



NATURAL

Natural
Remote Development
Version 5.1.1 for Windows



This document applies to Natural Version 5.1.1 for Windows and to all subsequent releases. Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

© June 2002, Software AG
All rights reserved

Software AG and/or all Software AG products are either trademarks or registered trademarks of Software AG. Other products and company names mentioned herein may be the trademarks of their respective owners.

Table of Contents

Remote Development - Overview	1
Remote Development - Overview	1
Application Workspace	2
Application Workspace	2
Using the Application Workspace	2
Mapping an Application Server	3
Creating a New Application	3
Mapping/Unmapping an Application	4
Properties of Applications	5
Displaying Application Server Properties	5
Changing Application Server Properties	5
Creating New Objects for an Application	6
Linking/Unlinking a Natural Object to/from an Application	6
Changing the Settings of an Application	7
CATALL for Applications	8
Using the Library Workspace	9
Using the Library Workspace	9
Map/Unmap Environment	9
Delete Environment	10
Show Properties of Environment	10
Object Locking	11
Object Locking	11
Object Locking Concept	11
Move, Delete and Rename	11
Save and Stow	12
Locking of Local Objects and Remote Objects	12
Local Objects	12
Remote Objects	12
Different Locking Behavior with Remote and Local Objects	12
Mainframe Server Restrictions	12
Natural XRef GUI Client - Overview	14
Natural XRef GUI Client - Overview	14
Introduction to XRef GUI Client	15
Introduction to XRef GUI Client	15
About XRef Data	15
What is XRef GUI Client?	15
Installation of XRef GUI Client	16
Installation of XRef GUI Client	16
Invoking XRef GUI Client	17
Invoking XRef GUI Client	17
Plug-In Manager Settings	17
Invoking XRef GUI Client From the Main Menu	17
Invoking XRef GUI Client From the Context Menu	19
Cross References	20
Active Cross References	20
Passive Cross References	21
List Children	22
Influence of Environmental and User-specific Settings	23
Terminal Emulation	24
Terminal Emulation	24
Purpose of the Terminal Emulation	24
Working in the Terminal Emulation Screen	24
Transferring Data Using the Clipboard	24

Modifying the Font	26
Modifying the Color Scheme	26
Printing the Contents of the Terminal Emulation Screen	27
Terminal Emulation Keys	27
Using Non-Latin Character Sets	28

Remote Development - Overview

This documentation covers those topics which apply only when you are working with Natural Studio in a remote development environment using Natural Single Point of Development for remote development on a mainframe computer.

The following topics are covered:

- **Application Workspace** Explains 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.
- **Library Workspace** Explains 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 to prevent concurrent updating of Natural objects.
- **Natural XRef GUI Client** Covers the following topics: introduction to XRef GUI Client, installation of XRef GUI Client and invoking XRef GUI Client. A tutorial describing the use of XRef GUI Client is included on the Natural documentation CD.
- **Terminal Emulation** Explains how to use the terminal emulation window that appears in a remote development environment to display non-GUI information.

For more information on Natural Single Point of Development (SPoD), refer to the SPoD-specific documentation on the current Natural documentation CD and in ServLine24 (Introducing Natural Single Point of Development, Tutorial, Administration Policies and Procedures, Natural Development Server, etc.)

Application Workspace

The application workspace is an area on the Natural Studio screen where all mapped applications and their objects are shown in a view which complements the existing logical, flat and file views. In this area, a tree structure comprising all objects linked to an application is displayed.

Applications are managed by the Natural Development Server and stored in the development server file.

The following topics are covered:

- Using the Application Workspace
- Mapping an Application Server
- Creating a New Application
- Mapping/Unmapping an Application
- Properties of Applications
- Creating New Objects for an Application
- Linking/Unlinking a Natural Object to an Application
- CATAL for Applications

Related Topics:

- Application Concept
- First Steps with Natural Single Point of Development (SPoD)

(These topics are described in the Natural SPoD documentation on the Natural documentation CD.)

Using the Application Workspace

Initially, when the new Natural version is started for the first time, the application workspace will not be shown.

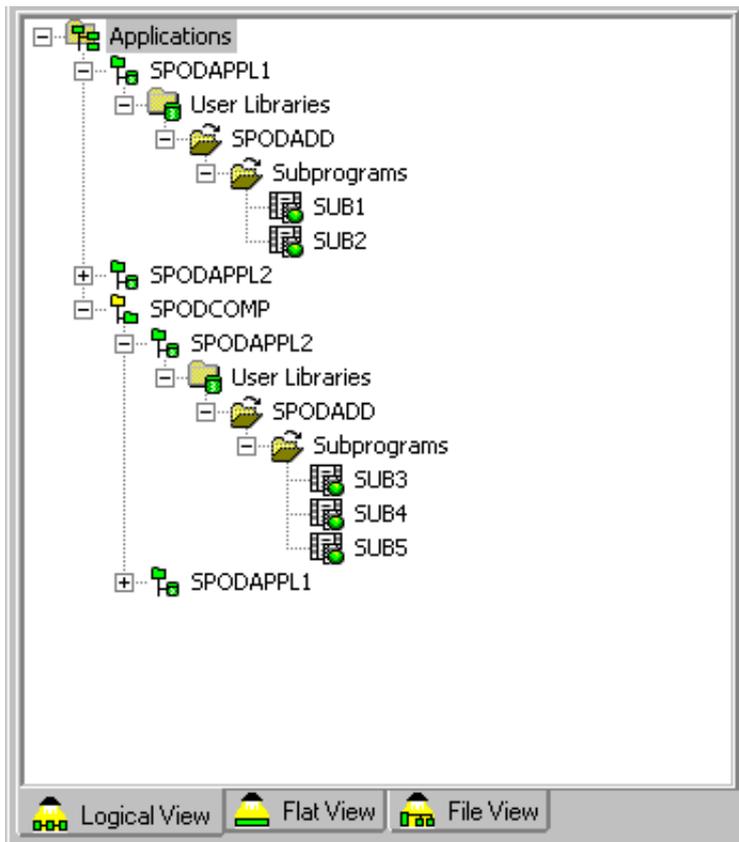
To toggle the application workspace display

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

When the application workspace is displayed for the first time it does not contain any applications; only the "Applications" node is displayed. You can either map existing applications or create new applications which are automatically mapped, using the context menu of the "Applications" node.

When Natural is started the next time all applications which were mapped in the previous Natural session will be automatically re-mapped.

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 as Natural objects in the library workspace.



The application workspace consists of a "Logical View", "Flat View" and "File View" like the library workspace.

This document describes only those commands which are not available in the library workspace; for the other commands see the library workspace documentation.

Mapping an Application Server

Before the first application is mapped or created, a development server session has to be started, which is used to connect the Application Manager which handles all application data. The "Map Application Server" dialog is displayed to collect the information for starting this development server session.

The settings for this session will be stored in the user profile (.PRU). When Natural is started the next time, the session will be started when the "Applications" node is expanded. However, the "Map" dialog will be displayed again if a password is required.

The Application server environment settings can be changed with the "Properties" dialog of the "Applications" node, see Changing Application Server Properties.

Creating a New Application

▶ To create a new application

1. In the application workspace window, select the "Applications" node.
2. Click the right mouse button and, from the resulting context menu, choose "New..".
The "New Application - Base Settings" dialog is displayed.
3. Enter the name of the application (must be unique over all applications) and choose the type of the

application ("Base" or "Compound").

The maximum length of the application name is 32 characters.

4. Choose "Next".

The "New Application - Description" dialog is displayed.

5. Enter a description for the application.

6. **For a compound application:** Choose "Finish".

For a base application: Choose "Next".

The "New Application - Environment Settings" dialog is displayed.

Enter the development server session settings for the application.

The server name and port number will be filled with the definitions from the Application Server (see Managing an Application Server).

These are the settings of the session which is started when the application is mapped.

Choose "Finish" if you want to create the application with the specified settings.

The server name and port number are mandatory.

Note: In every dialog, you can choose "Back" to go back to the previous dialog and change the settings.

Mapping/Unmapping an Application

From the pool of all available applications, you can view the applications you are working with in Natural Studio. You have to map every application you want to display in your environment.

A mapped application will be 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 (e.g. links an object).

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

When you map a compound application, all sub-applications of this application will be mapped as well. A sub-application cannot be unmapped as long as the compound application is mapped.

When you unmap an application, this application will be removed from the application workspace. Unmapping is an action that has a visual effect only. The changes a user has applied to the application will not be affected when you unmap the application.

To map an existing application

1. In the application workspace window, select the "Applications" node.
2. Click the right mouse button and, from the resulting context menu, choose "Map ..".
The "Map Application" dialog is displayed. It lists all available applications with the following information: application name, application type, development server name, development server port number.
3. Select the application you want to map.
4. Enter a password in the "Password" field if the development server session for the application requires a password.
5. Choose "OK" to map the application.

If the password is missing or incorrect (only for base applications), the "Map Application <applicationname>" dialog will prompt for it.

If the application session cannot be started successfully, e.g. because the specified profile does not exist, an error message will be displayed and the application node will not contain a "+" for expanding it.

The properties of such an application have to be changed so that a session on the server can be started.

The application is added in the application workspace window.

 **To unmap an application**

1. In the application workspace window, select the application to be unmapped.
2. Click the right mouse button and, from the resulting context menu, choose "Unmap".

The selected application is removed from the application workspace.

Properties of Applications

- Displaying Application Server Properties
- Changing Application Server Properties

Displaying Application Server Properties

 **To obtain general information about the mapped applications**

1. In the application workspace window, select the "Applications" node.
2. Click the right mouse button and, from the resulting context menu, choose "Properties".
The "Mapped Applications Properties" dialog is displayed.

The dialog has two tabs:

- **General**

The "General" tab shows the number of mapped applications, number of mapped base applications and number of mapped compound applications. This information cannot be changed.

- **Application Server**

The "Application Server" tab shows the name, port number and session parameters und User Id of the development server session, which is used to connect the Application Manager on the Natural development server. This information can be changed (see Changing the Application Server Properties).

Changing Application Server Properties

Changing the application server settings is basically not necessary, because all application data should be stored in one Natural Development Server File.



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

 **To change the properties of the application server**

1. In the application workspace window, select the "Applications" node.
2. Click the right mouse button and, from the resulting context menu, choose "Properties".
The "Mapped Applications Properties" dialog is displayed.
3. Click the "Application Server" tab, enter the new application server settings and choose "Apply" or "OK" to save your settings.
A message box appears asking you if you really want to change the application server settings.
4. Choose "OK" to confirm your changes.

The application server settings are changed, the started development server session is shut-down, all currently mapped applications are unmapped, and a new development server session is started.

Creating New Objects for an Application

You can either use a library that already exists in the attached FUSER system file or create a new library in the FUSER for an application. An existing library is unknown to the application as long as this library has no objects linked to it.

The action of creating a new object is similar to the procedure used for the library workspace, see also Introduction to Natural Studio and Natural Studio.

The new object is automatically linked to the selected application.

Note: Unlinking such an object from the application will remove not the Natural object itself but only the link to the application.

Linking/Unlinking a Natural Object to/from an Application

An application contains links to Natural objects. This means, that the application "knows" only the location where a Natural object is stored, but 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.

To link a Natural Object to an Application

1. In the application workspace window, select the application you want to link to the Natural object.
2. Click the right mouse button and, from the resulting context menu, choose "Link...".

For a base application: The "Link Objects to Application" dialog is displayed.

To link a Natural object from a library (e.g. a program)

1. Select the library in the "Look in" drop-down list box.
2. In the "Objects of Type" drop-down list box,
 - either select "All Programming Objects" if you want to list all programming objects from the library
 - or select a specific type if you want only to list objects of this type.
3. Select all objects you want to link.
Or enter their names in the "Object Names" edit field.
4. Choose "OK" to link the selected objects.

To link a Natural object which does not belong to a library (e.g. a DDM)

1. Select the type in the "Objects of Type" drop-down list box (e.g. "DDMs").
The "Look in" drop-down list box is disabled and all objects of the specified type are listed.
2. Select all objects you want to link.
Or enter their names in the "Object Names" edit field.
3. Choose "OK" to link the selected objects.

For a base application: The linked objects appear in the application workspace.

For a compound application: The "Link Application to Application" dialog is displayed. The dialog lists only base applications.

To link an application to a compound application

1. Select the application you want to link to the compound application.
2. Enter a password in the "Password" field
(if the development server session for the application requires a password).
3. Choose "OK" to link the application.

If the password is missing or incorrect, the "Map Application <applicationname>" dialog asks for it.

The linked application appears in the application workspace.

To unlink a Natural Object from an Application

1. In the application workspace window, select the node of the object you want to unlink from the application.
2. Click the right mouse button and, from the resulting context menu, choose "Unlink".

The selected object (either a Natural object or a sub-application) is removed from the application workspace.

Changing the Settings of an Application

All changes which are applied to an application (e.g. change name or link an object), will be stored directly in the Natural development server file.



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 change the settings of an application

1. In the application workspace window, select the node of the application you want to change.
2. Click the right mouse button and, from the resulting context menu, choose "Properties".
The "Application <applicationname> Properties" dialog is displayed.

This dialog has three tabs:

- **General**

The "General" tab shows the following information of an application: name, Database Id and file number of the FUSER system file where the linked objects are located (only for base applications), type of application, date when application was last modified, date when application was created. The information cannot be changed.

- **Description**

The "Description" tab shows the description of the application. The description can be modified.

- **Environment (only for base applications)**

The "Environment" tab shows the start-up parameters for the development server session, which is started for the selected application. The following information is displayed: the server name and port number, the profile name, database id and file number. All settings can be changed.

CATALL for Applications

If a CATALL command is issued for a library in the application workspace, only those Natural objects which are linked to the selected application are cataloged.

Using the Library Workspace

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

This document covers the following topics:

- Map/Unmap Environment
 - Delete Environment
 - Show Properties of Environment
-

Map/Unmap Environment

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

If you want to connect to a development server for the very first time, you have to map it as described below. Once you have connected to a development server, a node for this development server session is automatically shown in the tree the next time you invoke Natural Studio. The state of each development server is set to "unmapped".

Note:

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 click the corresponding node in your library workspace.

▶ To map to a development server

1. From the Tools menu, choose Map Environment.
Or click the following toolbar button



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

2. In the "Name" text box, enter the name of the development server on the mainframe.
3. In the "Port" text box, enter the TCP/IP port number of the development server.
4. If dynamic parameters are required for your development server, specify them in the "Session Parameters" text box. Otherwise, leave this text box blank.
5. If Natural Security is installed on the development server, specify the required password in the "Password" text box. Otherwise, leave this text box blank.
6. 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. The command line now shows the name of the library that is currently selected in the tree and the name of the active environment.

▶ To connect to a previously mapped development server

The development server you have previously mapped is now shown with the state "Unmapped".

1. Click the plus sign next to the node name
or double-click the node name or select the node and choose Map Environment from the Tools menu.
The "Map Environment" dialog box appears. It shows the information that you have previously provided for this development server.
2. If Natural Security is installed on the development server, specify the required password in the "Password"

text box. Otherwise, leave this text box blank.

3. Choose the OK button.

Delete Environment

It is also possible to delete an unmapped development server so that its name is no longer shown in the tree.

1. Select the server node and click the right mouse button.
2. Choose Delete from the resulting context menu.

Show Properties of Environment



To obtain general information about the mapped environment

1. Select the server node and click the right mouse button.
2. Choose Properties from the resulting context menu.

The "General" tab shows Server, Port, User Id and session parameters.

Object Locking

In a Natural Single Point of Development 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.

This document describes the locking concept, the objects affected, the different behavior with remote and local objects and the restrictions that apply in conjunction with a mainframe server.

The following topics are covered:

- Object Locking Concept
 - Locking of Local Objects and Remote Objects
 - Mainframe Server Restrictions
-

Object Locking Concept

When being edited in one of Natural Studio's editors or tools, the Natural objects listed below are locked for local or remote access.

- Program
- Subprogram
- Subroutine
- Copycode
- Help Routine
- 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 (like e.g. the Class Builder when local classes are changed), 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.

Move, Delete and Rename

When you are issuing a Move, Delete or Rename command on a Natural source object, the locking state of the object concerned is checked:

- If the source object is locked, the command execution is rejected.
- If the source object to be moved, deleted or renamed is not locked, the command is executed.

The execution of such a command does not cause the object to be locked.

Save and Stow

You can execute a `SAVE` or `STOW` command only on a source object that is not locked. If the object has been locked by another user, the command execution is rejected.

The following scenario may cause inconsistent data:

1. A source object is being edited in Natural session A. The object is locked.
2. The same object is displayed in Natural session B using the `READ` command (by the same or another user). In this case, the original state of the source object is read. The object cannot be edited because it is locked.
3. In Natural session A, the modified object is stored using the `SAVE` command and the editor is closed. The editor unlocks the object.
4. In Natural session B, the object read is loaded into the editor with the `EDIT` command. It is modified or it is not touched, but a `SAVE` or `STOW` command is executed on it. This causes the modifications made in session A to be overwritten.

Locking of Local Objects and Remote Objects

Local Objects

Locking of sources in the local environment does not require the existence of a Single Point of Development environment. It is effected using Windows operating system functions. There is no difference with respect to the locking mechanism available with Natural for Windows Version 4.1.

Remote Objects

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

Different Locking Behavior with Remote and Local Objects

In the local environment, it is possible to execute a `SAVE` or `STOW` command on a locked object only in that Natural session in which the object was locked.

In the remote environment, it is possible for a user who has locked an object in a Natural session to execute a `SAVE` or a `STOW` command on this object from within another session. However, this user will not be able to edit this object in a Natural Studio editor.

If, after session termination, Natural Studio is unable to unlock objects in the event of an error, then the locks on objects in the local environment will normally be released by the operating system. This does not apply to locked objects in the remote environment. These objects can be unlocked using the `UNLOCK` command.

Mainframe Server Restrictions

The locking of source objects of a Development Server is supported only in a Single Point of Development 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 used in a SPoD environment exclusively in that environment. Using the FUSER system file concurrently by another stand-alone Natural session may give rise to inconsistencies. This applies also to a Natural session which works with an FNAT system file with installed Remote Development Server but is not operated from within Natural Studio.

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

Natural XRef GUI Client - Overview

This documentation applies to the following topics related to XRef GUI Client:

XRef GUI Client Documentation

-  Introduction to XRef GUI Client
-  Installation of XRef GUI Client
-  Invoking XRef GUI Client

Introduction to XRef GUI Client

This section covers the following topics:

- About XRef Data
 - What is XRef GUI Client?
-

About XRef Data

In addition to documentation objects, XRef data are stored in the development server file. XRef data document objects of an application that have already been implemented. XRef data are created at compilation (catalog) time and contain name references or fully qualified references. Name references will be evaluated during cross reference analysis to reflect the logical structure in the current runtime assignment. Data of this type mirror the 'real world' of the implementation, providing a comprehensive summary of information for all implemented members of an application.

XRef data can be used to retrieve information on individual members about:

- its invocation structure
- the resources it uses

For further information on the generation of XRef data see [Activating the Generation of Cross-Reference Data](#) in section [First Steps with Natural Single Point of Development](#).

What is XRef GUI Client?

The XRef GUI Client plug-in is used to navigate through cross reference information created during CAT or STOW commands in a development server file. The information is displayed in a tree view within the Natural Studio. Two types of references - active and passive (i.e. referencing and referenced) can be displayed. Navigating through the hierarchies of active and passive references is possible within separate tree views.

Installation of XRef GUI Client

The XRef GUI Client installation is part of the Natural for Windows Installation. You can either install it during a first-time installation or install it later in a separate setup process.

For further information on the installation procedure see Installing the XRef GUI Client in the **Natural for Windows Installation** documentation.

Invoking XRef GUI Client

This section covers the following topics:

- Plug-In Manager Settings
 - Invoking XRef GUI Client From the Main Menu
 - Invoking XRef GUI Client From the Context Menu
 - Cross References
 - Active Cross References
 - Passive Cross References
 - List Children
 - Influence of Environmental and User-specific Settings
-

Plug-In Manager Settings

Before any action can be taken, the XRef GUI Client plug-in has to be activated in your Plug-In Manager. Detailed information on the activation procedure can be found in section Plug-In Manager.

Once it is activated, you can invoke XRef GUI Client either from the main menu, toolbar buttons or by right clicking on an object within the tree view where XRef data is stored.

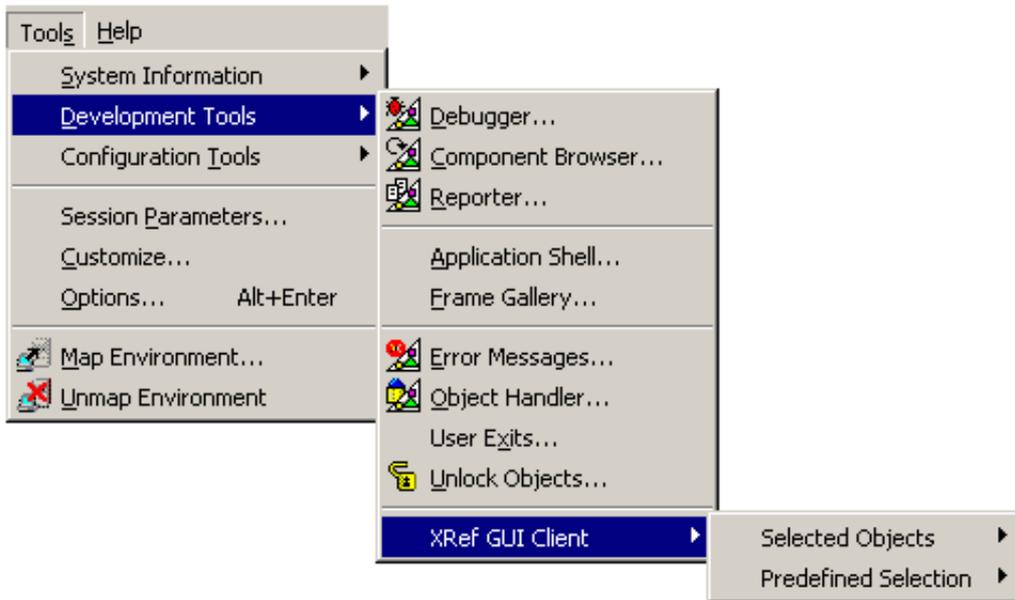
Note:

The tree view on XRef data that is presented by XRef GUI Client always depends on the LOGON library and Steplib settings that are active during initial invocation/instanciation of the corresponding cross reference tree view. For that reason, the results of the same XRef data retrieval can vary due to different environmental or user-specific settings. See Influence of Environmental and User-specific Settings for further information.

Invoking XRef GUI Client From the Main Menu

To invoke XRef GUI Client from the main menu:

Choose a library or an object within a library by left clicking on the library or object. From the main menu, choose Tools > Development Tools > XRef GUI Client. See the sample menu below.



You can choose from two options now. If you choose Selected Objects, only the active or passive cross references of the object you selected are displayed.

Note:

This option is only available, if you have selected an object within a library (for example a program, subprogram or subroutine).

Alternatively to invoking XRef GUI Client from the main menu, you can also use one of the following buttons:

-  for Active Cross References
-  for Passive Cross References.

However this option is also only available if you have selected an object within a library.

If you choose Predefined Selection, you can select a specific type of object to have its cross references displayed. The data displayed will then be referred to as "Predefined Selection Objects of Type X", where "X" represents one of the objects from the menu below:



The root object in the resulting tree view is also called "Predefined Selection Objects of Type X", where "X" represents the chosen object. The source objects, for example programs and their cross references (either active or passive) are represented on the next level in the tree view (one level below the root object "Predefined Selection").

Note:

This option is only available if you are logged on a library.

Invoking XRef GUI Client From the Context Menu

Select an object within a library by left clicking on it. Right click on your selection to display the context menu. Select either Active Cross References or Passive Cross References. See the sample menu.

**Note:**

It is not possible to define a Predefined Selection (i.e. displaying the cross references of a specific object type, such as program or class) from the context menu.

Cross References

There are two kinds of cross references which you can display separately in the XRef GUI Client: active and passive cross references.

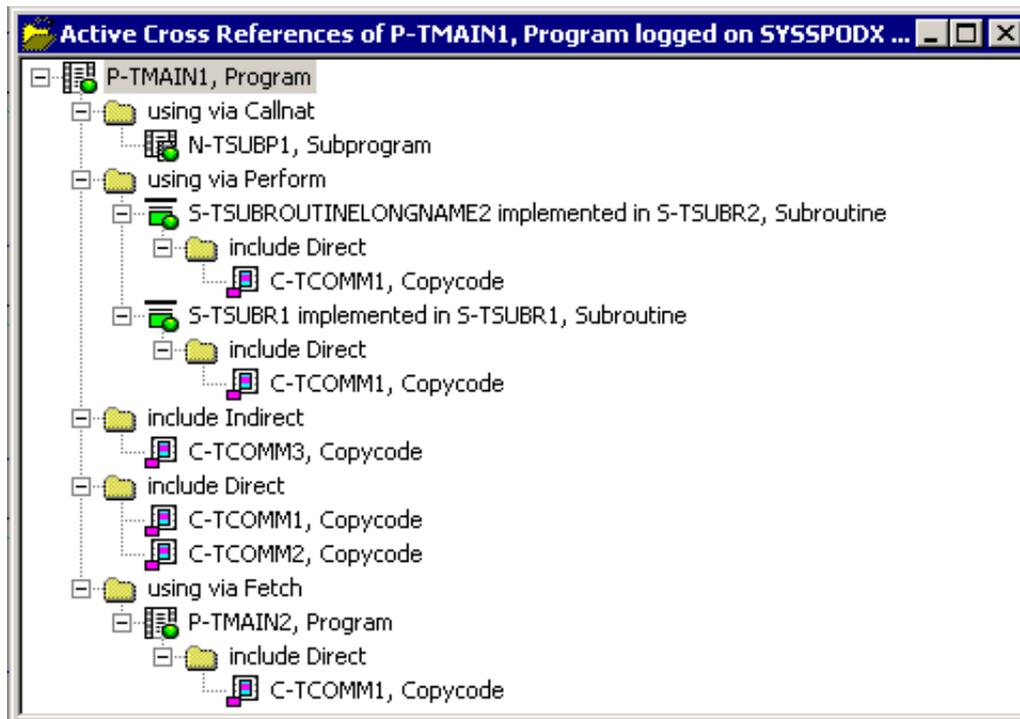
Active Cross References

If you select active cross references of an object, XRef GUI Client generates a tree view with the referencing ("using") object (for example a program) on top of the tree. Below this object, all objects which are referenced ("used") by the top level object are shown (for example other programs, subprograms or subroutines). For a more detailed explanation of the term object, see Contents of XRef Data.

Active cross references will provide you with an answer to the question: "Which parts of my application depend on the object I have currently selected?"

Example

In the following sample tree view the result of a request for active cross references is displayed. The referencing object P-TMAIN1 (a program) stands on top of the tree view. This is the object for which the request was performed:

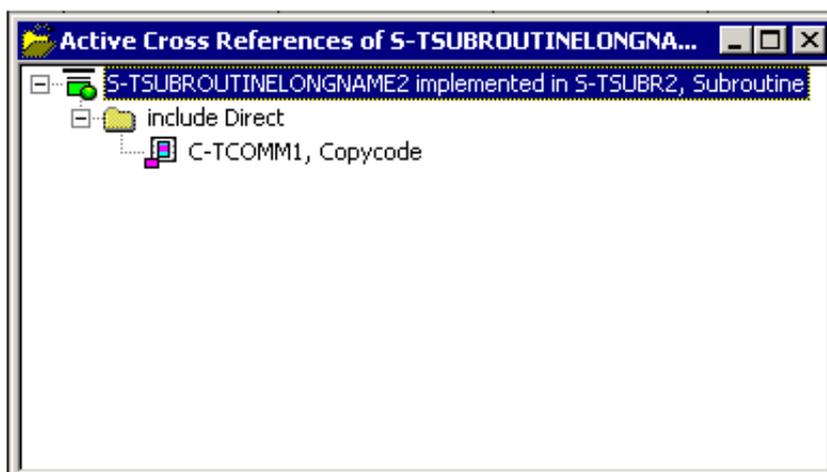


Note:

Subroutines with long names are also supported by the XREF GUI Client. Subroutines with long names are represented as follows:

<long name> implemented in <member name>, Subroutine

See sample tree view below.



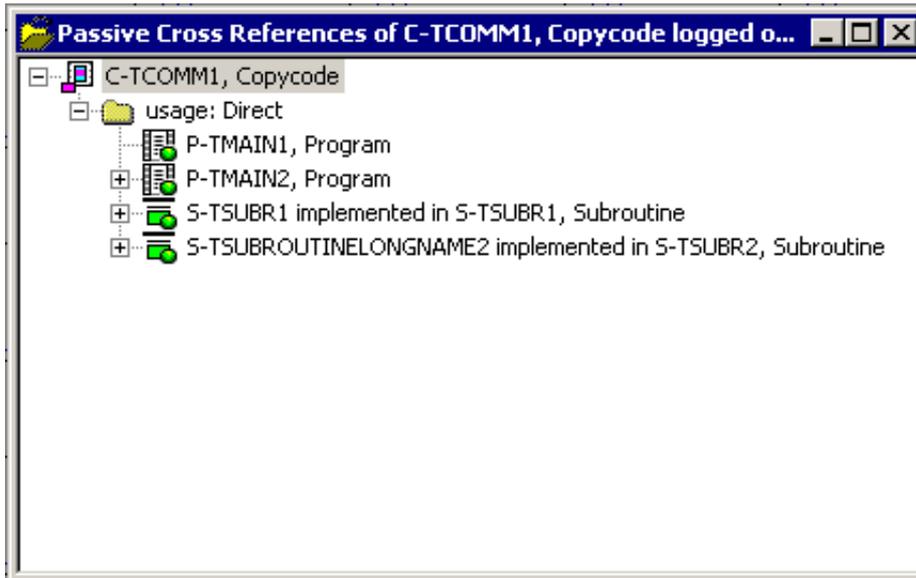
Passive Cross References

If you select passive cross references of an object, the XRef GUI Client generates a tree view with the referenced ("used") object on top of the tree. Below this object, all objects which reference ("use") the top level object are shown. For an explanation of the term object, see Contents of XRef Data.

Select passive cross references if you want to know which other objects use a certain object. For example, if you have a copycode you might want to know in which parts of your application it is included. There are some types of objects which by default only have passive cross references, such as copycodes, DDMs and methods.

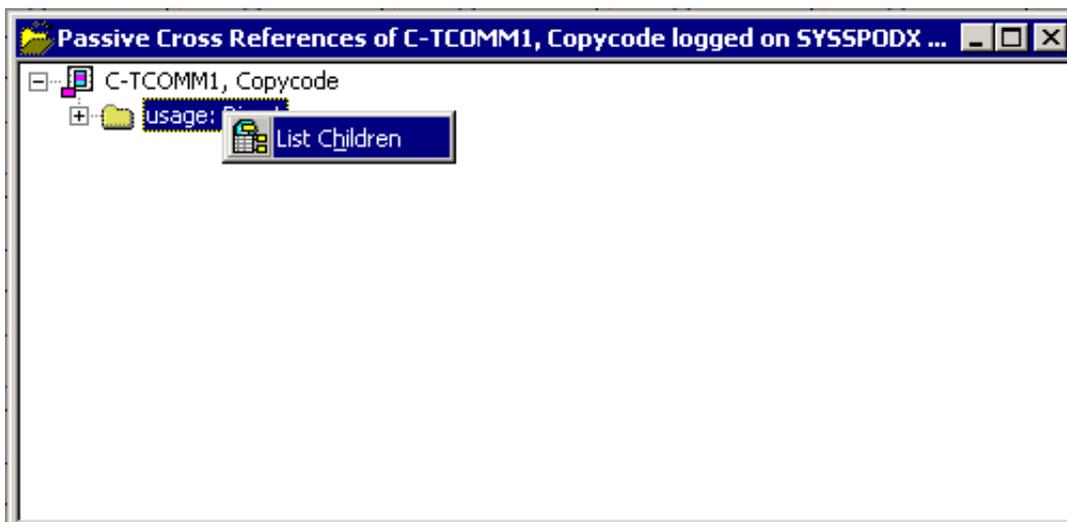
Example

In the following sample tree view, the result of a request for passive cross references is displayed. The referenced object C-TCOMM1 (a copycode) stands on top of the tree view. This is the object for which the request was performed:

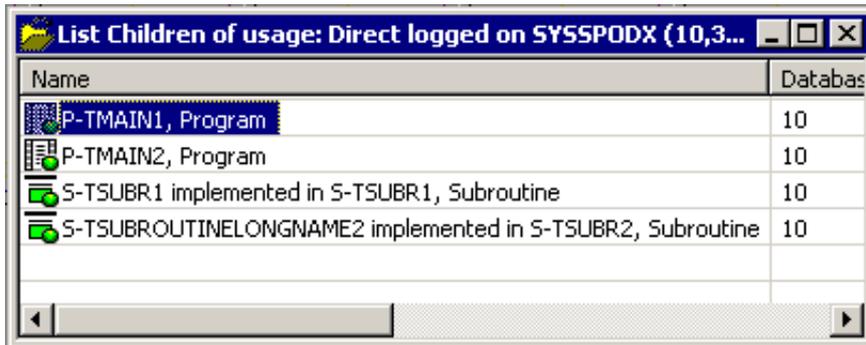


List Children

The option List Children is only available within the tree view on XRef data. This tree view is the result of your request processed by the XRef GUI Client. See the sample below.



The result of the List Children option is a list view of the children of the selected object (for example a program or subprogram) or node (for example *using via Perform* or *include Direct*). See the sample below.



Name	Databas
P-TMAIN1, Program	10
P-TMAIN2, Program	10
S-TSUBR1 implemented in S-TSUBR1, Subroutine	10
S-TSUBROUTINELONGNAME2 implemented in S-TSUBR2, Subroutine	10

In contrast to the tree view, you can select more than just one object within the list view and perform an action such as Catalog or Stow. This is very helpful when performing mass updates is necessary.

Influence of Environmental and User-specific Settings

The tree view on XRef data that is presented by XRef GUI Client always depends on the LOGON library and Steplib settings that are active during initial invocation/instanciation of the corresponding cross-reference tree view.

For example, if you switch to a different library while XRef GUI Client processes a request, the presented result will always refer to the library and Steplib settings that were active when you started your request. This may (but not necessarily has to) lead to different results if you start the same request on the same object, but with different LOGON libraries and/or Steplib settings.

Terminal Emulation

The following topics are covered:

- Purpose of the Terminal Emulation
- Working in the Terminal Emulation Screen
- Transferring Data Using the Clipboard
- Modifying the Font
- Modifying the Color Scheme
- Printing the Contents of the Terminal Emulation Screen
- Terminal Emulation Keys
- Using Non-Latin Character Sets

See also SPoD-Specific Limitations and Considerations in the Natural Single Point of Development documentation.

Purpose of the Terminal Emulation

The terminal emulation enables you to test a complete application inclusive of its character output, without having to leave Natural Studio.

A screen output of an executed program will automatically cause 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 Close button.

Natural for Mainframes utilities for which Natural Studio does not offer a graphical user interface are made available by the terminal emulation. Entering the corresponding system command in Natural Studio's command line will automatically open the terminal emulation window.

Working in the 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 at any position in the terminal emulation screen. Otherwise, you can use your keyboard to enter data as normal.

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. 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

1. Move the mouse pointer to the beginning of the text you want to select.
2. Press and hold down the left mouse button.
3. Drag the mouse until all desired text is selected.
4. Release the mouse button.

This clears any previous selection. You can now cut, copy or append the selected text.

▶ 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 clear the selection

- Click another screen position.
The existing selection is canceled.

▶ To delete the selection

- From the **Edit** menu, choose **Clear**.
Or press DEL.
The selection is deleted (except protected fields). It is not transferred to the clipboard.

▶ To copy the selection and append it after existing text in the clipboard

- From the **Edit** menu, choose **Append Copy**.
The selection is copied to the clipboard and appended after already existing text.

▶ To cut the selection and transfer it to the clipboard

- From the **Edit** menu, choose **Cut**.
Or press CTRL+X.
Or click the following toolbar button:



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.
Or click the following toolbar button:



▶ 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.
Or click the following toolbar button:



The content of the clipboard is inserted at cursor position.

Modifying the Font

You can define another font to be used for the current session directly from the terminal application. Your new font definition is stored for the next session.

▶ To modify the font

- From the **Session** menu, choose **Font**.
The "Fonts" dialog box appears:

The "Fonts" dialog box lets you define the font that is to be used for terminal emulation.

Font	The list box shows all installed monospaced fixed fonts and TrueType (scalable) fonts.
Size	The sizes available for the selected font are displayed in this list box. For a TrueType font, one or more entries named "TrueType" can be shown. Select one of these entries to display the corresponding style in the "Sample" field.
Adjust the font size to the window size	This feature 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.

Modifying the Color Scheme

You can define another color scheme to be used for the current session directly from the terminal application. Your modifications are stored for the next session.

▶ To activate another color scheme

1. From the **Session** menu, choose **Color**.
The "Color Selection" dialog box appears.
The active color scheme is indicated by an arrow.
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 directly from the terminal application. To do so, use one of the following command buttons in the "Color Selection" dialog box:

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.
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.

Printing the Contents of the 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

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.

Terminal Emulation Keys

Keys such as PF1 to PF24, PA1 to PA3, etc. that start a terminal function can be invoked either by clicking the corresponding symbol or by using a combination of keys on the keyboard, for example:

▶ To choose the CLR (clear) key using the PC keyboard

- Press CTRL+F1.

▶ To choose the PF1 key using the PC keyboard

- Press F1.

▶ To choose the PF13 key using the PC keyboard

- Press SHIFT+F1.

Terminal Keyboard	PC Keyboard		
	SHIFT	CTRL	(Single) 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
PF1-12			F1-F12
PF13-24	X		F1-F12
RESET		X	F11

Using Non-Latin Character Sets

By default, the terminal emulation uses a latin character set. You can define a different character set (e.g. Cyrillic) to be used for the terminal emulation.

To use a different character set

1. From the menu bar, choose **Tools > Options** and select the "Workspace" tab.
2. From the "Translation Table" selection box, select the desired character set.
3. Choose the **OK** button.

Your new definition is immediately active and is stored for the next session. In addition, you may have to modify the font of the terminal emulation accordingly, see Modifying the Font.