

Natural

Remote Development Using SPoD

Version 9.1.4

October 2021

This document applies to Natural Version 9.1.4 and all subsequent releases.

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

Copyright © 1992-2021 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: NATWIN-NNATSP0D-914-20211013

Table of Contents

Preface	v
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Accessing a Remote Development Environment	5
Mapping a Development Server	6
Dynamically Changing the CICS Transaction Name when Starting a Session	8
Connecting to a Mapped Development Server	8
Disconnecting a Mapped Development Server	9
Unmapping a Development Server	10
3 Object Locking	11
Object Locking Concept	12
Locking and Unlocking of Local and Remote Objects	13
Unlocking Objects Manually	13
Natural Development Server Restrictions	15
4 Natural Web I/O Interface Client	17
About the Natural Web I/O Interface Client	18
Prerequisites for the Natural Development Server	20
Working with the Natural Web I/O Interface Client	20
Modifying Color and Field Attributes	21
5 Terminal Emulation	25
About Terminal Emulation	26
Prerequisites for the Natural Development Server	27
Working in a Terminal Emulation Screen	27
Displaying a Keypad	28
Transferring Data Using the Clipboard	28
Modifying the Font	31
Modifying the Color Scheme	32
Printing the Contents of a Terminal Emulation Screen	37
Terminal Emulation Keys	38
Using Non-Latin Character Sets	39
Translation Tables	39
6 Managing Applications	45
About the Application Workspace	46
Connecting to an Application Server	47
Creating a New Application	48
Linking Objects and Applications	52
Mapping and Unmapping Applications	57
Connecting and Disconnecting Base Applications	59
Displaying/Changing the Properties of the Application Server	60
Displaying/Changing the Properties of an Application	61
Copying, Renaming and Deleting Applications	63

Cataloging Objects in the Application Workspace	64
7 Using the Application Wizard	65

Preface

This documentation applies to the client side of Natural Single Point of Development (SPoD). It is complementary to the *Using Natural Studio* documentation in that it provides information on several features of Natural Studio which apply only when you are working in a remote development environment using SPoD.



Note: The Natural Studio plug-ins which can only be used in a remote development environment (such as Object Description which can be used with Predict) are *not* documented in this Natural for Windows documentation. See the Single Point of Development documentation, which is provided separately, for detailed descriptions of these plug-ins.

This documentation is organized under the following headings:

Accessing a Remote Development Environment How to use the library workspace to map to remote server environments.

Object Locking Describes the locking mechanism that is available in a remote development environment.

Natural Web I/O Interface Client How to use the Natural Web I/O Interface client which appears in a remote development environment to display non-GUI information which contains Unicode characters.

Terminal Emulation How to use the terminal emulation window which appears in a remote development environment to display non-GUI information. The terminal emulation window is not Unicode-enabled.

Managing Applications How to use the application workspace which is used to administer all Natural applications in the remote environment(s) in a hierarchical manner as a tree view.

Using the Application Wizard How to invoke the Application Wizard.

Using the SPoD utility protocol, it is possible to extend your Natural development environment with user-written tools. See the Single Point of Development documentation, which is provided separately, for detailed information.

SPoD-specific limitations and considerations are included in the Natural Development Server documentation for the corresponding platform. This documentation is also provided separately.

1

About this Documentation

■ Document Conventions	2
■ Online Information and Support	2
■ Data Protection	3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. 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

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <https://documentation.softwareag.com>.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at <https://empower.softwareag.com/>.

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#).

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

Software AG Tech Community

You can find documentation and other technical information on the Software AG Tech Community website at <https://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have Tech Community credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

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 Accessing a Remote Development Environment

■ Mapping a Development Server	6
■ Dynamically Changing the CICS Transaction Name when Starting a Session	8
■ Connecting to a Mapped Development Server	8
■ Disconnecting a Mapped Development Server	9
■ Unmapping a Development Server	10

Natural Studio's library workspace can also be used to administer the Natural system files of a remote server environment.

Mapping a Development Server

To perform remote development, you have to activate a remote Natural environment. You do this by connecting to the appropriate development server. Each development server provides all remote services (such as access or update) for a specific FUSER (Natural system file for user programs).

If you want to use a development server for the first time, you have to map it as described below. Once you have mapped a development server, a node for this development server session is automatically shown in the library workspace.

It is possible to map the same development server more than once, for example, if you want to have development-server sessions with different session parameters. To switch to another session, you simply select the corresponding node in your library workspace.

If you do not know the name and port number for your development server, ask your administrator.

➤ To map a development server

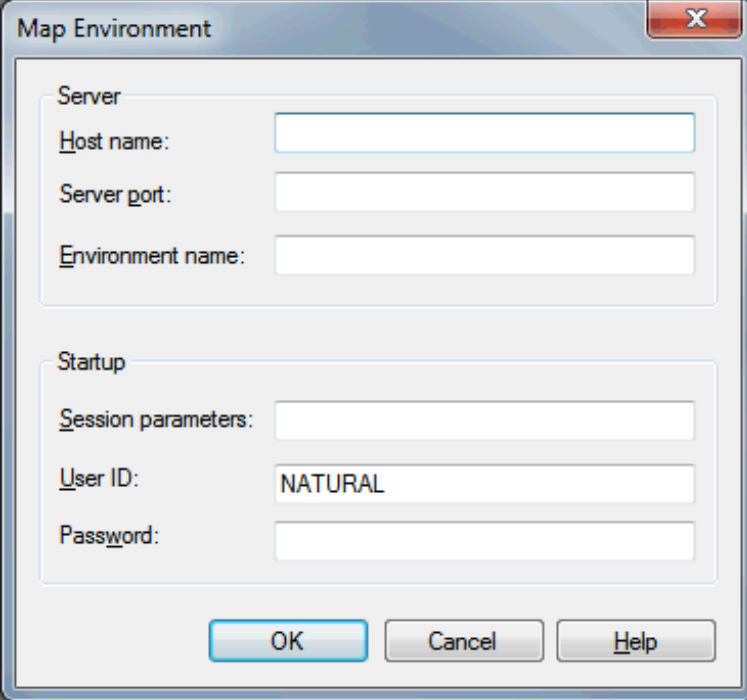
- 1 From the **Tools** menu, choose **Map > Environment**.

Or:

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



The **Map Environment** dialog box appears. Your user ID is automatically provided.

A screenshot of a 'Map Environment' dialog box. The dialog has a title bar with a close button (X). It contains two main sections: 'Server' and 'Startup'. The 'Server' section has three text boxes: 'Host name:', 'Server port:', and 'Environment name:'. The 'Startup' section has three text boxes: 'Session parameters:', 'User ID:', and 'Password:'. The 'User ID' text box contains the text 'NATURAL'. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

- 2 In the **Host name** text box, enter the name of the development server.
- 3 In the **Server port** text box, enter the TCP/IP port number of the development server.
- 4 Optionally. In the **Environment name** text box, enter the name that is to appear in the tree. If you leave this text box blank, a combination of server name and port number is shown in the tree.
- 5 If dynamic parameters are required for your development server, specify them in the **Session parameters** text box. Otherwise, leave this text box blank.
- 6 If you want to map the development server using a different user ID, specify it in the **User ID** text box. This is useful, for example, when Natural Security is active on the development server and administrator rights are to be used.
- 7 If Natural Security is active on the development server, specify the required password in the **Password** text box. Otherwise, leave this text box blank.
- 8 Choose the **OK** button.

Once you have mapped a development server, a node for this development server session is shown in the library workspace.

When the connection has been established, all libraries (according to the security profile) for this session are shown in the library workspace. You are automatically logged on to your default library.

The command line now shows the name of the library that is currently selected in the tree and the name of the active environment.



Note: For mapping to a development server using the command line, see the system command MAP.

Dynamically Changing the CICS Transaction Name when Starting a Session

The following description applies if you want to switch to a different CICS transaction on a mainframe.

You specify the CICS transaction name in the same text box in which you also specify the dynamic parameters for the Natural environment. So that the CICS transaction name can be evaluated, it is important that you specify it before any Natural parameters, using the following syntax:

```
<TA_NAME=name>
```

where *name* can be 1 to 4 characters long. This must be the name of an existing CICS transaction which applies to a CICS Adapter. It will override the transaction name which is currently defined in the configuration file for the CICS Adapter on the Natural Development Server (NDV). Ask your administrator for further information.

Make sure to put the entire definition in angle brackets. When this definition is followed by a Natural parameter, insert a blank before the Natural parameter. Example:

```
<TA_NAME=NA82> STACK=(LOGON SYSCP)
```

If the specified CICS transaction name cannot be found, an error message occurs and the session cannot be started.

Connecting to a Mapped Development Server

When you connect a **disconnected** development server, the connection to the corresponding server session is re-established.



Notes:

1. When you invoke Natural Studio, a **Map Environment** dialog box (see the description below) appears for each development server which was active (that is: connected) when you terminated Natural Studio. You can then decide whether you want to connect to this development server again.
2. When the development server has already been connected and you want to view the information that has been provided when the development server has been mapped, invoke the **Properties** dialog box. See *Properties for the Nodes* in the documentation *Using Natural Studio* for further information.

➤ **To connect a disconnected development server**

- 1 In the library workspace, select the node for a disconnected development server.
- 2 From the **Tools** menu, choose **Connect**.

Or:

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

Or:

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



The **Map Environment** dialog box appears. It shows the information that has been provided when the development server has been mapped.

- 3 If required, specify the Natural Security password in the **Password** text box. Otherwise, leave this text box blank.

You need not specify any other information.

- 4 Choose the **OK** button.

When the connection has been established, all libraries (according to the security profile) for this session are shown in your library workspace. You are automatically logged on to your default library.



Note: The **Connect** command is also available in the application workspace. See [Connecting and Disconnecting Base Applications](#).

Disconnecting a Mapped Development Server

When you disconnect a connected development server, the corresponding server session is shut down. All editor windows that have been opened for this server session are closed. The node for this development server is collapsed and the string "Disconnected" is shown to the right of the node name.

➤ **To disconnect a connected development server**

- 1 In the library workspace, select the node for a connected development server.
- 2 From the **Tools** menu, choose **Disconnect**.

Or:

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

Or:

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



Note: The **Disconnect** command is also available in the application workspace. See [Connecting and Disconnecting Base Applications](#).

Unmapping a Development Server

When you unmap a development server, its node is removed from the tree.

If the development server is still connected when unmapping it, the running session is shut down. All editor windows that have been opened for this server session are closed.

» To unmap a development server

- 1 In the library workspace, select the node for the development server that you want to unmap.
- 2 From the **Tools** menu, choose **Unmap**.

Or:

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

Or:

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



When the display of delete messages is enabled in the workspace options, a message box appears asking whether you really want to unmap the development server.



Notes:

1. For unmapping a development server using the command line, see the system command `UNMAP`.
2. The **Unmap** command is also available in the application workspace. See [Mapping and Unmapping Applications](#).

3

Object Locking

▪ Object Locking Concept	12
▪ Locking and Unlocking of Local and Remote Objects	13
▪ Unlocking Objects Manually	13
▪ Natural Development Server Restrictions	15

In a SPoD environment, Natural Studio provides a locking mechanism that prevents concurrent updating of Natural objects. These can be local Natural objects or objects accessed on a remote development server.

Object Locking Concept

When being edited in one of Natural Studio's editors or tools, the following Natural objects are locked:

Program
Subprogram
Subroutine
Copycode
Helproutine
Text
Dialog
Map
Class
Local data area
Global data area
Parameter data area
DDM
Application (only remote)

The locks are set exclusively by the editors or tools. If the objects are locked by an editor, the locked objects are unlocked as soon as you close the editor. If the objects are locked by a tool (for example, when local classes are changed using the Class Builder), the locked objects are unlocked when all internal links to the object are released. When you terminate your Natural session, all existing locks of that session are released.

Within your Natural session, you can edit different sources from different libraries in parallel in multiple editor sessions. In this case, a lock exists for all sources involved. You cannot edit the same source in different editor sessions, neither when you are in a single Natural session nor when you are in multiple Natural sessions. If you are using tools like the Class Builder, the object can be changed from all views; all views show the current state of the object.

Moving, Deleting and Renaming Objects

When you move, delete or rename a Natural source object, the locking state of the object concerned is checked:

- When the source object is locked, command execution is rejected.
- When the source object is not locked, the command is executed.

Saving and Stowing Objects

You can only save or stow a source object when you have opened it in an editor and have thus locked it.

Locking and Unlocking of Local and Remote Objects

Local Objects

Locking of sources in the local environment does not require the existence of a SPoD environment. It is accomplished by Windows operating system functions.

Remote Objects

Remote objects are locked only if they are edited using one of Natural Studio's local editors. The locks are managed by the development server and are stored as user-specific data in the Natural Development Server file.

Unlocking Objects

Natural Studio releases the lock held on an object as soon as the editor is closed. In the local environment, the operating system manages the release of the lock. In the remote environment, this is done by the Natural Development Server.

Unlocking Remote Objects in the Case of an Error

When Natural Studio loses the connection to the server (normally due to a hardware error) and is therefore unable to send an unlock request to the server, objects remain locked on the server even though the editor sessions are closed. The objects remain locked until they are manually unlocked (see below).

Unlocking Objects Manually

You can view Natural objects or documentation objects that are locked. If required, you can unlock them.



Note: The **Unlock Objects** command removes only the lock entry from the Natural Development Server file. In case of a Windows development server, the lock in the operating system remains until the corresponding editor window is closed.

> **To unlock a locked object**

- 1 In the library workspace, select a remote environment.



Note: In the local environment, unlocking of documentation objects is also possible, provided that Predict Version 4.4 or above is installed on Windows.

- 2 Optional. Select the library containing the locked object.

The library name will then be provided in the dialog box which is used for unlocking.


- 3 From the **Tools** menu, choose **Development Tools > Unlock Objects**.

Or:

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



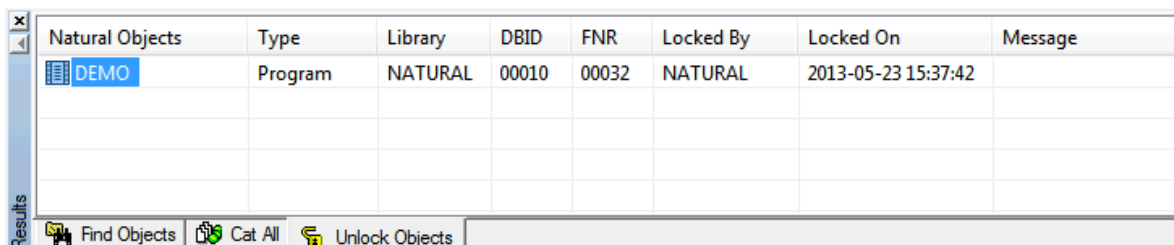
A dialog box appears. The name of the library that is currently selected in library workspace is automatically provided.

 **Note:** Depending on the settings in the remote environment, additional options may be shown in the dialog box. See also *Locking of Source Objects* in the *Editors* documentation of Natural for Mainframes.

See the description of the system command `UNLOCK` for detailed information on the options in this dialog box.

- 4 It is possible to choose the **OK** button without entering any information in this dialog box.

When locked objects exist, they are shown on the **Unlock Objects** tab of the results window. See also *Results Windows* in the documentation *Using Natural Studio*.



Natural Objects	Type	Library	DBID	FNR	Locked By	Locked On	Message
DEMO	Program	NATURAL	00010	00032	NATURAL	2013-05-23 15:37:42	

- 5 On the **Unlock Objects** tab, select the object that you want to unlock.
- 6 Invoke the context menu and choose **Unlock Objects**.

The object is still shown on the **Unlock Objects** tab. The **Message** column, however, indicates that the object has been unlocked.

 **Note:** If you want to remove the **Unlock Objects** tab from the results window, select any object on this tab, invoke the context menu and choose **Delete Tab**.

Natural Development Server Restrictions

The locking of source objects on a development server is supported only in a SPoD environment. The source objects are locked only when they are edited in Natural Studio.

To ensure correct object locking, you are strongly recommended to make changes on the same FUSER system file in a SPoD environment only. Avoid the use of the system command `READ` in a SPoD environment; it does not lock the object. Using the FUSER system file concurrently by another stand-alone, Natural session may lead to inconsistencies.

When working with Natural Studio, care must be taken to start all commands or utilities from within Natural Studio. It is not admissible to issue system commands in the terminal emulation window, for example at a `MORE` prompt or in a command line. In such a case, object locking will

not be supported. Particularly, editing of source objects directly with the server's editors can provoke inconsistencies.

On the development server, certain prerequisites apply for the locking of source objects. For detailed information on these prerequisites, see the Natural Development Server documentation.

4 Natural Web I/O Interface Client

■ About the Natural Web I/O Interface Client	18
■ Prerequisites for the Natural Development Server	20
■ Working with the Natural Web I/O Interface Client	20
■ Modifying Color and Field Attributes	21

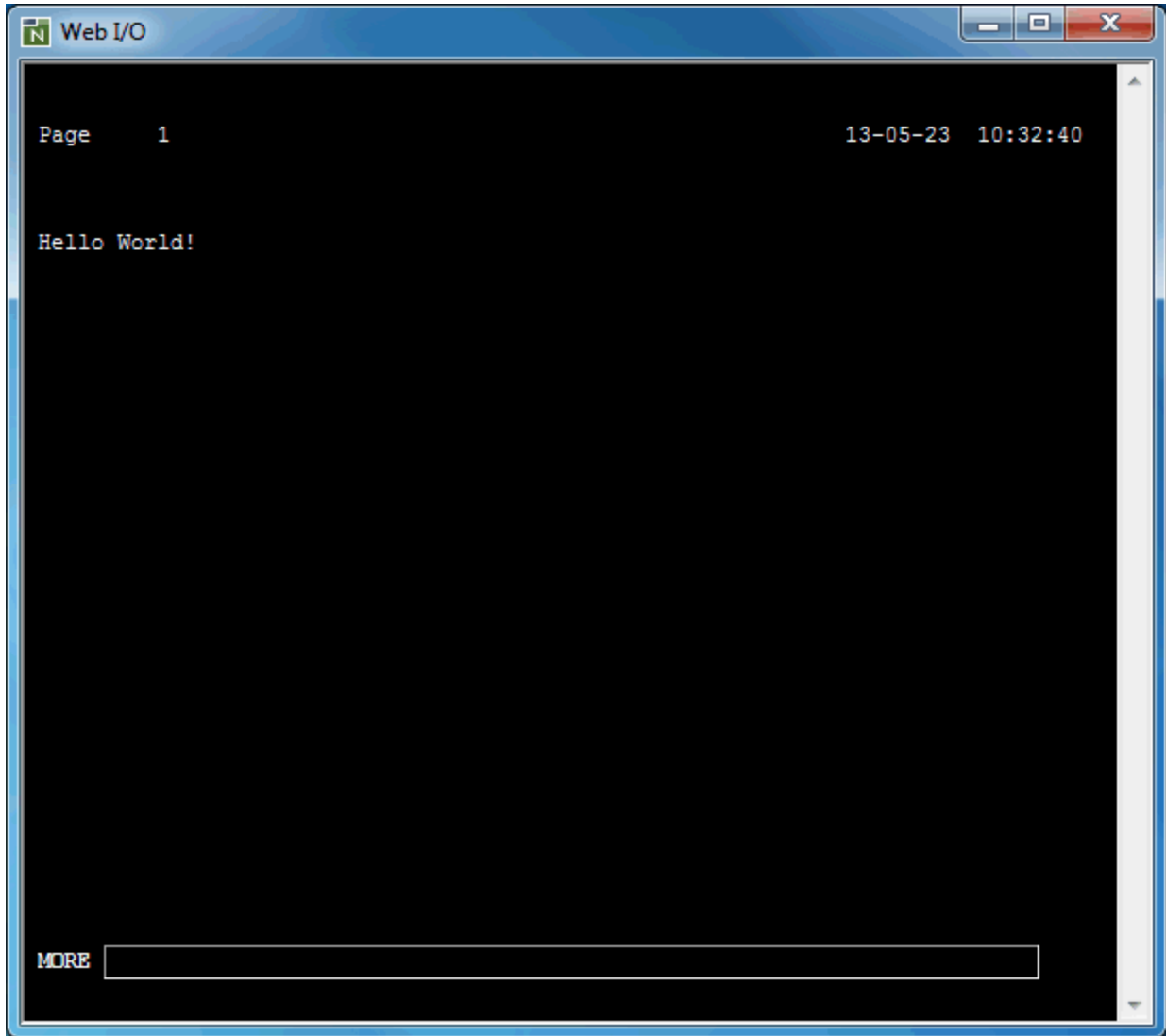
The Natural Web I/O Interface client is used in a remote development environment to display non-GUI information which contains Unicode characters. This is different from the local environment where the output is shown in an output window.

See also *Differences between Natural Web I/O Interface Client and Terminal Emulation* in the *Natural Web I/O Interface* documentation.

About the Natural Web I/O Interface Client

When specific parameters have been set on the Natural Development Server (see [below](#)), the Natural Web I/O Interface client is used instead of the [terminal emulation window](#) (which is not Unicode-enabled).

The following example shows a Natural Web I/O Interface client containing the output of a program that has been executed in a remote environment.



The Natural Web I/O Interface client enables you to test a complete application, including all of its character output, without having to leave Natural Studio. A screen output of an executed program automatically causes the Natural Web I/O Interface client to be opened and to be closed again when the program terminates (logically, the development server session reaches the NEXT level).

Utilities for which Natural Studio does not offer a graphical user interface in the remote environment are also made available by the Natural Web I/O Interface client. Entering the corresponding system command in Natural Studio's command line automatically opens a Natural Web I/O Interface client.

When you close the Natural Web I/O Interface client manually, this corresponds to pressing `ESC` in the application.

Prerequisites for the Natural Development Server

So that the Natural Web I/O Interface client can be invoked, the Natural Development Server (NDV) has to be configured as follows:

- **UNIX**

If you want to use the Natural Web I/O Interface client in a remote UNIX environment, the profile parameter `WEBIO` must be set to `ON` on the NDV server. See *Configuration Utility* in the Natural for UNIX documentation.

- **Mainframe**

If you want to use the Natural Web I/O Interface client in a remote mainframe environment, the NDV configuration parameter `TERMINAL_EMULATION` must be set to `WEBIO` on the NDV server. See *NDV Configuration Parameters* in the corresponding Natural Development Server documentation. Moreover, the Web I/O terminal converter module `NATWEB` must be linked to the Natural nucleus. The Natural profile parameter `TMODEL` can be used to determine the user screen size.

- **Windows**

In a remote Windows environment, the Natural Web I/O Interface client is always used, regardless of the setting of the profile parameter `WEBIO`.

Working with the Natural Web I/O Interface Client

When the Natural output is shown in the Natural Web I/O Interface client, any defined PF keys are shown as buttons in the window (instead of the key prompting lines which are normally shown in the native UNIX or mainframe environment). You can either use these buttons or you can press the corresponding keys on your keyboard (for example, `F5` for `PF5`).



Note: A refresh function is not executed when you press `F5`.

You can enter your input as usual. You can also use the usual Windows copy-and-paste functionality.

Modifying Color and Field Attributes

In the Natural Web I/O Interface client, the color attributes for the Natural fields (foreground colors), the background color, and the rendering of the Natural field attributes are controlled using an XSLT file. The name of this XSLT file is *natunistyle.xsl*. After the installation of Natural for Windows, this file can be found in the *\Program Files (x86)\Common Files\Natural\V1\bin* folder.

The following topics are covered below:

- [Setting the Foreground Colors](#)
- [Setting the Background Color](#)
- [Setting the Reverse Attribute](#)
- [Setting the Underline and Blinking Attributes](#)
- [Setting the Cursive/Italic Attribute](#)
- [Setting the Intensified Attribute](#)

Setting the Foreground Colors

The foreground colors for the font are mapped in the following section of the XSLT file:

```
<!-- Map foreground colors -->
<xsl:variable name="forecol">
  <xsl:choose>
    <xsl:when test="@forecol='Red'">red</xsl:when>
    <xsl:when test="@forecol='Green'">LimeGreen</xsl:when>
    <xsl:when test="@forecol='Blue'">DodgerBlue</xsl:when>
    <xsl:when test="@forecol='White'">white</xsl:when>
    <xsl:when test="@forecol='Pink'">fuchsia</xsl:when>
    <xsl:when test="@forecol='Turquoise'">aqua</xsl:when>
    <xsl:when test="@forecol='Yellow'">yellow</xsl:when>
    <xsl:when test="@forecol='Black'">white</xsl:when>
    <xsl:when test="@forecol='Transparent'">
      <xsl:choose>
        <xsl:when test="@intensified='True'">aqua</xsl:when>
        <xsl:otherwise><xsl:value-of select="$defforecolor"/></xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise><xsl:value-of select="$defforecolor"/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

For example, the color attribute Red of the Natural screen is mapped to the CSS color "red":

```
<xsl:when test="@forecol='Red'">red</xsl:when>
```

It is also possible to map the color attributes to RGB values. For example:

```
<xsl:when test="@forecol='Red'">#FF0000</xsl:when>
```

For the color attribute `Transparent`, it is checked whether the attribute `intensified` is set. If this is true, the color "aqua" is used. If not, the global variable `$defforecolor`, which is defined at the beginning of the XSLT file, is used.

More information on CSS colors can be found in the internet under https://www.w3schools.com/css-ref/css_colors.asp.

Setting the Background Color

The background color is set with the global variable `defbackcolor`, which is defined at the beginning of the XSLT file.

```
<!-- Background color -->
<xsl:variable name="defbackcolor" select="'black'"/>
```

Setting the Reverse Attribute

The field attribute for reverse video is set in the following section of the XSLT file, which occurs directly after the section in which the foreground colors are mapped:

```
<!-- START SET FOREGROUND AND BACKGROUND COLORS FOR REVERSE ATTRIBUTE -->
<!-- Set foreground colors -->
<xsl:variable name="ForeCol">
  <xsl:choose>
    <xsl:when test="@reverse='True'">color:<xsl:value-of select="$defbackcolor"/></xsl:when>
    <xsl:otherwise>color:<xsl:value-of select="$forecol"/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<!-- Set background colors -->
<xsl:variable name="BackCol">
  <xsl:choose>
    <xsl:when test="@reverse='True'">background-color:<xsl:value-of select="$forecol"/></xsl:when>
    <xsl:otherwise>background-color:<xsl:value-of select="$defbackcolor"/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>
<!-- END SET FOREGROUND AND BACKGROUND COLORS FOR REVERSE ATTRIBUTE -->
```

When the reverse attribute is set to true, the foreground color is set to the value of `$defbackcolor` and the background color gets the value of the foreground color.

When the reverse attribute is not set to true, the foreground color gets the value of the mapped foreground color and the background color gets the value of `$defbackcolor`.

Setting the Underline and Blinking Attributes

The field attributes for underlined and blinking text are set in the following section of the XSLT file:

```
<!-- Set underline and blinking attributes -->
<xsl:variable name="TextDecoStyle">
  <xsl:choose>
    <xsl:when test="@underline='True'">text-decoration:underline;</xsl:when>
    <xsl:when test="@blinking='True'">text-decoration:blink;</xsl:when>
    <xsl:otherwise>text-decoration:normal;</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

Since the Natural Web I/O Interface client uses the Internet Explorer which does not support blinking text, blinking text will not be displayed in the browser window.

Setting the Cursive/Italic Attribute

The field attribute for cursive/italic text is set in the following section of the XSLT file:

```
<!-- Set cursive attribute -->
<xsl:variable name="FontStyle">
  <xsl:choose>
    <xsl:when test="@italic='True'">font-style:italic;</xsl:when>
    <xsl:otherwise>font-style:normal;</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

Setting the Intensified Attribute

The field attribute for intensified text is set in the following section of the XSLT file:

```
<!-- Set intensified attribute - draw text as bold -->
<xsl:variable name="FontWeight">
  <xsl:choose>
    <xsl:when test="@intensified='True'">font-weight:bolder</xsl:when>
    <xsl:otherwise>font-weight:normal;</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```


5

Terminal Emulation

■ About Terminal Emulation	26
■ Prerequisites for the Natural Development Server	27
■ Working in a Terminal Emulation Screen	27
■ Displaying a Keypad	28
■ Transferring Data Using the Clipboard	28
■ Modifying the Font	31
■ Modifying the Color Scheme	32
■ Printing the Contents of a Terminal Emulation Screen	37
■ Terminal Emulation Keys	38
■ Using Non-Latin Character Sets	39
■ Translation Tables	39

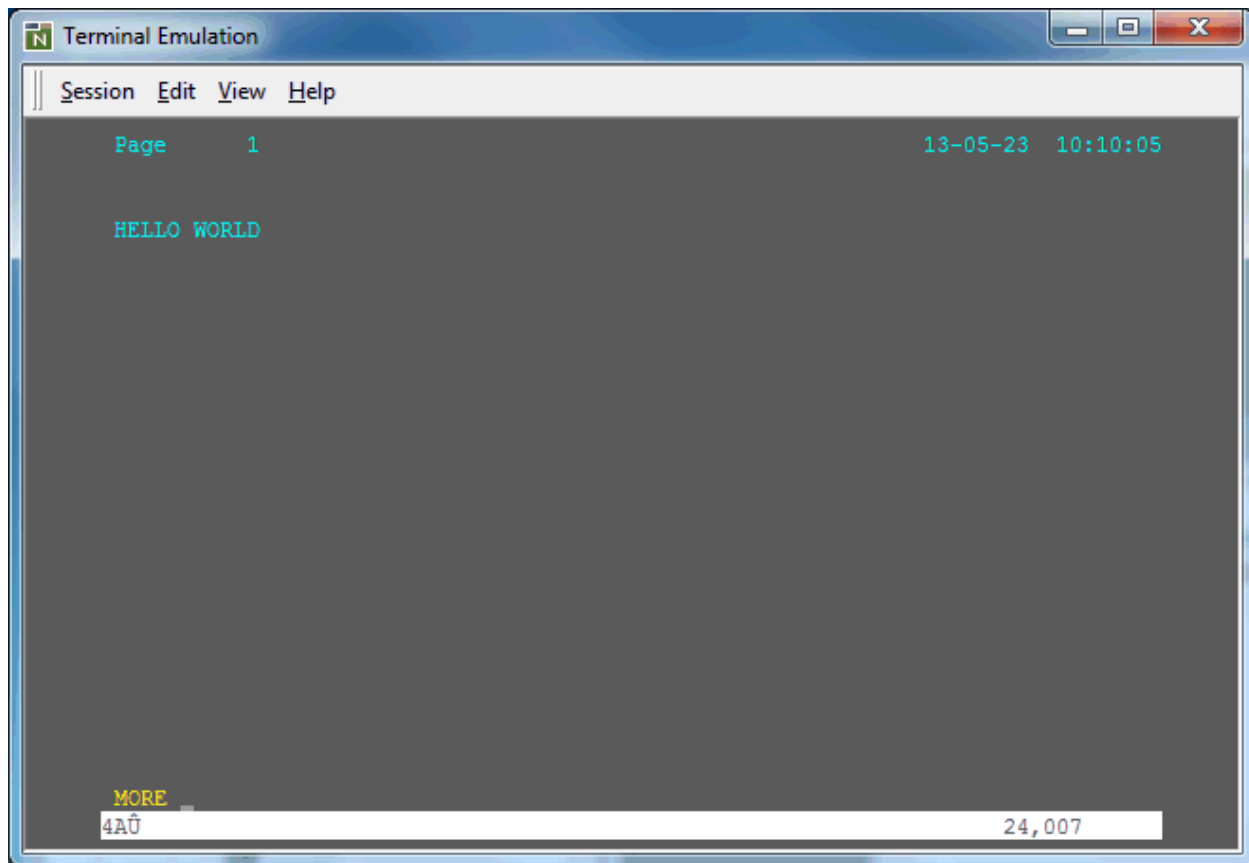
A terminal emulation window (which is not Unicode-enabled) is used in a remote development environment to display non-GUI information. This is different from the local environment where the output is shown in an output window.

Exception: In a remote Windows environment, the Natural Web I/O Interface client is always used instead of a terminal emulation window.

About Terminal Emulation

When specific parameters have been set on the Natural Development Server (see [below](#)), the terminal emulation window is used instead of the [Natural Web I/O Interface client](#).

The following example shows the terminal emulation window containing the output of a program that has been executed in a remote environment.



Terminal emulation enables you to test a complete application, including all of its character output, without having to leave Natural Studio. A screen output of an executed program automatically causes the terminal emulation window to be opened and to be closed again when the program terminates (logically, the development server session reaches the NEXT level).

The terminal emulation window cannot be closed manually using the standard Windows close button. As long as terminal emulation is active, you cannot work with Natural Studio.

Utilities for which Natural Studio does not offer a graphical user interface in the remote environment are also made available by terminal emulation. Entering the corresponding system command in Natural Studio's command line automatically opens the terminal emulation window.

Prerequisites for the Natural Development Server

So that the terminal emulation window can be invoked, the Natural development server (NDV) has to be configured as follows:

- **UNIX**

If you want to use the terminal emulation window in a remote UNIX environment, the profile parameter `WEBIO` must be set to "OFF" on the NDV server. See *Configuration Utility* in the Natural for UNIX documentation.

- **Mainframe**

If you want to use the terminal emulation window in a remote mainframe environment, the NDV configuration parameter `TERMINAL_EMULATION` must *not* be set to "WEBIO" on the NDV server. See *NDV Configuration Parameters* in the Natural Development Server documentation.

- **Windows**

In a remote Windows environment, the Natural Web I/O Interface client is always used instead of the terminal emulation window, regardless of the setting of the profile parameter `WEBIO`.

Working in a Terminal Emulation Screen

In the terminal emulation screen, you can use both mouse and keyboard. To change the cursor position, use the standard keys (e.g. `TAB` or the arrow keys) or the mouse. To use a host function key, press the **terminal emulation key** assigned to this function key. To simulate the `ENTER` key, double-click any position in the terminal emulation screen. Otherwise, you can use your keyboard to enter data as normal.

Displaying a Keypad

A keypad contains buttons that represent host function keys. You can display several keypads in the terminal emulation window. The following is an example of the keypad containing the 3270 keys.

CLR	ENTR	EEOF	ERI	INS	NLIN	HOME	PA1	PA2	PA3
-----	------	------	-----	-----	------	------	-----	-----	-----

➤ To switch the display of a keypad on and off

- From the **View** menu, choose *keypad-name*.

When a keypad is displayed, a check mark is shown next to its name.

Transferring Data Using the Clipboard

Using the commands from the **Edit** menu, you can copy or cut a terminal emulation screen (or part of it) and paste it somewhere else (for example, in another terminal emulation screen or in a PC application, such as, Microsoft Word). You can also copy or cut text from a PC application and paste it in a terminal emulation screen. Cut or copied data is stored on the Windows Clipboard.

You must first select the desired text before you can use the **Cut**, **Copy** or **Append Copy** command.

The following rules apply for a terminal emulation screen:

- You can only cut text from unprotected fields. If protected fields have been selected, they are copied.
- When you use the **Cut** or **Copy** command, the content of the clipboard is deleted and replaced with the new text. If you want to add data to the clipboard, you must use the **Append Copy** command.
- When you use the **Append Copy** command, the content of the clipboard is not deleted. The selected text is inserted behind already existing text on the clipboard. When you then choose the **Paste** command, the whole content of the clipboard is inserted at cursor position.
- You can only paste text into unprotected fields. If the field is not long enough, it is filled up to its maximum size and the remaining text is inserted in the next unprotected field(s). Example: the text to be pasted is "DISPLAY" and the cursor is located in a two character long input field. In this case, only the first two characters ("DI") are pasted into the input field.

Unless indicated otherwise, the left mouse button is always used. The *right* mouse button is used to select the **Edit** menu commands from a context menu.

➤ **To select part of a terminal emulation screen (mouse usage)**

- 1 Move the mouse pointer to the beginning of the text you want to select.
- 2 Press and hold down the left mouse button.

This clears any previous selection.

- 3 Drag the mouse until all desired text is selected.
- 4 Release the mouse button.

You can now cut, copy or append the selected text.

➤ **To select part of a terminal emulation screen (keyboard usage)**

- 1 Use the arrow keys to move the cursor to the beginning of the text you want to select.

This clears any previous selection.

- 2 Press and hold down **SHIFT**.
- 3 Use the arrow keys to select the desired text.
- 4 Release **SHIFT**.

You can now cut, copy or append the selected text.



Note: When holding down **CTRL+SHIFT** and then using the arrow keys, all text to the end of the line (with **RIGHT-ARROW**) or to the bottom of the screen (with **DOWN-ARROW**) is selected.

➤ **To select the entire content of the terminal emulation screen**

- From the **Edit** menu, choose **Select All**.

Or:

Press **CTRL+A**.

A box outline is shown around the terminal emulation screen.

You can now cut, copy or append the selected text.

➤ **To cancel the selection**

- Click another screen position.

Or:

When using the keyboard, just select another part of the terminal emulation screen as described above.

The previous selection is canceled.

➤ **To delete the selection in an unprotected field**

- From the **Edit** menu, choose **Delete**.

Or:

Press DEL.

Or:

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

The selection is deleted. It is not transferred to the clipboard.

➤ **To copy the selection and append it behind existing text in the clipboard**

- Invoke the context menu and choose **Append Copy**.

The selection is copied to the clipboard and appended behind already existing text.

➤ **To cut the selection and transfer it to the clipboard**

- From the **Edit** menu, choose **Cut**.

Or:

Press CTRL+X.

The selection is deleted and transferred to the clipboard. Exception: protected fields are not deleted; they are copied.

➤ **To copy the selection to the clipboard**

- From the **Edit** menu, choose **Copy**.

Or:

Press CTRL+C.

➤ **To paste the content of the clipboard into a terminal emulation screen**

- 1 Move the cursor to the position at which the text is to be inserted.

- 2 From the **Edit** menu, choose **Paste**.

Or:

Press CTRL+V.

The content of the clipboard is inserted at cursor position.

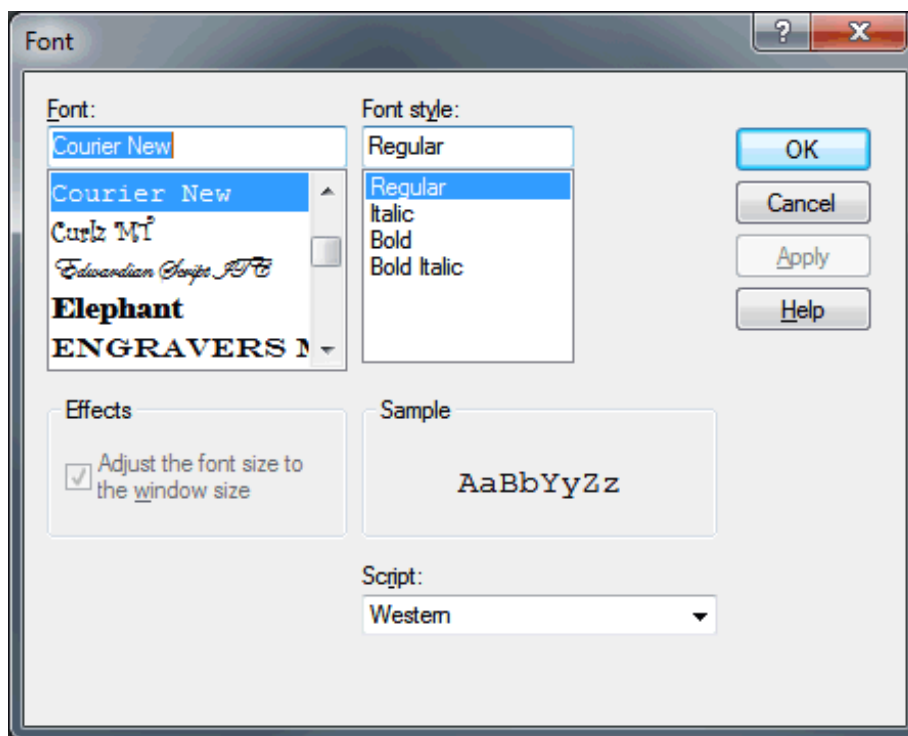
Modifying the Font

You can define another font to be used for terminal emulation.

> To modify the font

- 1 From the **Session** menu, choose **Font**.

The **Fonts** dialog box appears.



- 2 Define the desired font and font style.

The **Adjust the font size to the window size** check box applies to raster fonts only. For TrueType fonts, automatic resizing is always enabled. Select this check box to adjust the size

of the selected font to the window size. Adjusting the font size is done by increasing or reducing the spacing between characters and lines. This may result in truncated characters.

The **Script** drop-down list box contains a list of all available language scripts for the selected font. Make sure to specify a script that is compatible with the language that has been set up for your computer.

- 3 Choose the **OK** button.

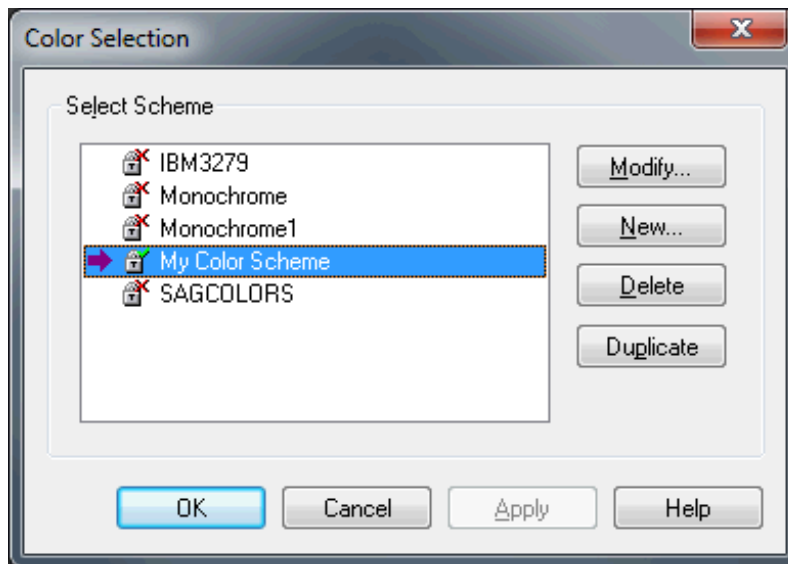
Modifying the Color Scheme

You can define another color scheme to be used for terminal emulation.

> To activate another color scheme

- 1 From the **Session** menu, choose **Color**.

The **Color Selection** dialog box appears.



The dialog box lists all color schemes that are available for the currently connected environment (for example, mainframe).

The active color scheme is indicated by an arrow. When the lock symbol next to the color scheme name shows a green check mark, the color scheme can be modified. When it shows a red cross, it cannot be modified; it can only be duplicated in this case.

- 2 Select the color scheme you want to activate.
- 3 Choose the **OK** button.

Command Buttons

Besides activating another color scheme, you can also create, modify, delete or duplicate a color scheme. To do so, use one of the following command buttons in the **Color Selection** dialog box:

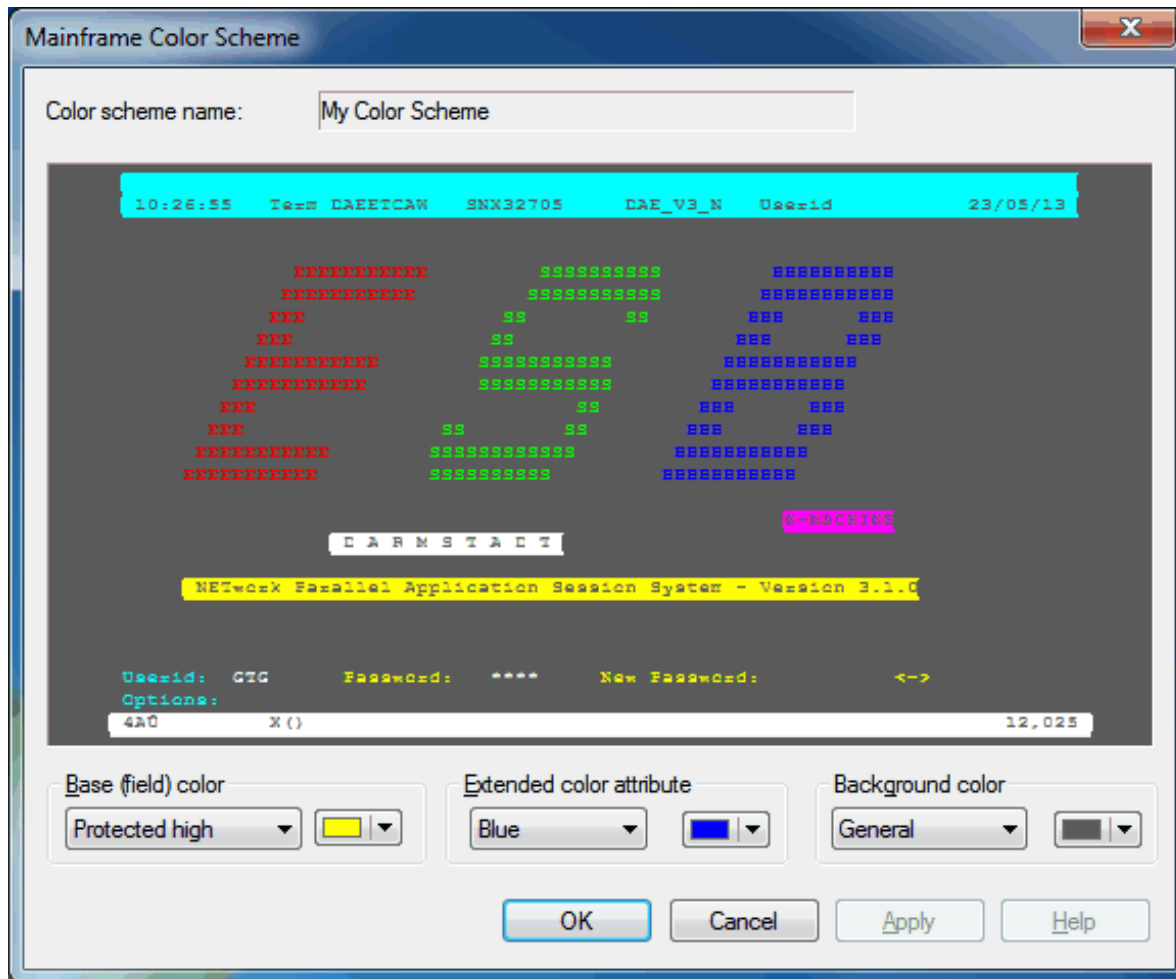
Command Button	Description
Modify	Modifies the selected color scheme. You can only modify the color scheme which is currently active. If you select another color scheme, you are asked whether you want to make the selected color scheme active in order to continue.
New	Creates a new color scheme. Specify a name for the new color scheme in the resulting dialog box and choose the OK button. You are then asked whether you want to make the new color scheme active in order to modify it.
Delete	Deletes the selected color scheme. You are asked to confirm the deletion. The predefined color schemes cannot be deleted.
Duplicate	Creates a copy of the selected color scheme. Specify a name for the copy in the resulting dialog box and choose the OK button. You are then asked whether you want to make the copy active in order to modify it.

Detailed information on the dialog box which appears when you create, modify or duplicate a color scheme is provided below. Depending on the currently connected environment, one of the following dialog boxes appears:

- [Mainframe Color Scheme](#)
- [VT Color Scheme](#)

The color scheme name can be up to 32 characters long and may contain blanks. It can only be specified when adding a new color scheme. Later, the name cannot be modified.

Mainframe Color Scheme



When you move the mouse pointer over the sample screen that is shown in the **Mainframe Color Scheme** dialog box, tooltips appear. The name of the base (field) color or extended color attribute is shown on the left of the tooltip. The name of the background color is shown on the right of the tooltip.

> To define other colors

- 1 Select a text entry from the first drop-down list box in one of the following group boxes:

- **Base (field) color**

The colors for the different field types in a terminal emulation screen.

- **Extended color attribute**

The extended colors sent by the host application. For example, you can specify that all fields with the extended color attribute "Green" are to appear blue in the terminal emulation screen.

- **Background color**

The background colors for specific field types. The background color underlays text. If the field contains no text, the field area is colored with the background color.

- 2 Select the desired color from the corresponding drop-down list box to the right.

This drop-down list box also provides the following buttons:

- **Other**

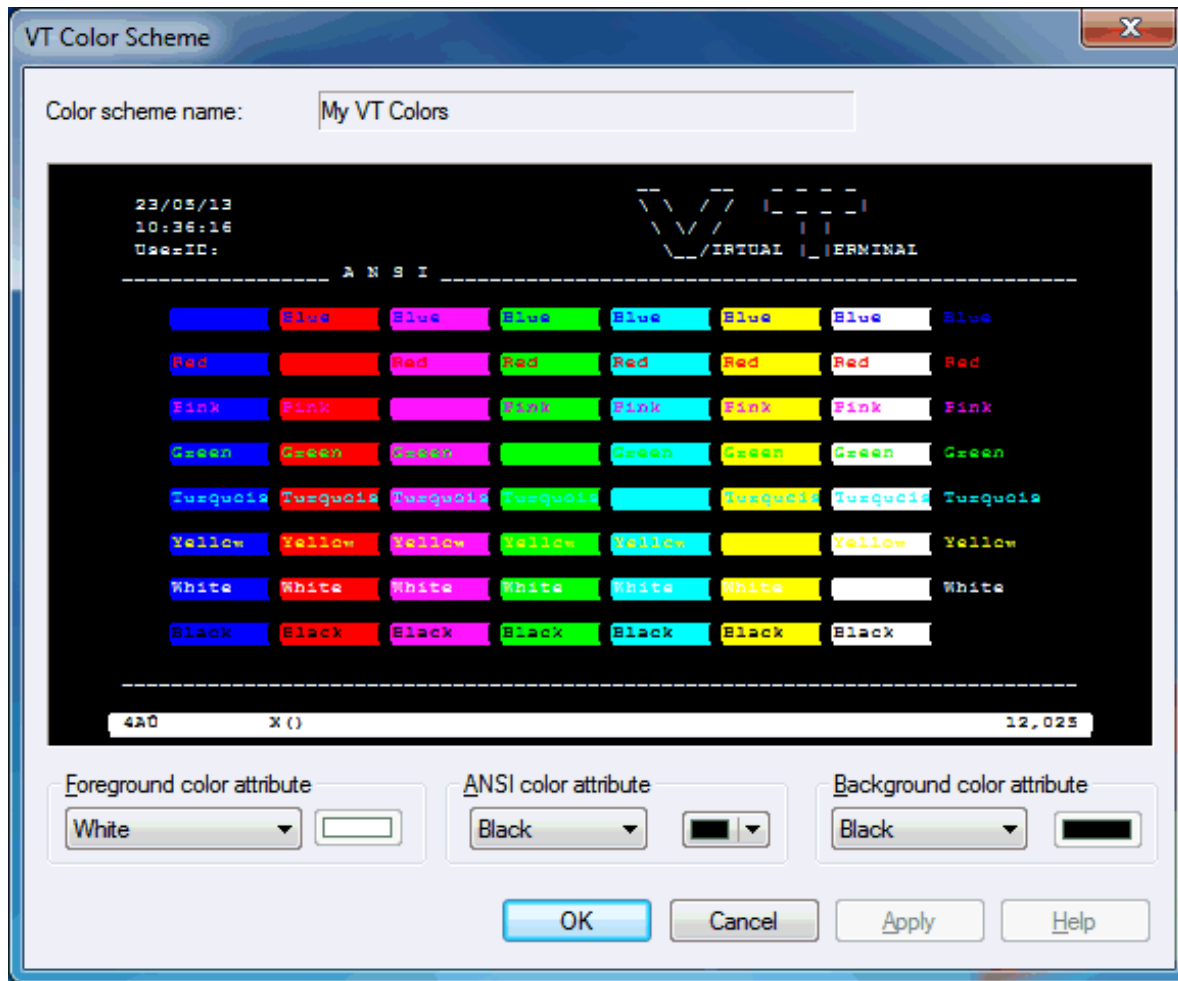
Lets you define additional colors.

- **Default**

Resets the modified color palette to the default colors.

- 3 Repeat the above steps until all desired colors have been defined. The **Mainframe Color Scheme** dialog box always reflects the current color settings.
- 4 Optional - choose the **Apply** button to save the current color settings without closing the dialog box.
- 5 Choose the **OK** button.

VT Color Scheme



When you move the mouse pointer over the sample screen that is shown in the **VT Color Scheme** dialog box, tooltips appear. The following applies for the ANSI color attributes: the name of the foreground color is shown on the left of the tooltip, and the name of the background color is shown on the right of the tooltip.

> To define other colors

- 1 Select a text entry from the first drop-down list box in one of the following group boxes:
 - **Foreground color attribute**
The foreground color for the terminal emulation screen. The VT terminal type 340+ uses this setting when starting the session and as long as no specific color commands are sent by the host. All other VT terminal types use this setting during the entire session.

■ **ANSI color attributes**

When the VT terminal type 340+ (color terminal) is set in Natural on the host, the host can send color commands for the foreground and background. In this mode, eight color attributes are available. For each possible character position or cursor position, the host can set a foreground and background color. You can modify the colors belonging to the color attributes (see below).

■ **Background color attribute**

The background color for the terminal emulation screen. The VT terminal type 340+ uses this setting when starting the session and as long as no specific color commands are sent by the host. All other VT terminal types use this setting during the entire session.

- 2 Select the desired color for an ANSI color attribute from the corresponding drop-down list box to the right.

This drop-down list box also provides the following buttons:

■ **Other**

Lets you define additional colors.

■ **Default**

Resets the modified color palette to the default colors.

- 3 Repeat the above steps until all desired colors have been defined.

The **VT Color Scheme** dialog box always reflects the current color settings.

- 4 Optional - choose the **Apply** button to save the current color settings without closing the dialog box.
- 5 Choose the **OK** button.

Printing the Contents of a Terminal Emulation Screen

You can print the contents of the terminal emulation screen on a printer that is defined under Windows.

➤ **To preview the screen to be printed**

- 1 From the **Session** menu, choose **Print Preview**.

The print preview window appears.

- 2 Optionally: use the **Zoom In** and **Zoom Out** buttons to view the information in this window. Or choose the **Print** button to invoke the **Print** dialog box.
- 3 To return to the terminal emulation screen, choose the **Close** button.

➤ To print the screen contents using a dialog box

- 1 From the **Session** menu, choose **Print**.

Or:

Press CTRL+P.

The **Print** dialog box appears.

- 2 Choose the **OK** button to print the current screen.

➤ To print the screen contents directly, without displaying a dialog box

- Press the PRINT SCREEN key.

The contents of the current terminal emulation screen are immediately printed. The current printer settings are used.

Terminal Emulation Keys

Keys such as PF1 or PA1 which start a terminal function can be used either by choosing the corresponding button in a **keypad** or by pressing the corresponding key combination.

The following table lists all available key combinations. When you have to press SHIFT or CTRL, this is indicated by an X. For example, when you want to use PF13, you have to press SHIFT+F1.

Terminal Keyboard	PC Keyboard		
3270 Key	SHIFT	CTRL	F-Key
CLR		X	F1
ENTR		X	F2
EEOF		X	F3
ERI		X	F4
INS		X	F5
NLIN		X	F6
HOME		X	F7
PA1		X	F8
PA2		X	F9
PA3		X	F10
PF1 through PF12			F1 through F12
PF13 through PF24	X		F1 through F12

Terminal Keyboard	PC Keyboard		
RESET		X	F11

Using Non-Latin Character Sets

By default, the Latin character set is used for terminal emulation (3270 translation).

You can define a different character set (e.g. Cyrillic) to be used for the terminal emulation. See *Workspace Options* in the documentation *Using Natural Studio* for further information.



Note: When you define another character set, you may have to modify the font of the terminal emulation. See [Modifying the Font](#).

Translation Tables

Translation tables convert characters when sending or receiving data to or from a host while using terminal emulation. Translation tables contain hexadecimal values which correspond to a national code page.

The following topics are covered below:

- [Translation Table Files](#)
- [Table Structure](#)
- [Adapting and Importing Translation Tables](#)
- [Adapting Translation Tables for Natural Development Server under BS2000](#)
- [Correcting a Wrong Conversion](#)

Translation Table Files

Each language setting has its own translation file. This file contains the character translation tables for translations from the PC to the host and vice versa. The file name reflects the communication mode. The extension of a translation file is ".tra".

In the case of a default installation, the translation files can be found in the following folder:

`<drive>:\Program Files (x86)\Common Files\Software AG\Natural\V1\etc`

The following translation files are available:

- `N3270_USER1.tra`
- `N3270_USER2.tra`
- `N3270_USER3.tra`

■ *N3270_USER4.tra*

These files give you the opportunity to create your own character translation tables.

Each file contains two translation tables where each table corresponds to one communication direction (one for sending and one for receiving). Example:

■ **ASC2***xxxxyy*

Translation table for PC-to-host communication.

■ *xxxxyy***2ASC**

Translation table for host-to-PC communication.

The table names contain the following information:

Part of Name	Description
ASC2 or 2ASC	Code page on the PC. Standardized name convention.
<i>xxx</i>	Code page on the host. Can be defined by the user.
<i>yy</i>	Language. Can be defined by the user.

Table Structure

The table for PC-to-host communication contains the hexadecimal values for the host code page. The row and column names correspond to the hexadecimal value of the PC code page.

The table for host-to-PC communication contains the hexadecimal values for the PC code page. The row and column names correspond to the hexadecimal value of the host code page.

The first set of characters of the hexadecimal value indicates the row and the second set indicates the column.

Example

The character "b" in the file *N3270_USER1.tra* has the following hexadecimal values:

Code Page (communication from)	Hexadecimal Value	Location
Host	82	Row 08, column 02
PC	62	Row 06, column 02

Adapting and Importing Translation Tables

Translation tables must be adapted in the following situations:

- When a character sent from the PC to the host (or vice versa) is translated or displayed differently than it was sent.
- When the tables lack special regional characters.

To adapt a translation table, use a code page which corresponds to the host or PC code page settings. You can obtain code pages as graphics from <https://msdn.microsoft.com/de-de>; look for a link to "References > Code Pages".



Important: Make backups of all ".tra" files before beginning.

When the translation tables have been adapted, you have to import them as described below. Entire Connection's Configuration Manager is used for importing the translation tables. With a default installation, the file *PCCAdmin.exe*, which invokes the Configuration Manager, is provided in the following Natural folder:

`<drive>:\Program Files (x86)\Common Files\Software AG\Natural\V1\Terminal\configmgr`



Note: Documentation for Entire Connection's Configuration Manager can be found at <http://documentation.softwareag.com/> (Empower login required).

» To import the adapted translation tables

- 1 Execute the file *PCCAdmin.exe* to start the Configuration Manager.
If user information is required, a logon dialog box appears
- 2 In the logon dialog, enter your user name and, if required, password. The default user name is "SYSTEM".
- 3 From the **File** menu choose **Open Share File**.
- 4 In the resulting dialog box, select the file *NATSHARE411.sag* and choose the **Open** button. This file is located in the following folder:

`<drive>:\Program Files (x86)\Common Files\Software AG\Natural\V1\etc`
- 5 In the left pane of the Configuration Manager, expand the node **Tables** and then select the node **Translation**.
- 6 From the **Edit** menu, choose **Import Table**.
- 7 In the resulting dialog box, select the translation table files that you want to import and choose the **Import** button.
- 8 Confirm that the translation tables for the user-defined files are to be overwritten.

Natural's user-defined files contain the following translation tables:

```
ASC2(Ansi)EBCDICUser1
ASC2(Ansi)EBCDICUser2
ASC2(Ansi)EBCDICUser3
ASC2(Ansi)EBCDICUser4
EBCDICUser1(Ansi)2ASC
EBCDICUser2(Ansi)2ASC
EBCDICUser3(Ansi)2ASC
EBCDICUser4(Ansi)2ASC
```

- 9 So that a translation table can be used in Natural, it has to be specified in Natural Studio's workspace options. See *Workspace Options* in the documentation *Using Natural Studio*.

Adapting Translation Tables for Natural Development Server under BS2000

For the Natural Development Server under BS2000 the translation tables have to be adapted as follows:

ASC2EBCDIC

Position	Value	Character
C4	BB	Ä
D6	BC	Ö
DC	BD	Ü
DF	FF	ß
E4	FB	ä
F6	4F	ö
FC	FD	ü

EBCDIC2ASC

Position	Value	Character
BB	C4	Ä
BC	D6	Ö
BD	DC	Ü
FF	DF	ß
FB	E4	ä
4F	F6	ö
FD	FC	ü

Correcting a Wrong Conversion

This example tells you what you can do if the backslash (\), which is a special character, is not converted correctly in the file *N3270_USER1.tra*.

➤ To correct a wrong conversion

- 1 Make a backup of the file *N3270_USER1.tra*.
- 2 Look up the hexadecimal values for the backslash character. The following values are used in this example:
 - The number of the code page on the PC is 850. In this code page, the backslash is defined as "5C".
 - The number of the code page on the host is 273. In this code page, the backslash is defined as "EC".
- 3 Edit the file *N3270_USER1.tra*.
- 4 Go to row 05 and column 0C in table ASC2(Ansi)EBCDICGR.
- 5 Enter the hexadecimal value for the host which is "EC".
- 6 Go to row 0E and column 0C in table EBCDICGR(Ansi)2ASC.
- 7 Enter the hexadecimal value for the PC which is "5C".
- 8 Save the file *N3270_USER1.tra*.
- 9 Import the changed file *N3270_USER1.tra*.

6

Managing Applications

■ About the Application Workspace	46
■ Connecting to an Application Server	47
■ Creating a New Application	48
■ Linking Objects and Applications	52
■ Mapping and Unmapping Applications	57
■ Connecting and Disconnecting Base Applications	59
■ Displaying/Changing the Properties of the Application Server	60
■ Displaying/Changing the Properties of an Application	61
■ Copying, Renaming and Deleting Applications	63
■ Cataloging Objects in the Application Workspace	64

A Natural application is a collection of Natural objects and non-Natural objects which build a functional unit from the business point of view. Such an application does not contain the objects themselves. It is just a collection of links to the objects or to other applications. Each link describes where the corresponding object or application is stored. Each object or application can be linked to more than one application.

All applications are managed by the Natural Development Server and are stored in the Natural Development Server file (see the Natural Development Server documentation for further information).

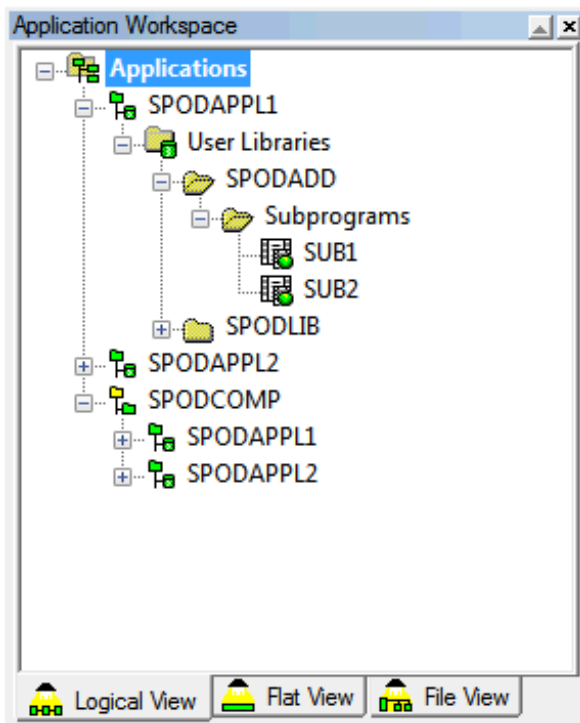


Note: This chapter describes only those commands which are not available in the library workspace; for the other commands see the documentation *Using Natural Studio*.

About the Application Workspace

Applications are created and maintained in Natural Studio's application workspace. Initially, the application workspace is not shown. For information on how to display it, see *Application Workspace* in the documentation *Using Natural Studio*.

The application workspace shows the mapped applications and their linked objects in a tree structure. It provides the same views as the library workspace (logical view, file view and flat view).



Connecting to an Application Server

When the application workspace is displayed for the first time, it does not contain any applications; only the **Applications** node is shown.

Before you can map or create your first application, you have to start a session on the development server. This session, which is called “application server session”, is used to access application data.

The information that you specify in a dialog box when connecting the application server (see below) is stored in the user profile. When you start Natural Studio the next time and expand the **Applications** node, the session is automatically started. If a password is required, the dialog box in which you have specified the connection information is shown again.

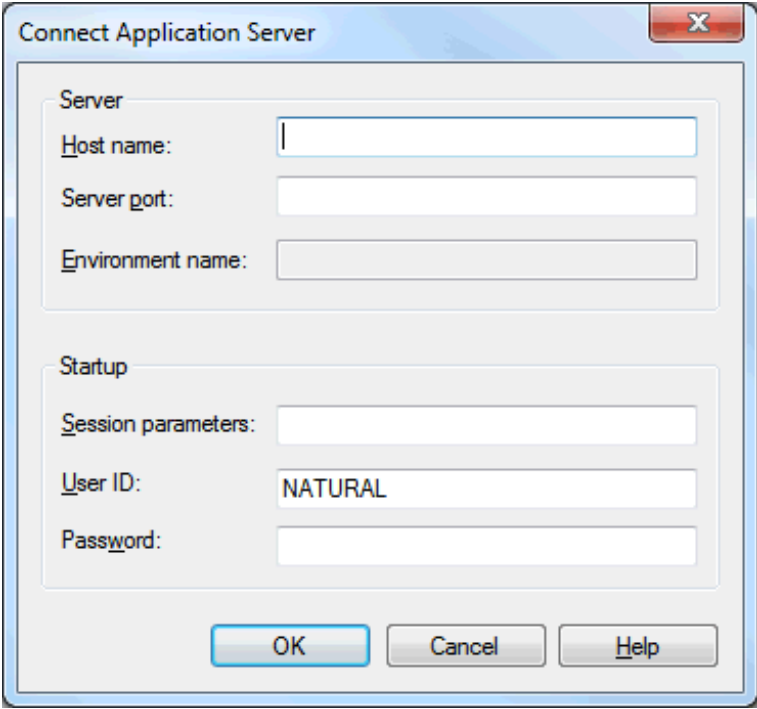
➤ To connect to an application server for the first time

- 1 In the application workspace, select the **Applications** node, invoke the context menu and choose either **New** or **Map** (these commands are described later).

Or:

Try to expand the **Applications** node.

When an application server has not yet been connected, the following dialog box appears. Your user ID is automatically provided.



The image shows a Windows-style dialog box titled "Connect Application Server". It has a standard title bar with a close button (X). The dialog is divided into two main sections. The top section, labeled "Server", contains three input fields: "Host name:" (with a cursor in the text box), "Server port:", and "Environment name:". The bottom section, labeled "Startup", contains three input fields: "Session parameters:", "User ID:" (which has the text "NATURAL" pre-filled), and "Password:". At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

- 2 Specify the following information:

Option	Description
Host name	The name of the development server on which the application data are stored.
Server port	The TCP/IP port number of the development server.
Environment name	Does not apply in this case.
Session parameters	If dynamic parameters are required for your development server, specify them in this text box. Otherwise, leave this text box blank.
User ID	If you want to connect to the development server using a different user ID, specify it in this text box.
Password	If Natural Security is installed on the development server, specify the required password in this text box. Otherwise, leave this text box blank.

- 3 Choose the **OK** button.

The application server is connected.

When you have previously chosen the **New** or **Map** command, the corresponding dialog box appears, and you have to proceed as described under [Creating a New Application](#) or [Mapping and Unmapping Applications](#).

Creating a New Application

There are two types of applications:

■ Base Application

A base application is defined as a set of Natural object links. The associated Natural objects pertain to one specific development server and are all located on the same `FUSER` system file. You can link objects of different libraries to a base application.

■ Compound Application

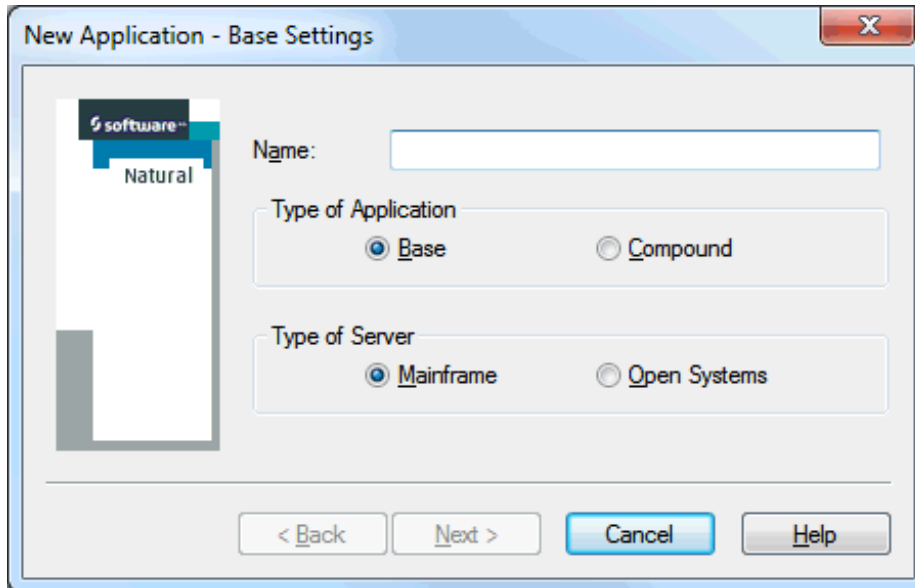
A compound application enables you to combine several base applications. It is defined as a set of links to these base applications. The base applications involved in a compound application can be spread across different `FUSER` system files or different development servers. Each base application may have a different parameter setting.

➤ To create a new application

- 1 In the application workspace, select the **Applications** node.
- 2 Invoke the context menu and choose **New**.



Note: When an application server session has not yet been established, a dialog box appears first in which you specify the development server that you want to use. See [Connecting to an Application Server](#).



With this dialog box, you can either define a base application or a compound application.

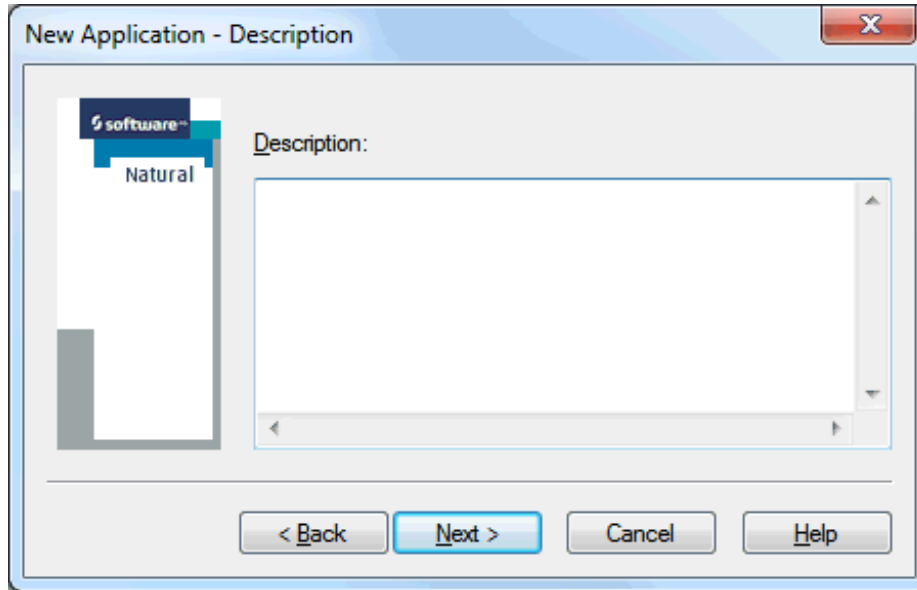
- 3 In the **Name** text box, specify a name for your application. The name must be unique among all applications and it may be up to 32 characters long.
- 4 Select the option button for type of application that you want to create: **Base** or **Compound**.

The option buttons for the type of server are only available when you have selected the option button for a base application. The type of server on which your application server session is running is automatically selected. Since it is possible to store mainframe applications on a UNIX or Windows development server, and vice versa, you can select a different type of server.

- 5 For a base application: Select the option button for the type of server:
 - **Mainframe**, or
 - **Open Systems** (for example, a UNIX or Windows server).

- 6 Choose the **Next** button.

The following dialog box appears.



Note: When you have selected a compound application, a **Finish** button is shown instead of the **Next** button which is shown for a base application.

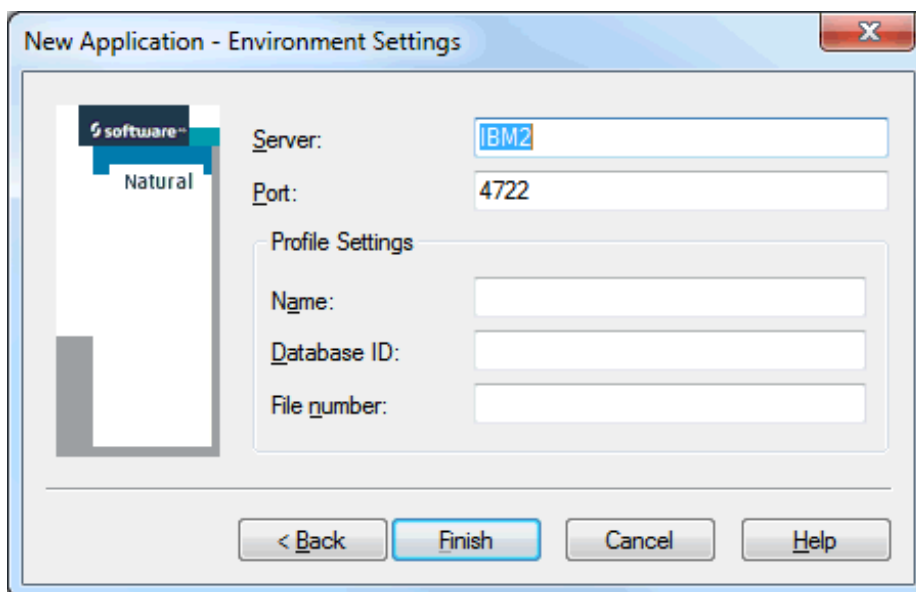
- 7 Optional. Enter a description for your application. This can be any text.
- 8 For compound application, choose the **Finish** button to create a node, with the name you have specified, in the application workspace. You can now proceed as described under [Linking Objects and Applications](#).

Or:

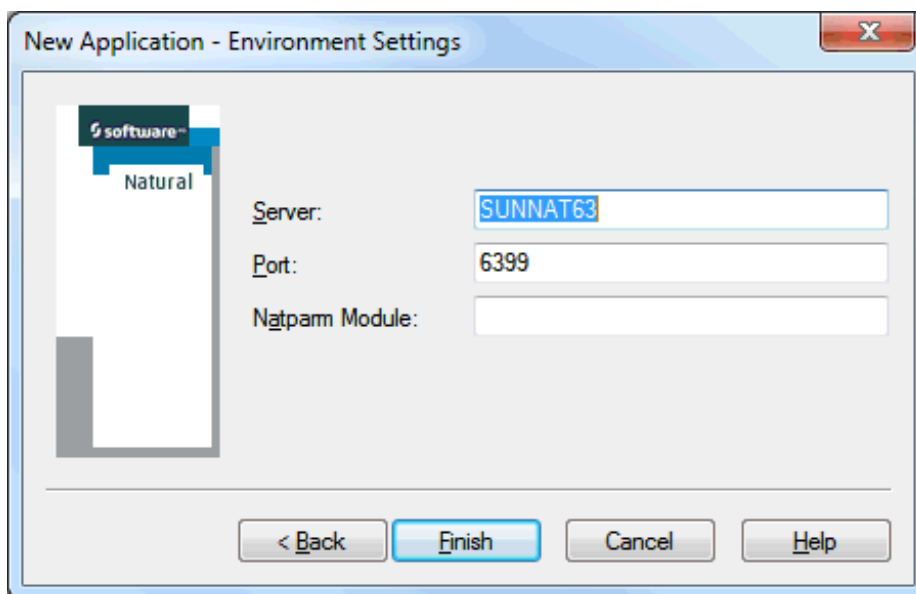
For base application, choose the **Next** button to display a dialog box (see below) in which you have to define the environment settings for the session that is to be started when the application is mapped.

Name and port number of the application server to which you are currently connected are provided as the default values for the new base application. It is also possible to use another development server.

The following dialog box appears when you have selected the server type **Mainframe**.



The following dialog box appears when you have selected the server type **Open Systems**.



The server name and port number are mandatory; the profile settings (name, database ID and file number) for server type **Mainframe** or the NATPARM module name setting for server type **Open Systems** are optional. The text boxes are pre-filled with the settings of the application server session.

- 9 Change the settings as needed.
- 10 Choose the **Finish** button to create a node for the base application, with the name you have specified, in the application workspace. You can now proceed as described under [Linking Objects and Applications](#).



Note: If a password is required, the dialog box in which you have specified the connection information is shown again. Server name, port number and session parameters cannot be changed in this dialog box. They are fixed for an application.

Linking Objects and Applications

When you have created a new application, it is empty. You must then create the links to the Natural objects that are to be part of the application.

Since an application contains just links to Natural objects, it knows only the location where a Natural object is stored. The contents of this object are not part of the application data. The Natural object is stored in the `FUSER` system file which is defined by the environment settings (server, server port and profile) of the application.

For this reason, not all of the commands which are available for a Natural object in the library workspace are available for the same object in the application workspace. It is not possible to delete, rename or move a Natural object in the application workspace. However, all commands which are needed to change the source and the executable object of a Natural object are available in the application workspace.

The following topics are covered below:

- [Linking Objects to a Base Application](#)
- [Linking Base Applications to a Compound Application](#)
- [Linking Objects Using Drag-and-Drop](#)
- [Linking Objects Using Copy-and-Paste](#)
- [Linking Objects by Adding them in the Application Workspace](#)
- [Unlinking Objects and Applications](#)

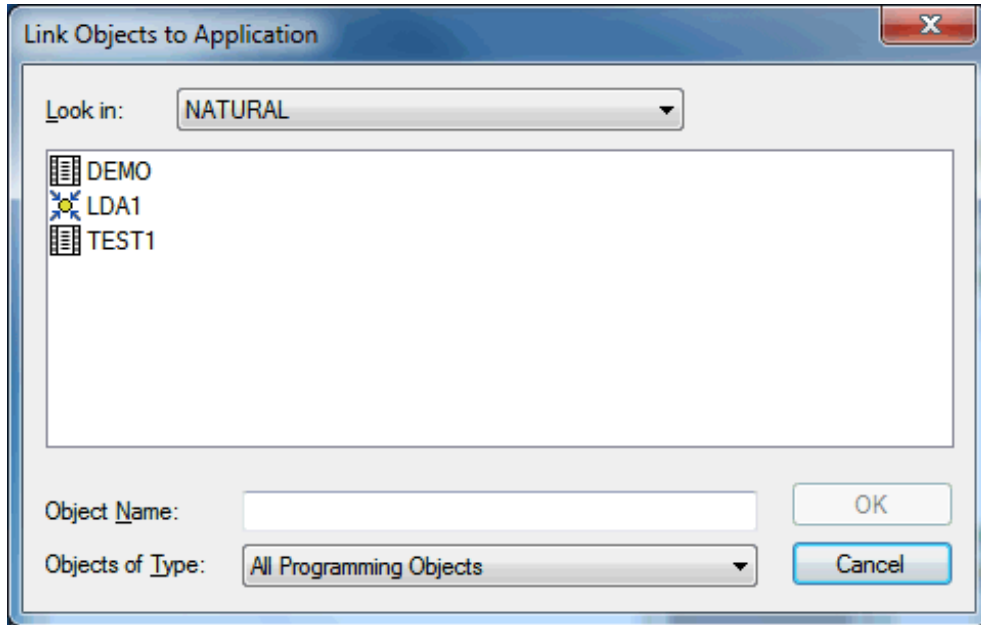
Linking Objects to a Base Application

You can link any existing Natural object of the connected application server (for example, a program or map) to your application.

➤ To link objects to a base application

- 1 In the application workspace, select the base application.
- 2 Invoke the context menu and choose **Link**.

The following dialog box appears.



- 3 From the **Look in** drop-down list box, select the library which contains the objects that you want to link to your application.

The content of the selected library is now shown in the dialog box.

- 4 Optional. If you want to restrict the list of shown objects to a specific type, select the type from the **Objects of Type** drop-down list box.



Note: When you select **DDMs** from this drop-down list box, all DDMs are shown in the dialog box. Since DDMs do not belong to a library, the **Look in** drop-down list box is not available in this case.

- 5 Select all objects that you want to link to the application.
- 6 Choose the **OK** button.

The linked objects are now available in the application workspace. The sub-nodes of an application node are the linked objects, ordered by the library where they are located. These nodes appear in a similar manner to the nodes for the Natural objects in the library workspace.

If you have linked DDMs to your application, they are available in a DDMs node.

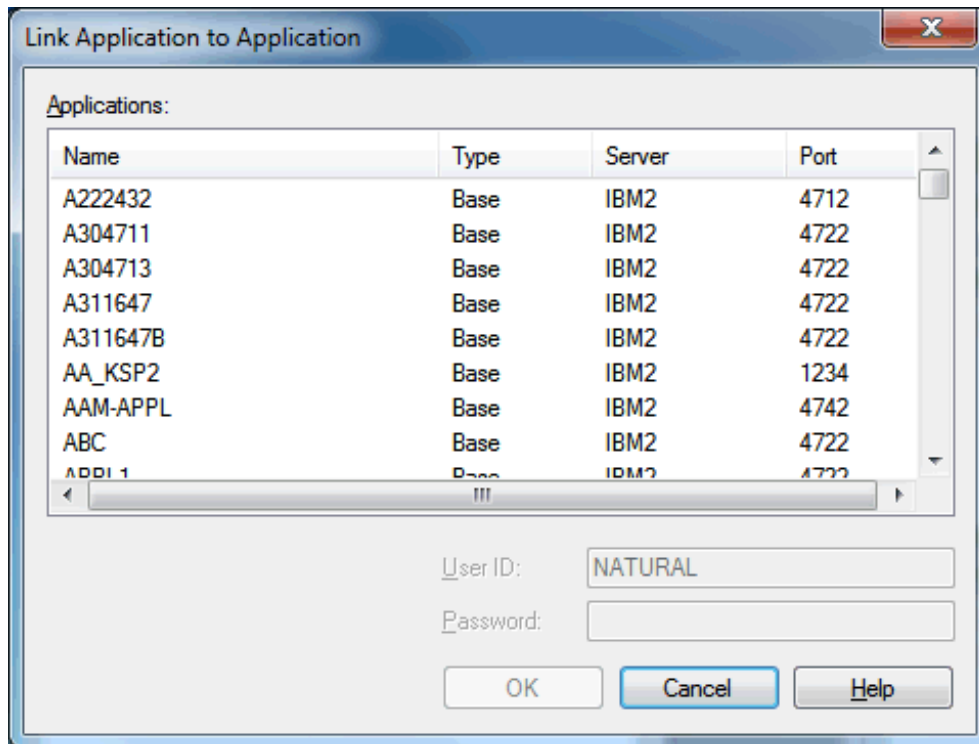
Linking Base Applications to a Compound Application

You can link any existing base application on the connected application server to a compound application. You can also link base applications which have different development server settings (that is, the session that is started for the application server may run on a different development server). A compound application thus allows you to combine objects that cannot be combined in the library workspace.

➤ To link objects to an application

- 1 In the application workspace, select the compound application.
- 2 Invoke the context menu and choose **Link**.

The following dialog box appears. It contains all available base applications (including those that you have not yet **mapped**).



- 3 Select all base applications that you want to link to the compound application.
- 4 Choose the **OK** button.

The linked base applications are now available in the node of the corresponding compound application.

Linking Objects Using Drag-and-Drop

In addition to the way objects or applications can be linked as described above, you can use the drag-and-drop feature. Objects can be linked to base applications by dragging them from the library workspace to the application workspace and dropping them on the target application. This applies to single objects, libraries or groups (for example, all the programs in a library) under the following conditions:

- The session in the library workspace must be started with the same server name and port number as the application session in the application workspace. The database IDs and file numbers of the `FUSER` of both sessions must be the same.
- Both sessions must be connected.
- Only sources can be linked to applications.

Linking Objects Using Copy-and-Paste

When the corresponding environment has been mapped in the library workspace (see the above conditions), you can copy objects, libraries or groups in the library workspace and paste links to these items into a base application.

➤ To link an object to a base application using copy-and-paste

- 1 Copy the objects, libraries or groups in the library workspace. See also *Copying Objects* in the documentation *Using Natural Studio*.
- 2 In the application workspace, select a base application.
- 3 Invoke the context menu and choose **Paste Link**.

Or:

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



Or:

Press CTRL+V.

A link for each copied item is now available in the node of the corresponding base application. It is placed into a library node which corresponds to the library node in the library workspace.

Linking Objects by Adding them in the Application Workspace

You can also link objects to an application by adding them in the application workspace. When you add, for example, a new program in the application workspace, it will be visible in both the library workspace (when the corresponding environment has been mapped; see the above conditions) and the application workspace. Remember that the object in the application workspace is just a link. Creating a new object in the application workspace is similar to the procedure used in the library workspace. See *Creating a Natural Object* in the documentation *Using Natural Studio*.

If you want to edit an object, you can do this either in the library workspace or in the application workspace. When an object is currently being edited by another user, the corresponding lock message is shown. A lock message is also shown when you try to open an object in the library workspace that you are currently editing in the application workspace (and vice versa).


Unlinking Objects and Applications

You can unlink objects from a base application and you can unlink base applications from a compound application.



Note: When you delete an object in the library workspace, the corresponding link in the application workspace is removed (you may have to refresh the display to see the change). Libraries are only known to an application when they contain objects. When you delete all objects of a library in the library workspace, this library is no longer shown in the application workspace.

➤ To unlink objects or applications

- 1 In the application workspace, select the object or application that you want to unlink.
 -  **Tip:** As in the library workspace, you can open a list view window from the application workspace. In the list view window, it is possible to select several objects at the same time (which is not possible in the tree view window). See *List View Windows* in the documentation *Using Natural Studio* for further information.
- 2 Invoke the context menu and choose **Unlink**.

Mapping and Unmapping Applications

In addition to the applications you create yourself, you can also map other applications so that they are shown in the application workspace.

A mapped application is displayed in the application workspace until you unmap this application. An application can be mapped by several users at the same time. It will only be locked momentarily when a command changes the application settings (for example, links an object).

When you map a base application, a session is started on the specified development server for working with the linked objects. The profile settings for this session are defined by the base application.

When you unmap an application, it is removed from the application workspace. Unmapping is an action that has a visual effect only. It has no influence on the changes that a user has applied to the application.

See also the descriptions of the system commands `MAP` and `UNMAP`.

> To map an existing application

- 1 From the **Tools** menu, choose **Map > Application**.

Or:

In the application workspace, select the **Applications** node, invoke the context menu and choose **Map**.

Or:

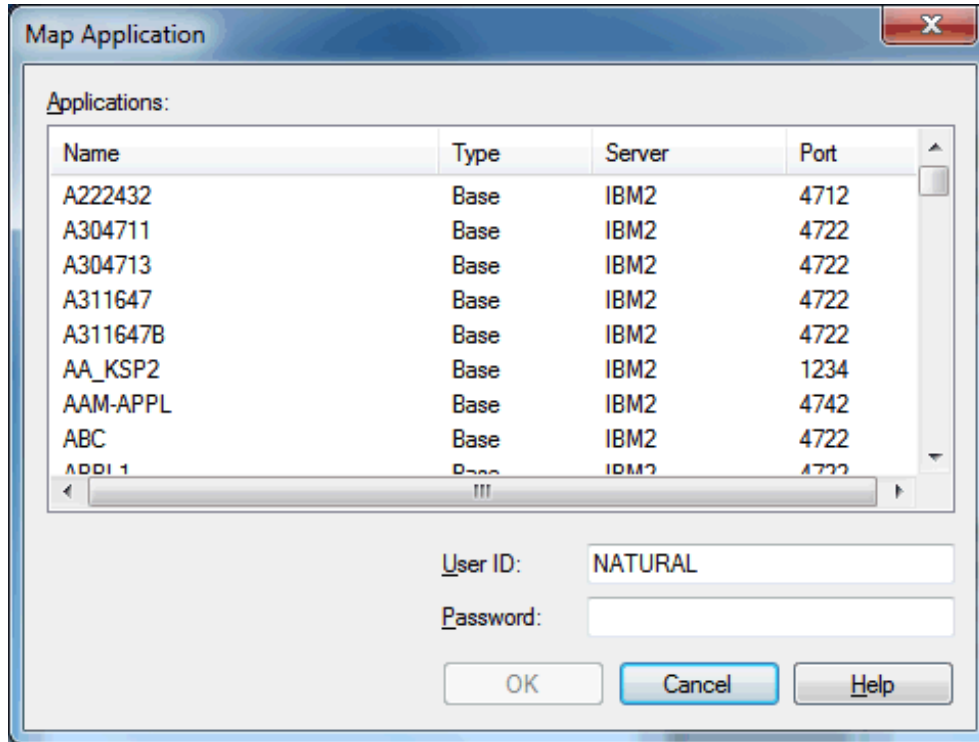
When the Tools toolbar is shown, choose the following toolbar button.



The following dialog box appears providing a list of all defined applications.



Note: When the connection to the application server session has not yet been established, a dialog box appears first in which you specify the development server that you want to use. See [Connecting to an Application Server](#).



- 2 Select all applications that you want to map.
- 3 Optional. If a password is required for the application server session, specify it in the **Password** text box. If you do not specify a required password, an additional dialog box will later prompt you for this information.
- 4 Choose the **OK** button.

The applications are now shown in the application workspace.

> To unmap an application

- 1 In the application workspace, select the application that you want to unmap.



Tip: As in the library workspace, you can also open a list view window from the application workspace. In the list view window, it is possible to select several objects at the same time (which is not possible in the tree view window).

- 2 From the **Tools** menu, choose **Unmap**.

Or:

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

Or:

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



Connecting and Disconnecting Base Applications

When you **create** a new base application or **map** an existing base application, the application is automatically mapped and connected.



Note: When you map a compound application, all linked base applications are initially disconnected.

When you terminate Natural Studio, the base applications remain mapped but are disconnected. When you restart Natural Studio, all mapped base applications are initially disconnected.

A mapped base application can thus be in one of the following states:

- **Connected**

A server session is running for this application. A connected application can be identified by its colored icon in the tree.

- **Disconnected**

No server session is currently running for this application. A disconnected application can be identified by its greyed icon in the tree.

If you want to work with the linked objects of a base application, the application must be connected.

➤ To connect a disconnected base application

- Expand the node of the base application.

Or:

Select the base application, invoke the context menu and choose **Connect**.

Or:

From the **Tools** menu, choose **Connect**.

Or:

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



If a password is missing or incorrect, a dialog box appears.

➤ To disconnect a connected base application

- Select the base application, invoke the context menu and choose **Disconnect**.

Or:

From the **Tools** menu, choose **Disconnect**.

Or:

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




Note: When you disconnect a compound application, all of its base applications are disconnected.

Displaying/Changing the Properties of the Application Server

By displaying the properties of the application server, you can obtain general information about the mapped applications.

You can also connect to a different application server as described below. However, changing the application server settings is basically not necessary, since all application data should be stored in one Natural Development Server file (see the Natural Development Server documentation for further information).

 **Important:** Keep in mind that all mappings in the application workspace are lost if you change the application server settings, because it is uncertain which Natural Development Server file will be used from the new application server.

➤ To display/change the properties of the application server

- 1 In the application workspace, select the **Applications** node.
- 2 From the **View** menu, choose **Properties**.

Or:

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

Or:

Press ALT+ENTER.

A dialog box appears displaying the properties of the application server. The following tabs are shown in the dialog box:

■ **General**

Shows the state (connected or disconnected) of the application server. If the state is "connected", the number of mapped applications, the database ID and file number of the Natural Development Server file are shown. This information cannot be changed.

■ **Application Server**

Shows the information that has been specified for connecting application server session on the development server. This information can be changed. For a description of the options in this dialog box, see [Connecting to an Application Server](#).

- 3 Choose the **OK** button to close the dialog box.

When you have changed the application server settings, a dialog box appears in which you have to confirm your changes. When you choose the **OK** button in this dialog box, the started application server session is shut down, all currently mapped applications are unmapped, and a new application server session is started.

Displaying/Changing the Properties of an Application

All changes that are applied to an application (for example, when you rename it or link objects), are stored directly in the Natural Development Server file (see the Natural Development Server documentation for further information).



Caution: If you change the environment settings of an application, keep in mind that changing the **FUSER** and **FDIC** settings can have the result that the Natural objects linked to the application are no longer found. Moreover, the **FDIC** settings should never be changed in the application workspace, because this implies that a different application data pool is used.

➤ To display/change the properties of the application server

- 1 In the application workspace, select a base application or compound application.
- 2 From the **View** menu, choose **Properties**.

Or:

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

Or:

Press ALT+ENTER.

A dialog box appears displaying the properties for the selected application. The following tabs may be shown in the dialog box:

- **General**

Shows the name of the application, the type of application (for a base application, the platform is also shown), the date and time when the application was created and last modified, and the ID of the user who created/changed the application. For a base application, the state (connected or disconnected) is also shown. This information cannot be changed.

- **Description**

Shows the description of the application. This information can be changed.

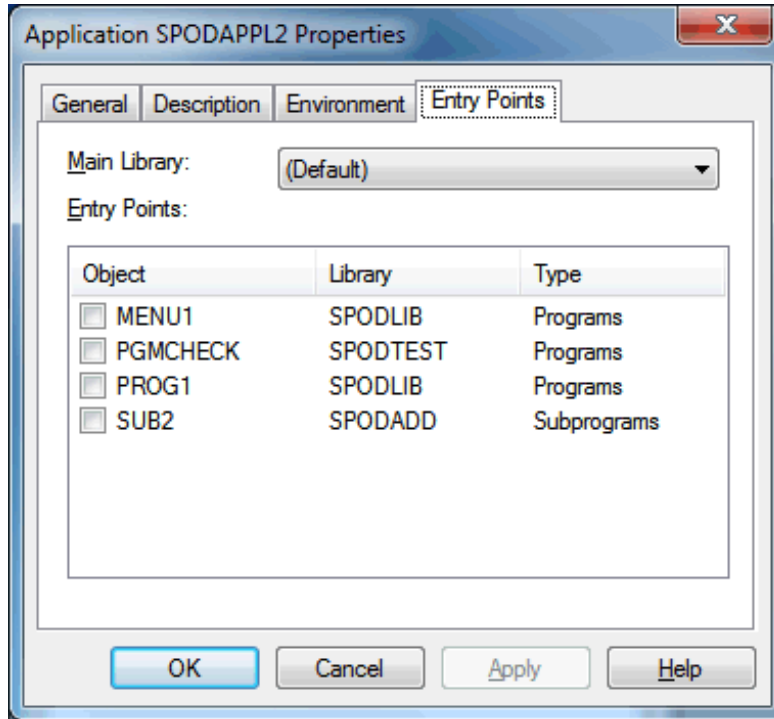
- **Environment**

Only shown for base applications. Shows the startup parameters for the application server session, which is started for the selected application. These are the same environment settings that are specified when [creating a new application](#).

The server name and port number can be changed as long as no objects have been linked to the application. All other settings can be changed at any time. These are the user ID and the profile settings (name, database ID and file number) for server type **Mainframe** or the NATPARM module name setting for server type **Open Systems**.

- **Entry Points**

Only shown for base applications (including those which have been linked to compound applications), and only when the base application is connected. The information on this tab is only used by XRef Evaluation.



You can use this tab for descriptive purposes. XRef Evaluation will use the information specified here for cross-reference analysis of an application.

The **Main Library** drop-down list box provides for selection of the libraries which contain the currently linked objects. You can select the logon library from which your application is to be started.

The **Entry Points** list box contains all objects that are linked to the application and that are possible entry points. You can select one or more objects to become entry points for your application.

- 3 Choose the **OK** button to close the dialog box.

When you have changed any settings, they are saved.

Copying, Renaming and Deleting Applications

You can copy, rename or delete each application in the application workspace.

You rename or delete an application in the same way as you rename or delete a node in the library workspace. See *Renaming Objects* and *Deleting Objects* in the documentation *Using Natural Studio*.

When you delete an application, it is removed physically. It is not possible to redisplay it in the application workspace. The objects linked to the application, however, are not deleted; they are still available in the library workspace.

For copying applications, the application workspace provides a special command; see below.

➤ To copy an application

- 1 In the application workspace, select a base application or compound application.
- 2 Invoke the context menu and choose **Create Copy**.

A copy of the selected application is placed in the application workspace. This may take a while. The copied application has the old name plus the addition "-1". This number is incremented by one for each new copy of the original application.

Cataloging Objects in the Application Workspace

The following exception applies in the application workspace when cataloging the objects in a library: only the objects that have been linked with the application are cataloged (i.e. only the objects that are shown in the application workspace). The objects that are only shown in the library workspace are ignored. See also *Cataloging the Objects in a Library* in the documentation *Using Natural Studio*.

7

Using the Application Wizard

The Application Wizard is a tool which can be used as a simple alternative to the application workspace. You can use it to create, copy, rename and delete applications, to modify the description of an application, and to link/unlink Natural objects to/from an application.

➤ To start the Application Wizard

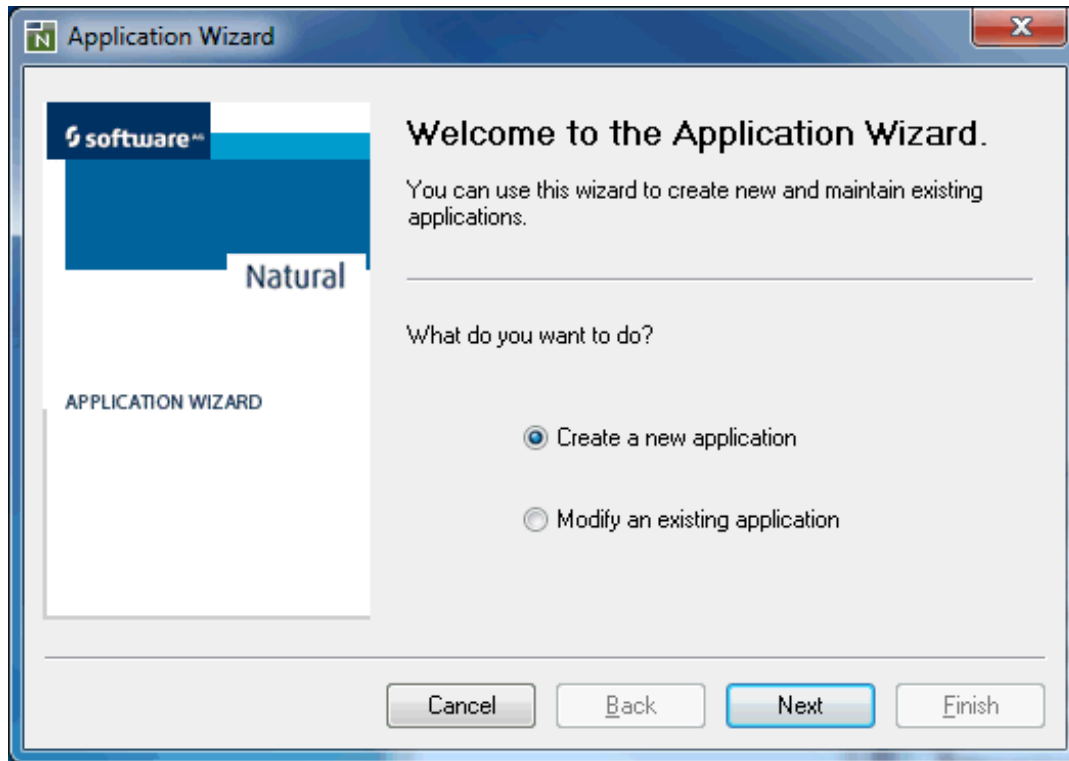
- 1 From the **Tools** menu, choose **Application Wizard**.



Note: This command is only available when the Application Wizard has been activated in the Plug-in Manager. See *Plug-in Manager* in the documentation *Using Natural Studio* for further information.

When an application server has not yet been connected, a dialog box appears first in which you have to specify the required connection information. See [Connecting to an Application Server](#) for detailed information on this dialog box.

When an application server has been connected, the Application Wizard is started.



- 2 Select the required option button, choose the **Next** button and follow the instructions on the following pages of the Application Wizard.