parameters. This application is invoked from Natural Studio.

#### To invoke the Natural Session Parameters window

- From the **Tools**, choose **Session Parameters**.

  **Note:** This corresponds to issuing the system command `GLOBALS` without parameters.

  The following window appears.



When you select a node for a parameter group in the tree, the corresponding parameters are shown on the right of the window.

## Locating and Changing a Parameter

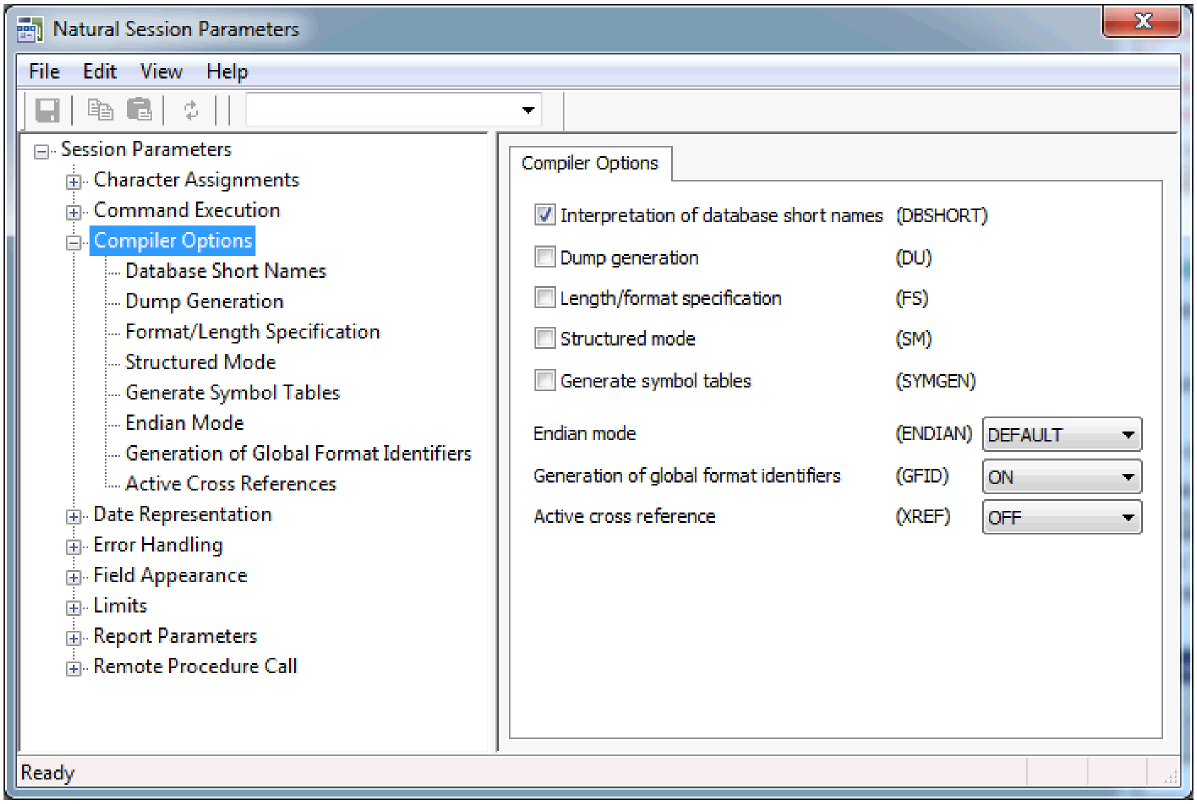
You can locate a parameter in different ways:

- If you know in which parameter group the parameter can be found, you can expand the tree and select the node.
- If you do not know the parameter group, you can search for the parameter.

### To locate and change a parameter

1 Select the parameter group containing the parameter that you want to change.

The names of the Natural session parameters are shown in parenthesis on the right of the window, behind the label of the corresponding control. Example:



| | | |
|---|---|---|
| ☑ Interpretation of database short names | (DBSHORT) | |
| ☐ Dump generation | (DU) | |
| ☐ Length/format specification | (FS) | |
| ☐ Structured mode | (SM) | |
| ☐ Generate symbol tables | (SYMGEN) | |
| Endian mode | (ENDIAN) | DEFAULT |
| Generation of global format identifiers | (GFID) | ON |
| Active cross reference | (XREF) | OFF |

Or:

Expand the node for a parameter group.

Descriptions for the parameters are shown in the tree, below the node. When you select a description in the tree, all parameters for this group are shown on the right of the window. For the description that has been selected in the tree, the corresponding parameter is also selected on the right of the window and can be modified immediately.

2 Set the parameter to the required value (see the remainder of this section for further information).

**Note:** The following applies for check boxes: when a check box is selected, the corresponding parameter is set to "ON"; when a check box is not selected, the corresponding parameter is set to "OFF".

When a parameter has been changed, an asterisk (*) appears next to the top-level node in the tree.

3 Optional. From the **File** menu, choose **Restore Saved** to discard your modifications and thus to restore the settings to the last saved state.

Or:

Choose the following toolbar button:

🔄

4 From the **File** menu, choose **Save** to save your modifications.

Or:

Choose the following toolbar button:

💾

Or:

Press CTRL+S.

The new settings are now used in Natural Studio.

5 Close the window in which you have set the session parameters.

### To find a parameter

1 From the **Edit** menu of the **Natural Session Parameters** window, choose **Find**.

Or:

Press CTRL+F.

When the toolbar is currently not shown, it is automatically activated and the drop-down list box containing the parameters is opened.

**Note:** When the toolbar is already visible, you can simply open the drop-down list box, without having to use the above command.

2 Select the parameter from the drop-down list box in the toolbar.

The parameter group containing the selected parameter is shown on the right of the window. You can now change the parameter setting and save your changes as described above.

**Alternatives**

Session parameters can also be set as follows:

- Enter the system command `GLOBALS` with the required parameter in the command line (session level).
- Reporting mode only: use the statement `SET GLOBALS` with the required parameter in a Natural program (program level).

- Set the session parameter(s) at the statement and/or field level with certain Natural statements (`FORMAT`, `DISPLAY`, `INPUT`, `REINPUT`, `WRITE`, `PRINT`).

## Character Assignments

The following session parameters are used to change default character assignments.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Terminal command character** | `CF` |
| **Decimal character** | `DC` |
| **Input assign character** | `IA` |
| **Input delimiter character** | `ID` |
| **Thousands separator character** | `THSEPCH` |

All character assignments must be mutually exclusive.

## Command Execution

The following session parameter is used to control the execution of commands.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Disable Natural commands** | `NC` |

## Compiler Options

The following session parameters are used to set options for the Natural compiler.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Interpretation of database field short names** | `DBSHORT` |
| **Dump generation** | `DU` |
| **Length/format specification** | `FS` |
| **Structured mode** | `SM` |
| **Endian mode** | `ENDIAN` |

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Generation of global format identifiers** | `GFID` |
| **Active cross reference** | `XREF` |

See also *Compiler Options* in the *Configuration Utility* documentation and the description of the system command `COMPOPT`.

## Date Representation

The following session parameters are used to control the representation of dates.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Date format in output** | `DFOUT` |
| **Date format in STACK** | `DFSTACK` |
| **Date format in report titles** | `DFTITLE` |

## Error Handling

The following session parameters are used to control error handling within Natural.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Automatic REINPUT** | `REINP` |
| **Sound terminal alarm** | `SA` |
| **Record hold processing** | `WH` |
| **Zero division** | `ZD` |
| **Code page conversion error** | `CPCVERR` |

## Field Appearance

The following session parameters influence I/O handling.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Filler chars protected** | `FCDP` |
| **Overwriting protected** | `OPF` |
| **Zero printing** | `ZP` |
| **Message line position** | `ML` |
| **Print mode** | `PM` |

## Limits

The following session parameters are used to prevent a single program from consuming an excessive amount of internal resources.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Error on loop limit** | `LE` |
| **Processing loop limit** | `LT` |

## Report Parameters

The following session parameters are used to control various attributes of Natural reports.

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Page eject** | `EJ` |
| **Line size** | `LS` |
| **Page size** | `PS` |
| **Spacing factor** | `SF` |
| **Default input terminal mode** | `IM` |

## Remote Procedure Call

The following session parameters apply to the Natural Remote Procedure Call (RPC).

| Option | See the *Parameter Reference* for a description of this session parameter |
|---|---|
| **Retry service on alternative server** | TRYALT |
| **Send-buffer compression** | COMPR |
| **Request timeout** | TIMEOUT |

# 14 Displaying System Information

This chapter provides information on the commands that are available in the **Tools > System Information** menu.

## System Files

You can display the current assignments for all Natural system files (such as database ID and file number).

### To display the current system file assignments

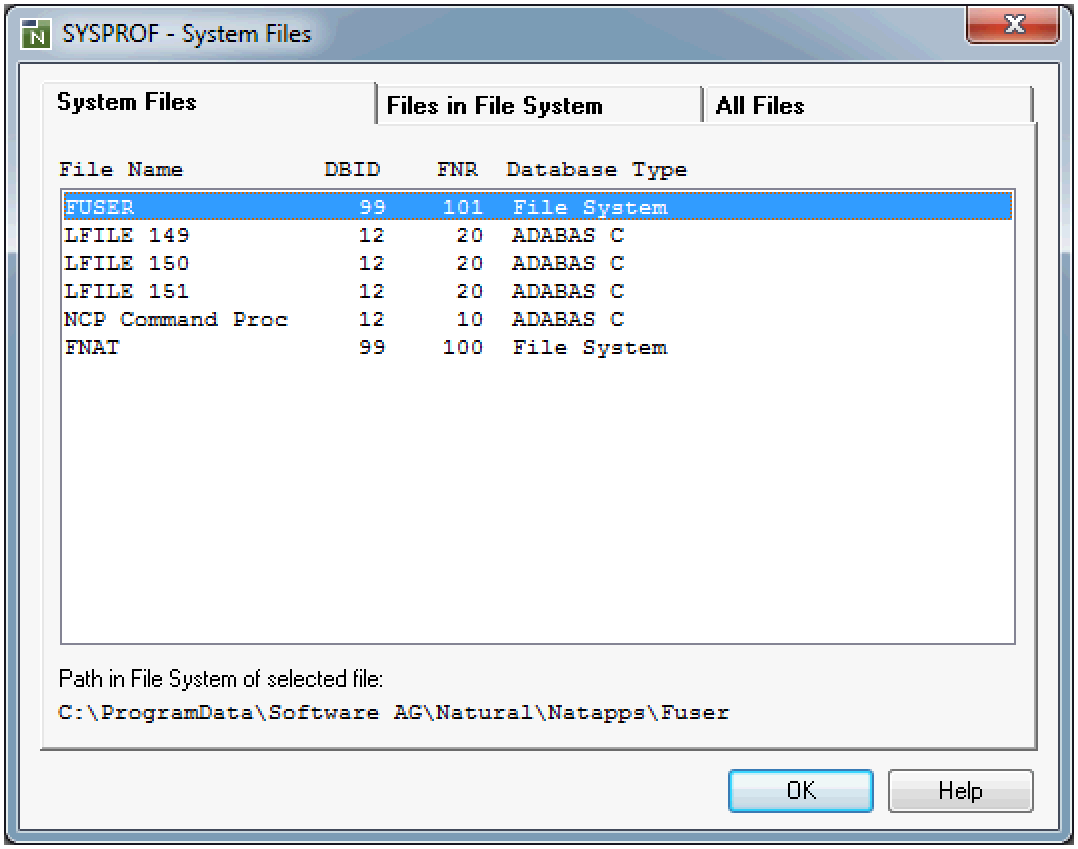
1 In the library workspace, select either the local environment or the remote environment.

2 From the **Tools** menu, choose **System Information > System Files**.

Or:

When the Tools toolbar is shown, choose the following toolbar button.

The assignments are shown in a dialog box. The content of this dialog box depends on the environment that is currently active.



```
File Name           DBID    FNR  Database Type
FUSER                 99    101   File System
LFILE 149             12     20   ADABAS C
LFILE 150             12     20   ADABAS C
LFILE 151             12     20   ADABAS C
NCP Command Proc      12     10   ADABAS C
FNAT                  99    100   File System
```

Path in File System of selected file:
`C:\ProgramData\Software AG\Natural\Natapps\Fuser`

See also the description of the system command `SYSPROF`.

## Work and Print Files

You can display work and print files information.

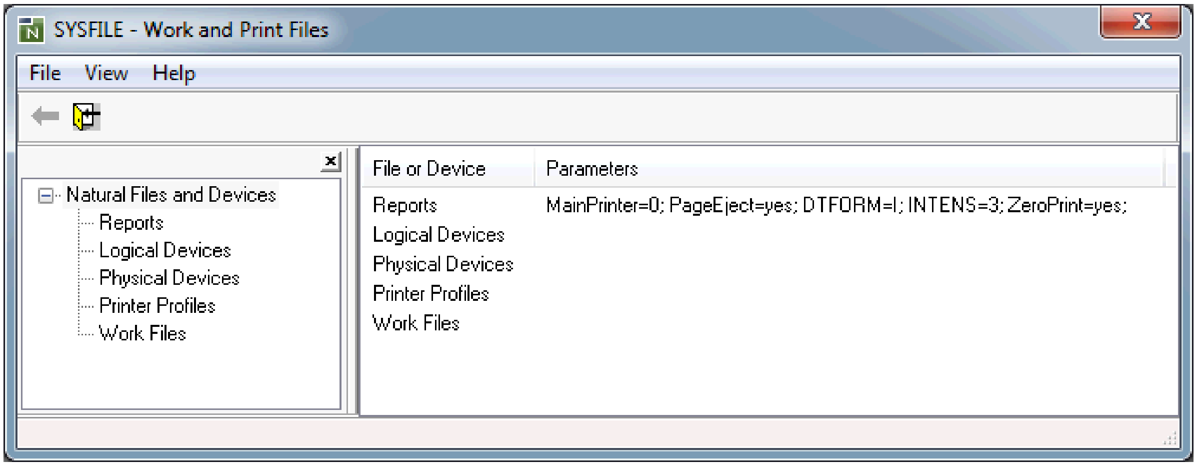
### To display work and print file information

1. In the library workspace, select either the local environment or the remote environment.
2. From the **Tools** menu, choose **System Information > Work and Print Files**.

   Or:

   When the Tools toolbar is shown, choose the following toolbar button.

The information is shown in a dialog box. The content of this dialog box depends on the environment that is currently active.



| File or Device | Parameters |
|---|---|
| Reports | MainPrinter=0; PageEject=yes; DTFORM=I; INTENS=3; ZeroPrint=yes; |
| Logical Devices | |
| Physical Devices | |
| Printer Profiles | |
| Work Files | |

3 In the tree, which is shown on the left, select or expand a node to display the corresponding information. For example, select **Work Files** to display information about the defined work files.

See also the description of the system command `SYSFILE`.

4 Optional. From the **View** menu, choose **Product Workspace** to toggle the display of the tree.

## Last Message

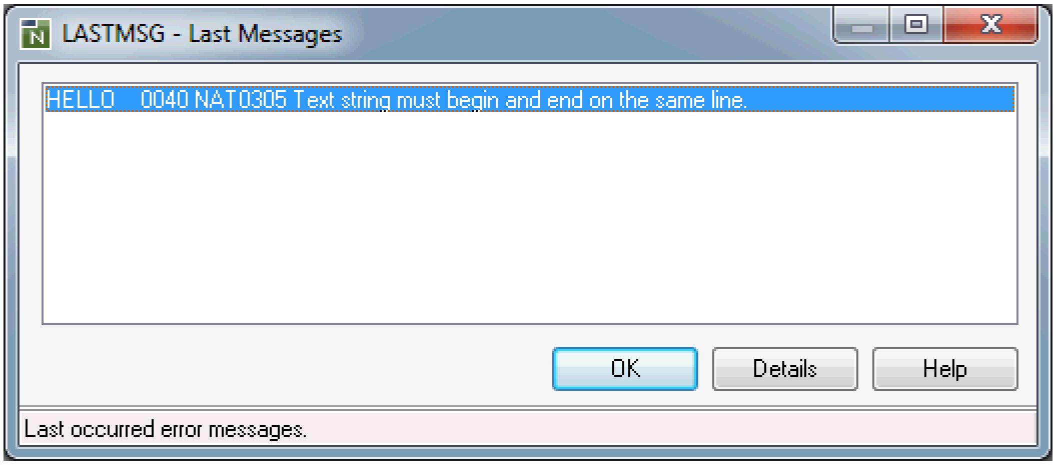
You can display additional information about the error situation which has occurred last in the active environment.

### To display information on the last error

1 In the library workspace, select either the local environment or the remote environment.

2 From the **Tools** menu, choose **System Information > Last Message**.

When an error has occurred previously in the selected environment, the corresponding error message is shown in a dialog box. Example:



HELLO 0040 NAT0305 Text string must begin and end on the same line.

Last occurred error messages.

For a remote environment, the environment name is also shown in the title bar.

3 Choose the **Details** button to display detailed information on the selected error in a dialog box.

See also the description of the system command `LASTMSG`.

## Technical Information

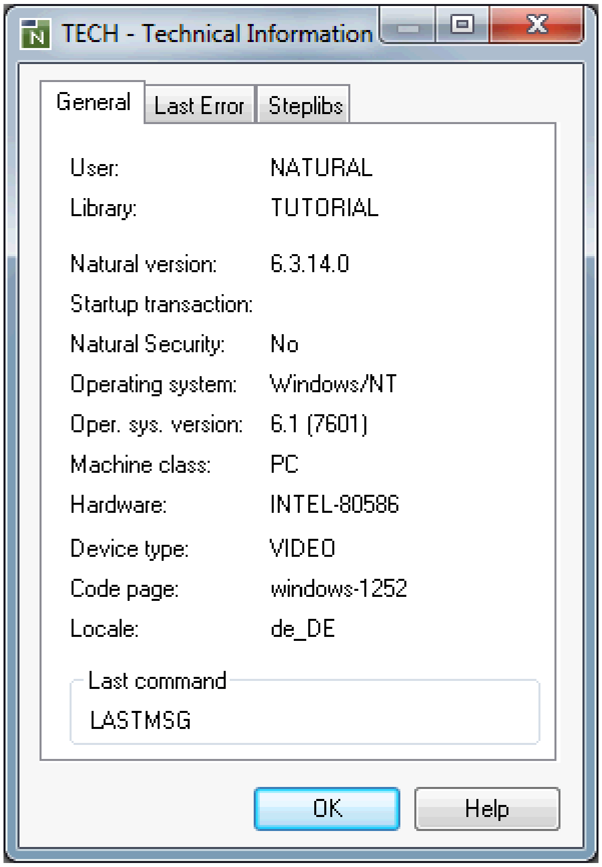
You can display technical information about the active environment (such as the operating system and the version number of the platform-specific Natural installation, or whether Natural Security is active).

### To display technical information

1 In the library workspace, select either the local environment or the remote environment.

2 From the **Tools** menu, choose **System Information > Technical Information**.

The information is shown in a dialog box.



User: NATURAL
Library: TUTORIAL

Natural version: 6.3.14.0
Startup transaction:
Natural Security: No
Operating system: Windows/NT
Oper. sys. version: 6.1 (7601)
Machine class: PC
Hardware: INTEL-80586
Device type: VIDEO
Code page: windows-1252
Locale: de_DE

Last command
LASTMSG

See also the description of the system command `TECH`.

## Product Information

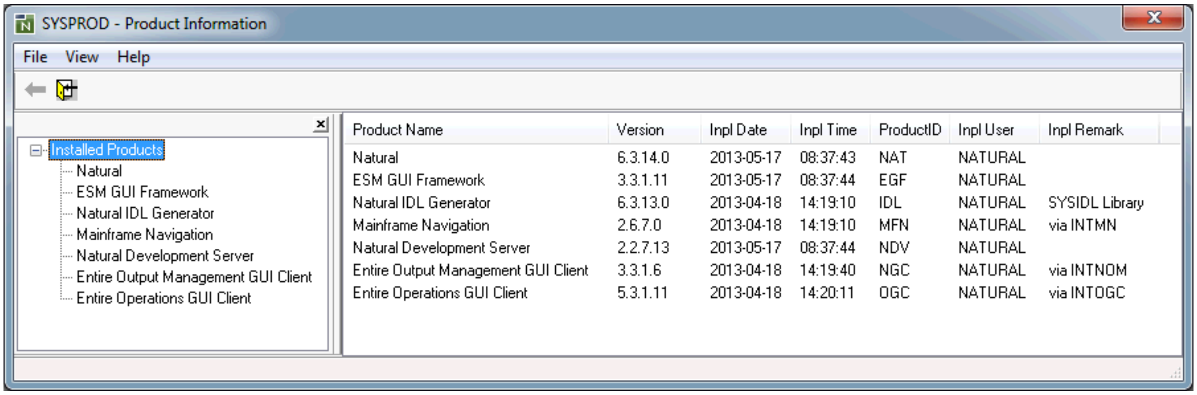
You can display a list of all products that are installed in the active environment.

### To display product information

1 In the library workspace, select either the local environment or the remote environment.

2 From the **Tools** menu, choose **System Information > Product Information**.

The information is shown in a window.


SYSPROD - Product Information
File
View
Help
Installed Products
Natural
ESM GUI Framework
Natural IDL Generator
Mainframe Navigation
Natural Development Server
Entire Output Management GUI Client
Entire Operations GUI Client

| Product Name | Version | Inpl Date | Inpl Time | ProductID | Inpl User | Inpl Remark |
|---|---|---|---|---|---|---|
| Natural | 6.3.14.0 | 2013-05-17 | 08:37:43 | NAT | NATURAL | |
| ESM GUI Framework | 3.3.1.11 | 2013-05-17 | 08:37:44 | EGF | NATURAL | |
| Natural IDL Generator | 6.3.13.0 | 2013-04-18 | 14:19:10 | IDL | NATURAL | SYSIDL Library |
| Mainframe Navigation | 2.6.7.0 | 2013-04-18 | 14:19:10 | MFN | NATURAL | via INTMN |
| Natural Development Server | 2.2.7.13 | 2013-05-17 | 08:37:44 | NDV | NATURAL | |
| Entire Output Management GUI Client | 3.3.1.6 | 2013-04-18 | 14:19:40 | NGC | NATURAL | via INTNOM |
| Entire Operations GUI Client | 5.3.1.11 | 2013-04-18 | 14:20:11 | OGC | NATURAL | via INTOGC |

3 Optional. From the **View** menu, choose **Product Workspace** to toggle the display of the tree which is shown on the left.

See also the description of the system command `SYSPROD`.

# 15 Using Development Tools

This chapter provides information on the commands that are available in the **Tools > Development Tools** menu.

When a plug-in (such as Object Description) has been activated, an additional command for this plug-in is available at the bottom of this menu. See the documentation for the corresponding plug-in for detailed information.

## Component Browser

Only available in the local environment.

Starts the Component Browser which is used to view ActiveX components which are available for developing NaturalX applications. See *Component Browser* in the *Tools and Utilities* documentation for detailed information.

## Data Browser

Starts the data browser wizard which is used to create a report from a file. See *Data Browser* in the *Tools and Utilities* documentation for detailed information.

## Reporter

Only available in the local environment.

Starts the Reporter which is used to generate a report from a Natural program. See the online help which is provided with the Reporter for detailed information.

**Note:** The Natural Reporter is not part of the Natural installation. For Natural for Windows customers, it is only available for download.

## Error Messages

Starts the `SYSERR` utility which is used to write your own application-specific messages. See *SYSERR Utility* in the *Tools and Utilities* documentation for detailed information.

## Object Handler

Starts the Object Handler which is used to process Natural and non-Natural objects for distribution. See *Object Handler* in the *Tools and Utilities* documentation for detailed information.

## Mainframe Navigation

Only available in a remote environment.

Starts Mainframe Navigation which is used to access and manipulate objects stored on a mainframe. See the Mainframe Navigation documentation, which is available separately, for detailed information.

## Application Programming Interfaces

Starts the `SYSEXT` utility which is used to list the Natural application programming interfaces contained in the library `SYSEXT`. See *SYSEXT - Natural Application Programming Interfaces* in the *Tools and Utilities* documentation for detailed information.

## Unlock Objects

Only available in a remote environment.

Displays a dialog in which you can view the Natural objects or documentation objects that are locked. If required, you can unlock them. See *Unlocking Objects Manually* in the *Remote Development Using SPoD* documentation.

# 16 Using Configuration Tools

This chapter provides information on the commands that are available in the **Tools > Configuration Tools** menu.

**Note:** A separate application, the Configuration Utility, is also available. See the *Configuration Utility* documentation for further information.

## Compiler Options

You can set options that are to be evaluated when a Natural object is compiled.

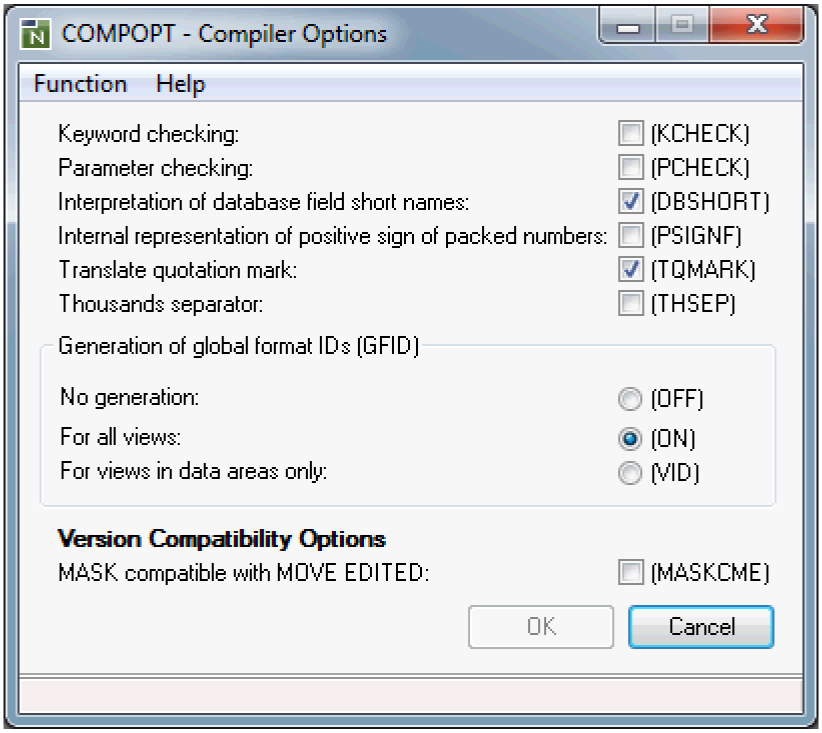
**Note:** See the description of the system command `COMPOPT` for further possibilities of specifying compiler parameters.

### To set the compiler options

1 In the library workspace, select either the local environment or the remote environment.

2 From the **Tools** menu, choose **Configuration Tools > Compiler Options**.

The compiler options that can be set are shown in a dialog box. The content of this dialog box depends on the environment that is currently active.



Keyword checking: ☐ (KCHECK)
Parameter checking: ☐ (PCHECK)
Interpretation of database field short names: ☑ (DBSHORT)
Internal representation of positive sign of packed numbers: ☐ (PSIGNF)
Translate quotation mark: ☑ (TQMARK)
Thousands separator: ☐ (THSEP)

Generation of global format IDs (GFID)
No generation: ○ (OFF)
For all views: ● (ON)
For views in data areas only: ○ (VID)

**Version Compatibility Options**
MASK compatible with MOVE EDITED: ☐ (MASKCME)

3 Set the required compiler options.

See the description of the system command `COMPOPT` for information on the options in this dialog box.

4 Optional: select one of the following commands in the **Function** menu:

- **Update Options**: applies your changes without closing the dialog box.
- **Set to Default**: reset the compiler options to the default values.

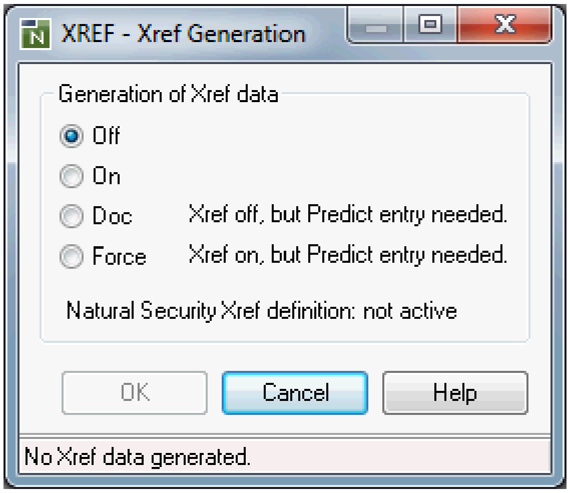
## XRef Generation

When Predict has been installed, you can specify whether active cross-reference (XRef) data is to be stored or not. See the Predict documentation for further information on active cross-references.

### To control the generation of XRef data

1 From the **Tools** menu, choose **Configuration Tools > XRef Generation**.

A dialog box appears.



XREF - Xref Generation

Generation of Xref data

● Off
○ On
○ Doc Xref off, but Predict entry needed.
○ Force Xref on, but Predict entry needed.

Natural Security Xref definition: not active

No Xref data generated.

2 Set the required option.

See the description of the system command `XREF` for information on the options in this dialog box.

## Remote Procedure Call

Starts the `SYSRPC` utility which is used to maintain remote procedure calls. See *SYSRPC Utility* in the *Tools and Utilities* documentation for detailed information.

## Plug-in Manager

The Natural Studio user interface is extensible by plug-ins. Part of the Natural Studio functionality itself is delivered in the form of plug-ins.

A plug-in needs not to be active and available in every Natural session and for every user. Which plug-ins are actually active and visible is configurable on a per user basis. The information which plug-ins are active for you is stored in your user profile in the registry. You activate and deactivate plug-ins using the Plug-in Manager.

The following topics are covered below:

- Enabling Plug-in Activation
- Invoking the Plug-in Manager
- Activating and Deactivating a Plug-in
- Defining Automatic or Manual Activation Mode for a Plug-in

**Note:** If you want to create your own plug-ins, see *Quick Start* in the documentation *Extending Natural Studio with Plug-ins*. This documentation also contains information on the plug-in example which can be activated in the Plug-in Manager.

### Enabling Plug-in Activation

The activation of plug-ins can be entirely disabled and enabled on a per user basis. In order to work with plug-ins, you must make sure that plug-in activation has been enabled in the **workspace options**.

### Invoking the Plug-in Manager

When plug-in activation has been enabled, you can invoke the Plug-in Manager as described below.

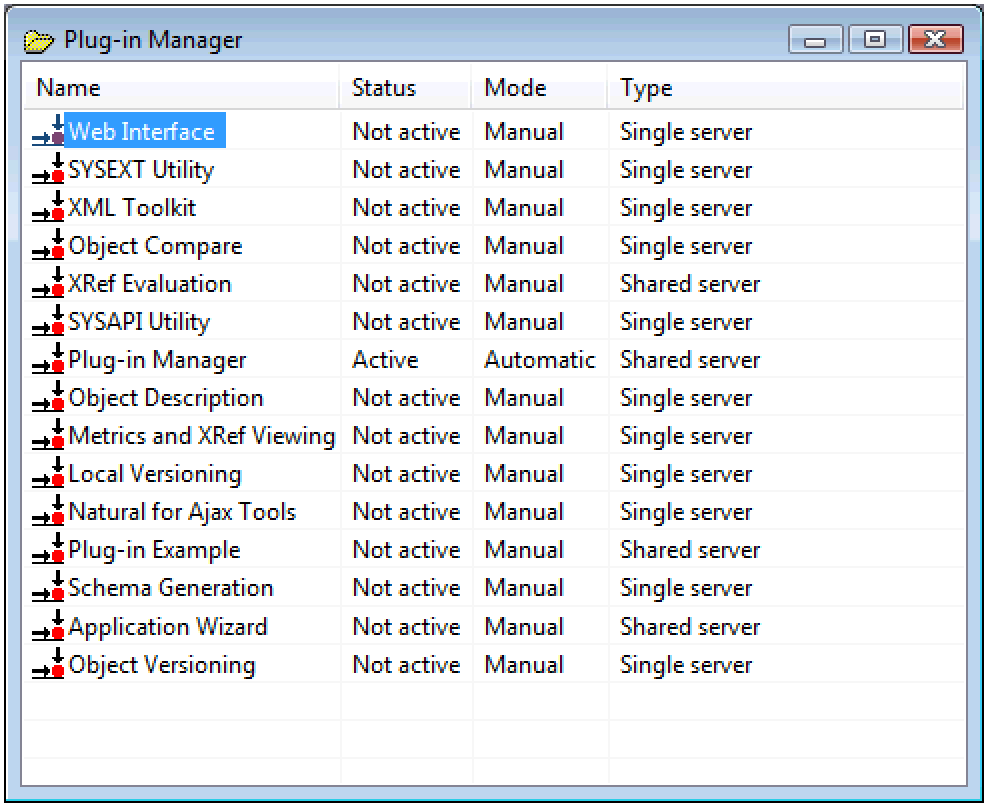
**To invoke the Plug-in Manager**

- From the **Tools** menu, choose **Configuration Tools > Plug-in Manager**.

  Or:

When the Tools toolbar is shown, choose the following toolbar button.

The **Plug-in Manager** window appears. For example:



| Name | Status | Mode | Type |
|---|---|---|---|
| Web Interface | Not active | Manual | Single server |
| SYSEXT Utility | Not active | Manual | Single server |
| XML Toolkit | Not active | Manual | Single server |
| Object Compare | Not active | Manual | Single server |
| XRef Evaluation | Not active | Manual | Shared server |
| SYSAPI Utility | Not active | Manual | Single server |
| Plug-in Manager | Active | Automatic | Shared server |
| Object Description | Not active | Manual | Single server |
| Metrics and XRef Viewing | Not active | Manual | Single server |
| Local Versioning | Not active | Manual | Single server |
| Natural for Ajax Tools | Not active | Manual | Single server |
| Plug-in Example | Not active | Manual | Shared server |
| Schema Generation | Not active | Manual | Single server |
| Application Wizard | Not active | Manual | Shared server |
| Object Versioning | Not active | Manual | Single server |

The Plug-in Manager (which is shown in a list view window) lists all installed plug-ins and shows their name, activation status, activation mode and type. It enables you to activate and deactivate installed plug-ins in order to configure your personal development environment.

**Note:** The type (single server or shared server) is defined when the plug-in is created. It cannot be modified here.

The following plug-ins are documented in this Natural for Windows documentation:

- Web Interface
- SYSEXT Utility
- XML Toolkit
- Object Compare
- SYSAPI Utility
- Local Versioning

- Natural for Ajax Tools
- Plug-in Example
- Application Wizard
- Plug-in Manager (see the information below in *Activating and Deactivating a Plug-in*)

Other plug-ins listed in the **Plug-in Manager** window are documented separately (that is, they are not part of this Natural for Windows documentation). In these cases, you have to refer to the corresponding product documentation for further information.

### Activating and Deactivating a Plug-in

The activation status of a plug-in is changed using a context menu. A plug-in which is not active can be activated, and a plug-in which is active can be deactivated.

The Plug-in Manager itself is implemented as a plug-in written in Natural. By definition, the Plug-in Manager itself cannot be dynamically activated. If plug-in activation is enabled in the workspace options, the command for **invoking** the Plug-in Manager is always available in Natural Studio. The Plug-in Manager is included in the list of plug-ins, where it can be deactivated like any other plug-in. But, of course, it can then not be reactivated manually in the same Natural Studio session, because in order to do so, you would need an active Plug-in Manager. However, as its activation mode is always "Automatic" (it cannot be changed to "Manual"), it will be activated again at the start of the next Natural Studio session if plug-in activation is enabled in the workspace options.

**Note:** The plug-ins for the utilities `SYSAPI` and `SYSEXT` are automatically activated when you invoke them with the corresponding command; therefore, you need not activate them manually using the Plug-in Manager.

#### To activate a plug-in

1. Select the plug-in in the **Plug-in Manager** window.
2. Invoke the context menu and choose **Activate**.

#### To deactivate a plug-in

1. Select the plug-in in the **Plug-in Manager** window.
2. Invoke the context menu and choose **Deactivate**.

### Defining Automatic or Manual Activation Mode for a Plug-in

The activation mode of a plug-in is changed using a context menu. Automatic activation means that the plug-in is started each time you start Natural Studio. Manual activation means that the plug-in stays inactive when you start Natural Studio; if you want to work with the plug-in, you have to activate it as described above.

#### To cause the plug-in to be activated each time you start Natural Studio

1. Select the plug-in in the **Plug-in Manager** window.
2. Invoke the context menu and choose **Activation Mode > Automatic**.

#### To cause the plug-in to stay inactivate when you start Natural Studio

1. Select the plug-in in the **Plug-in Manager** window.
2. Invoke the context menu and choose **Activation Mode > Manual**.

# 17 Using Help

This chapter lists all commands that are available in the **Help** menu.

When a plug-in (such as Object Description) has been activated, an additional command for invoking help for this plug-in is available in the **Help** menu.

## Contents

In addition to the online documentation which is available at ***https://documentation.softwareag.com/***, the Natural documentation is also available as an HTML help file (this is a standard Windows online help file with the extension ".chm").

### To invoke online help

- From the **Help** menu, choose **Contents**.

  The overview page appears.

### To invoke context-sensitive help

- Choose the **Help** button in a dialog box.

  Or:

  Press F1.

  See also *Using Context-Sensitive Help* in the *Program Editor* section of the *Editors* documentation.

  Help on the current topic appears.

## System Commands

You can display help for all Natural system commands that can be entered in the command line. This is part of the online help.

### To display help for a system command

1. From the **Help** menu, choose **System Commands**.

   The overview page of the system commands documentation appears.
2. Choose a system command in the selection box.

   The page appears on which this system command is explained.

## Natural Errors

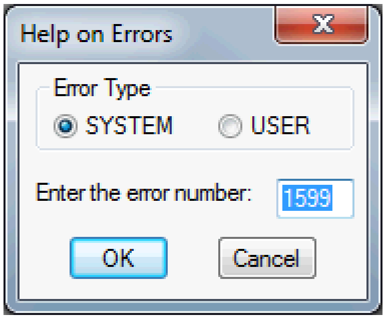
You can display a message text for an error number. This can either be a Natural system error message or a user-defined error message.

See also *Messages and Codes*.

### To display help on an error number

1 From the **Help** menu, choose **Natural Errors**.

The following dialog box appears.



Error Type
● SYSTEM ○ USER

Enter the error number: 1599

2 Select the option button for the required type of help message.

3 Enter the error number for which you want to display the message text.

The message text (long and short explanation) is shown in a dialog box.

## What's New

This cascading menu contains the following commands:

- Feature List
- Animated Demos

When a plug-in (such as Object Description) has been activated, an additional command for invoking “What's New?” information for this plug-in is also available.

### Feature List

You can display a list of new features in the current version. This list is part of the online help.

#### To display the new features

- From the **Help** menu, choose **What's New > Feature List**.

### Animated Demos

You can display animated demos of the new features in the current version.

#### To display animated demos

- From the **Help** menu, choose **What's New > Animated Demos**.

  An overview page appears listing the animated demos that can be started from this page.

## Software AG on the Web

You can invoke the company home page (*https://www.softwareag.com/*) directly from Natural Studio.

#### To invoke the Software AG home page

- From the **Help** menu, choose **Software AG on the Web**.

## About Natural for Windows

You can display information about Natural for Windows in a dialog box (for example, the version number).

#### To display information about Natural for Windows

- From the **Help** menu, choose **About Natural for Windows**.

# 18 Shortcut Keys

Many Natural commands are mapped to shortcut keys.

This chapter lists the default assignments of the shortcut keys which can be changed by the user.

You can define additional shortcut key assignments or overwrite the default assignments on the **Keyboard** page of the **Customize** dialog box. See *Customizing Natural Studio* for further information.

## Generally Available Shortcut Keys

The following table lists the default shortcut keys that are available anywhere within Natural Studio where applicable (provided they have not been overwritten in the **Customize** dialog box).

| Shortcut Key | Description |
|---|---|
| ALT+1 | Toggle display of library workspace. |
| ALT+2 | Toggle display of application workspace. |
| ALT+3 | Toggle display of command line. |
| ALT+4 | Toggle display of status bar. |
| ALT+5 | Toggle display of results window. |
| ALT+6 | Toggle display of debugger window. |
| ALT+ENTER | Display properties of selected object. |
| ALT+F4 | Exit Natural session. |
| ALT+F7 | Go to next event (debugger). |
| ALT+NUM* | Show trace position (debugger). |
| CTRL+A | Select all. |
| CTRL+C | Copy to clipboard. |
| CTRL+D | Toggle between command line and active list view or editor window. |
| CTRL+E | Check current object. |
| CTRL+F | Find in editor. |
| CTRL+G | Go to specific line in editor. |
| CTRL+H | Replace in editor. |
| CTRL+L | Toggle between application workspace and active list view or editor window. |
| CTRL+O | Open Natural object or list view window. |
| CTRL+P | Print current object. |
| CTRL+N | Create new program. |
| CTRL+R | Run current object. |
| CTRL+S | Save current object. |
| CTRL+T | Stow current object. |

| Shortcut Key | Description |
|---|---|
| CTRL+U | Toggle between results window and active list view or editor window. |
| CTRL+V | Paste from clipboard. |
| CTRL+W | Toggle between library workspace and active list view or editor window. |
| CTRL+X | Cut to clipboard. |
| CTRL+Y | Redo last undo action in editor. |
| CTRL+Z | Undo last action in editor. |
| CTRL+ALT+1 | Toggle display of variables window (debugger). |
| CTRL+ALT+2 | Toggle display of break- and watchpoints window (debugger). |
| CTRL+ALT+3 | Toggle display of call stack window (debugger). |
| CTRL+F3 | Replace next in editor. |
| CTRL+F4 | Close active list view or editor window. |
| CTRL+F7 | Start debugger. |
| CTRL+F11 | Step out of object (debugger). |
| CTRL+SHIFT+A | Save all objects. |
| CTRL+SHIFT+B | Go to break- and watchpoints window (debugger). |
| CTRL+SHIFT+C | Go to call stack window (debugger). |
| CTRL+SHIFT+D | Delete line in editor. |
| CTRL+SHIFT+K | Delete to end of line in editor. |
| CTRL+SHIFT+W | Add a watchpoint (debugger). |
| CTRL+SHIFT+T | Add a watchvariable (debugger). |
| CTRL+SHIFT+V | Go to variables window (debugger). |
| CTRL+SHIFT+F7 | Restart debugger. |
| CTRL+SHIFT+TAB | Activate previous list view or editor window. |
| CTRL+TAB | Activate next list view or editor window. |
| F1 | Display help. |
| F2 | Rename selected object. |
| F3 | Find next in editor. |
| F5 | Refresh. |
| F7 | Go to next watchpoint or breakpoint (debugger). |
| F10 | Step over another object (debugger). |
| F11 | Step into another object (debugger). |
| SHIFT+F7 | Stop debugger. |

## Program Editor Shortcut Keys

The following table shows the shortcut keys that are available in the active program editor, in addition to the default shortcut keys.

| Shortcut Key | Description |
|---|---|
| CTRL+B | Toggle expanded/collapsed state of block. |
| CTRL+J | Incremental find in editor. |
| CTRL+M | Add comment mark(s) to beginning of current line or to beginning of all selected lines. |
| CTRL+MINUS | Collapse current block. |
| CTRL+PLUS | Expand current block. |
| CTRL+SPACEBAR | Highlight text. |
| CTRL+ALT+B | Toggle bookmark. |
| CTRL+ALT+C | Convert from hexadecimal to character format. |
| CTRL+ALT+L | Clear bookmarks. |
| CTRL+ALT+N | Next bookmark. |
| CTRL+ALT+P | Previous bookmark. |
| CTRL+ALT+U | Convert characters of a text constant to hexadecimal Unicode representation. |
| CTRL+ALT+X | Convert characters of a text constant to hexadecimal code page representation. |
| CTRL+ALT+MINUS | Collapse current and all nested blocks. |
| CTRL+ALT+PLUS | Expand current and all nested blocks. |
| CTRL+F10 | Run to current cursor position (debugger). |
| CTRL+SHIFT+L | Translate selected text to lowercase. |
| CTRL+SHIFT+M | Remove comment mark(s) from beginning of current line or from beginning of all selected lines. |
| CTRL+SHIFT+P | Replay recording. |
| CTRL+SHIFT+R | Start recording. |
| CTRL+SHIFT+S | Stop recording. |
| CTRL+SHIFT+U | Translate selected text to uppercase. |
| F6 | Toggle between split windows. |
| F9 | Toggle breakpoint at current cursor position (debugger). |
| SHIFT+F9 | Modify variable at current cursor position (debugger). |

## Dialog Editor Shortcut Keys

The following table shows the shortcut keys that are available in the active dialog editor, in addition to the default shortcut keys.

| Shortcut Key | Description |
|---|---|
| ALT+RIGHT-ARROW | Unify spacing for selected controls horizontally. |
| ALT+UP-ARROW | Unify spacing for selected controls vertically. |
| CTRL+ALT+C | Open source code in program editor. |
| CTRL+ALT+E | Open event handlers dialog. |
| CTRL+ALT+G | Open global data area dialog. |
| CTRL+ALT+I | Open timer controls dialog. |
| CTRL+ALT+J | Open image list dialog. |
| CTRL+ALT+L | Open local data area dialog. |
| CTRL+ALT+M | Open menu editor dialog. |
| CTRL+ALT+N | Open signal controls dialog. |
| CTRL+ALT+O | Open comment dialog. |
| CTRL+ALT+P | Open parameter data area dialog. |
| CTRL+ALT+Q | Toggles control sequence definition mode on or off. |
| CTRL+ALT+S | Open inline subroutines dialog. |
| CTRL+ALT+T | Open toolbar dialog. |
| CTRL+ALT+W | Open wallpapers dialog. |
| CTRL+ALT+X | Open context menus dialog. |
| CTRL+ALT+DOWN-ARROW | Align selected controls down. |
| CTRL+ALT+LEFT-ARROW | Align selected controls left. |
| CTRL+ALT+RIGHT-ARROW | Align selected controls right. |
| CTRL+ALT+UP-ARROW | Align selected controls up. |
| CTRL+SHIFT+E | Open event handlers dialog for selected control. |
| CTRL+SHIFT+F9 | Align selected controls centered. |
| F9 | Center selected controls vertically. |
| SHIFT+F9 | Center selected controls horizontally. |

## Data Area Editor Shortcut Keys

The following table shows the shortcut keys that are available in the active data area editor, in addition to the default shortcut keys.

| Shortcut Key | Description |
|---|---|
| CTRL+M | Add comment mark(s) to beginning of current line or to beginning of all selected lines. |
| CTRL+ALT+L | Save column layout. |
| CTRL+SHIFT+I | Go to next level. |
| CTRL+SHIFT+J | Go to previous level. |
| CTRL+SHIFT+M | Remove comment mark(s) from beginning of current line or from beginning of all selected lines. |
| CTRL+PLUS | Best fit. |
| F2 | If a field row is selected, select the leftmost cell of this field. |
| SHIFT+SPACEBAR | If a single cell is selected, select the entire field row. |

## DDM Editor Shortcut Keys

The following table shows the shortcut keys that are available in the active DDM editor, in addition to the default shortcut keys.

| Shortcut Key | Description |
|---|---|
| CTRL+ALT+L | Save column layout. |
| CTRL+PLUS | Best fit. |
| F2 | If a field row is selected, select the leftmost cell of this field. |
| SHIFT+SPACEBAR | If a single cell is selected, select the entire field row. |

## Reserved Shortcut Keys

The following reserved shortcut keys cannot be modified in the **Customize** dialog box. They provide the same functionality as in Windows.

BACKSPACE

DEL

DOWN-ARROW

END

ENTER

ESC

F1

HOME

INS

LEFT-ARROW

PAGE-DOWN

PAGE-UP

RIGHT-ARROW

SPACEBAR

TAB

UP-ARROW

CTRL+DOWN-ARROW

CTRL+END

CTRL+HOME

CTRL+LEFT-ARROW

CTRL+RIGHT-ARROW

CTRL+UP-ARROW

SHIFT+F10

# 19 Rules and Naming Conventions

This section describes Natural-specific rules and naming conventions.

## Object Naming Conventions

This section describes the naming conventions that apply when saving and/or cataloging a Natural object in a Natural system file.

The name of a Natural object can be 1 to 8 characters (listed in the following table) where the *first* character must be one of the following:

- an upper-case alphabetical character
- a number sign (#)
- a plus sign (+)

If the first character is a number sign (#) or a plus sign (+), the name must consist of at least one additional character.

**Exception:**

The name of a Natural DDM can be 1 to 32 characters (listed in the following table) where the *first* character must be an upper-case alphabetical character.

The name of a Natural object can consist of the following characters:

| Character | ISO Character Name | Remark |
|---|---|---|
| A - Z | Latin capital letter A - Z | Upper-case alphabetical character |
| 0 - 9 | Digit zero - digit nine | Numeric character |
| - | Hyphen-minus | Hyphen |
| _ | Low line | Underscore |
| / | Solidus | Slash |
| @ | Commercial at | |
| $ | Dollar sign | |
| & | Ampersand | Only allowed in language codes<br>See also *Defining the Language of a Natural Object* in the *Programming Guide*. |
| # | Number sign | Hash sign |
| + | Plus sign | |

## Library Naming Conventions

This section describes the naming conventions that apply to a Natural library.

The name (ID) of a library can be 1 to 8 characters and must *not* start with "SYS". The prefix "SYS" is reserved for Natural system libraries.

A library name must start with an upper-case alphabetical character.

A library name should not end with an underscore.

A library name can consist of the following characters:

| Character | ISO Character Name | Remark |
|---|---|---|
| A - Z | Latin capital letter A - Z | Upper-case alphabetical character |
| 0 - 9 | Digit zero - digit nine | Numeric character |
| - | Hyphen-minus | Hyphen |
| _ | Low line | Underscore |

## Naming Conventions for User-Defined Variables

This section describes the naming conventions that apply to a user-defined variable:

- Length of Variable Names
- Limitations of Variable Names
- Characters Allowed in Variable Names
- First Character of Variable Names
- Case of Characters in Variable Names

For further information on user-defined variables, refer to the section *User-Defined Variables* in the *Programming Guide*.

### Length of Variable Names

The name of a user-defined variable can be 1 to 32 characters long.

You can use variable names of over 32 characters (for example, in complex applications where longer meaningful variable names enhance the readability of programs); however, only the first 32 characters are significant and must therefore be unique, the remaining characters will be ignored by Natural.

### Limitations of Variable Names

The name of a user-defined variable must *not* be a Natural reserved keyword.

Within one Natural program, you must *not* use the same name for a user-defined variable and a database field, because this might lead to referencing errors (see *Qualifying Data Structures* in the *Programming Guide*).

### Characters Allowed in Variable Names

The name of a user-defined variable can consist of the following characters:

| Character | ISO Character Name | Remark |
|---|---|---|
| A - Z | Latin capital and/or small letter A - Z | Upper-case and/or lower-case alphabetical character<br>Lower-case *not* allowed as the first character |
| 0 - 9 | Digit zero - digit nine | Numeric character |
| - | Hyphen-minus | Hyphen |
| _ | Low line | Underscore |
| / | Solidus | Slash |
| @ | Commercial at | |
| $ | Dollar sign | |
| & | Ampersand | |
| # | Number sign | Hash sign |
| + | Plus sign | Only allowed as the first character |

### First Character of Variable Names

The first character of the name must be one of the following:

| Character | ISO Character Name | Remark |
|---|---|---|
| A - Z | Latin capital letter A - Z | Upper-case alphabetical character |
| & | Ampersand | |
| # | Number sign | Hash sign |
| + | Plus sign | |

If the first character is a number sign (#), a plus sign (+) or an ampersand (&), the name must consist of at least one additional character.

Variables in a global data area (GDA) with a plus sign (+) as the first character must be defined at Level 1; see also *Global Data Area* in the *Programming Guide*. Other levels are only used in a redefinition.

A plus sign (+) as the first character of a name is only allowed for application-independent variables (AIVs) and variables in a global data area (GDA).

Names of AIVs must begin with a plus sign (+); see also *Defining Application-Independent Variables* in the description of the `DEFINE DATA` statement in the *Statements* documentation.

An ampersand (&) as the first character of a name is used in conjunction with dynamic source program modification (see the `RUN` statement in the *Statements* documentation), and as a dynamically replaceable character when defining processing rules; see the relevant description in the *Map Editor* documentation.

### Case of Characters in Variable Names

With Natural for Windows and Linux, lower-case characters entered as part of a variable name are internally converted to upper case.

**Caution:** If you use lower-case characters as part of the variable name, variable names must be unique regardless of their case.