natural

Single Point of Development

Limitations and Considerations

software AG

This document applies to Natural Version 4.1.2 for Mainframes, Version 6.1.1 for UNIX, Version 6.1.1 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.

Table of Contents

SPoD-Specific Limitations and Considerations

When you are working with Natural Single Point of Development, you will encounter a few limitations which are due to the different capabilities of the graphical user interface available on the local site and the character-based user interface that exists on the remote site. Also, some restrictions exist which will be eliminated in one of the next releases. In addition, this document includes hints which are important for the efficient use of the remote development facilities.

The following topics are covered:

- Limitations
- Performance Considerations
- Accessing Work Files

Limitations

- Execution of Programs Calling CICS-Related 3GL Programs
- LE370 3GL Programs
- Execution of Programs Accessing DL/I Databases
- Execution of Programs Accessing VSAM Databases
- Execution of Programs Accessing DB2 Databases
- PC Down/Uploads Using Natural Connection
- System Commands
- Moving/Copying Error Messages
- LIST DDM, EDIT DDM
- Classes in Tree View
- Maps Containing GUI Elements
- Field Sensitive Maps
- Resources
- Dialogs
- Natural ISPF Macros and Recordings
- SYSLIB/SYSLIBS
- Allow Lower Case Input in Program Editor of Natural Studio
- Session Parameter NC
- Terminal Emulation
- Dependencies between XRef Evaluation and Predict
- Remote Debugging
- Working without Natural Security

Execution of Programs Calling CICS-Related 3GL Programs

The execution of programs calling 3GL programs which in turn use CICS-specific information or issue CICS-specific calls (CICS EXEC ...) is not possible with this version of Natural Single Point of Development, but is planned for a future version.

LE370 3GL Programs

The execution of LE370 3GL programs is not supported by the Natural Development Server under SMARTS on VSE/ESA.

Execution of Programs Accessing DL/I Databases

The execution of programs accessing DL/I databases is not possible with this version of Natural Single Point of Development, but is planned for a future version.

Execution of Programs Accessing VSAM Databases

If Com-plete is not installed under SMARTS, access of VSAM databases is not supported by the Natural Development Server under SMARTS on VSE/ESA.

Execution of Programs Accessing DB2 Databases

In the case of an access to DB2 from a program executed on the Natural Development Server, the user ID for the database access is not the client's user ID, but the job or task name of the development server's started task.

Access of DB2 databases is not supported by the Natural Development Server under SMARTS on VSE/ESA.

PC Down/Uploads Using Natural Connection

The execution of programs which use Natural Connection to perform a PC down/upload is not possible with this version of Natural Single Point of Development. It is planned for a future version.

System Commands

- System Command DELETE, RENAME
- System Command SYSDDM
- System Commands Unavailable for Remote Development
- System Commands Entered Directly on the Development Server

System Commands DELETE, RENAME

These system commands are not available with this version, but are planned for a future version. Instead of DELETE, you can use the system commands UNCAT, PURGE and SCRATCH and the Delete dialog from the context menu. Instead of RENAME, you can use the rename dialog from the context menu.

System Command SYSDDM

The system command SYSDDM is not available, since the DDMs are listed in the tree view under the node DDM, and because all functions of the utility SYSDDM are available by using the context menu or the menu bar.

System Commands Unavailable for Remote Development

The following system commands are not available, since their use would make no sense with a graphical user interface:

- EDT
- HELLO
- MAINMENU

System Commands Entered Directly on the Development Server

All system commands which are entered not in the user interface of Natural Studio but directly on the server, via the stack by using the Natural session parameter STACK or in a program by using the STACK TOP COMMAND or by entry in the terminal emulation window, are executed directly by the Development Server without control of Natural Studio. As a result, the character-based representation of the corresponding command appears in the terminal emulation window. For example, when you specify the session parameter STACK=(LOGON XYZ;L * *) when mapping a server, the result will be that the system command LIST is output in the terminal emulation window.

It is even possible to invoke the mainframe editors. However, this may lead to inconsistencies (see also Object Locking in the Natural for Windows documentation). Therefore, you are strongly recommended to use only the GUI editors.

For the next version of of Natural Single Point of Development, it is planned that all system commands for which a graphical presentation exists in Natural Studio will be passed by the Development Server to Natural Studio for execution in itsgraphical user interface.

The commands HELLO and MAINMENU do not cause a screen output on the development server side, since this would not make any sense in the SPoD environment. The menu-driven user interface is that of Natural Studio.

Moving/Copying Error Messages

Moving and copying of error messages is different in remote and local environments:

- When error messages are moved or copied within the remote environment or are moved or copied from the local to the remote environment or vice versa:
  the error messages involved are merged, that is,
  - error messages which already exist in the target environment are replaced,
  - messages which do not exist in the source library are kept in the target library,
  - messages which do not exist in the target library are added.
- When error messages are moved or copied within the local environment, the messages involved are handled on file level, that is,
  - all error messages (that is, files) of a language are deleted and
  - the file from the source library is created anew in the target library.

LIST DDM, EDIT DDM

In contrast with a pure Natural mainframe environment, that is, without remote development from Natural Studio, the command EDIT DDM is available also from a user library. This means that it is not necessary to expand the DDM node in the tree view to be able to edit a specific DDM. However, when Natural Security is used, the use of the commands LIST DDM and EDIT DDM can be restricted only via the security profile of the mainframe Natural utility SYSDDM.

Classes in Tree View

Although classes are shown in the tree view, the command OPEN is available only in the file view display. Performing an OPEN command on a class in the file view will invoke the program editor. Similarly, new classes can be created only in the file view, that is, by first creating an object of type Program which then can be saved as an object of type Class.

With one of the next versions of Natural Single Point of Development, it will be possible to create/edit the classes in all types of views using the Class Browser.

Maps Containing GUI Elements

Maps containing GUI elements can be moved or copied from the local environment to a remote environment. However, the GUI elements are not displayed when the map is being tested or executed on the remote environment.

Field Sensitive Maps

For these maps the consistency check for a map field is made as soon as the data field is filled out by the user. Field Sensitive Maps can be moved or copied from the local environment to a remote environment. However, a field sensitive map can not be tested or executed on a remote mainframe environment.

Resources

Since the object type Resource is unknown in a mainframe environment, it is not possible to copy or move objects of this type from the local environment to a remote environment. Combined operations, for example, simultaneous moving of resources and programs are not possible.

Dialogs

Dialogs can be stored on the mainframe. Therefore it is possible to move or copy dialogs from the local environment to a remote environment. Private Resource Files of a dialog will not be moved or copied together with the dialog. It is also possible to list dialogs in a remote environment. The creation of new dialogs or editing dialogs is not possible in a remote environment.

Natural ISPF Macros and Recordings

As the object types Natural ISPF Macro and Recording available with Natural for Mainframes cannot be processed by Natural Studio, they will not be displayed in the tree view of the library work space. If a library consists only of such object types, the library will be displayed nevertheless in the tree view, but without any subnodes.

If a library containing such object types is deleted, then the objects of these two specific object types will not be deleted and the library will continue to be displayed in the tree view.

Objects of the types Natural ISPF Macro and Natural ISPF Recording cannot be linked to an application.

SYSLIB/SYSLIBS

The restricted libraries SYSLIB/SYSLIBS of the server are not shown in Natural Studio’s tree view, because a logon to these libraries is not possible. These libraries can be modified only by using a Natural utility such as INPL, NATUNLD/NATLOAD or SYSMAIN.

Allow Lower Case Input in Program Editor of Natural Studio

The program Editor of Natural Studio is case-sensitive, that is, lower case input will be included in the program source in lower case. The compiler on the Development Server, however, expects upper case code in its normal setting. This issue can be fixed by setting the compiler option LOWSRCE=ON. But this setting will have specific side effects which should be noticed. Refer to the CMPO profile parameter in the Natural Reference documentation.

Session Parameter NC

In a Natural Development Server environment on mainframe computers, the value OFF will be assumed for the Natural Development Server, even if NC=ON has been specified.

With this version, it is not possible to modify the NC parameter setting for the Development Server using the Globals dialog.

Terminal Emulation

The terminal emulation supports 3270 Model 2 screens. The support of 3270 Model 3, 4 and 5 screens is planned for one of the next versions of Natural Single Point of Development.

Dependencies between XRef Evaluation and Predict

If you are using dynamic language assigned when calling other objects such as ’INPUT USING MAP ’MAP1&’, the connection between caller and called object cannot be retrieved by using XRef Evaluation.

Natural on the mainframe supports case-sensitive calls to other objects such as ’PERFORM SUBROUTINE’. With the current version of SPoD, this may lead to strange results when, in XRef Evaluation, trees are expanded and it is not possible to request case-sensitive calls with the filter dialog.

Remote Debugging

When the remote debugging facility was implemented, the goal was not to provide any new functions, but to support the existing essential debugging functions under the Natural Development Server. These functions are:

- Stepmode
- Breakpoints
- Watchpoints
- Display and modification of variables and their contents during a break

Generally, it was intended to provide for compatibility between the debug functionality that exists in a Natural on mainframes and a Natural on PC environment. Hence, the current state of development constitutes the lowest possible common denominator. Especially, the debug statistics as supported on mainframe are not yet supported in a remote debug environment.

Which Differences Exist in Debugging on Mainframe and on PC?

The following is an overview of differences that exist between Natural debugging in a mainframe (MF) environment and debugging in a PC environment (PC). The remote debugging between PC and mainframe supports the functionality as pointed out in the cases below for mainframe (MF):

- **Restarting a Debug Process**

  MF The restart function is not supported.

  PC Debug on PC offers a special restart function which is not available for remote debugging on mainframe.

- **System Variables**

  MF System variables can be displayed, but not modified. It is not possible to set watchpoints for system variables.

  PC System variables can be modified. It is possible to set watchpoints for system variables.

- **Display of Binary Variables**

  MF Either alphanumeric or hexadecimal display of binary variables can be selected. In alphanumeric display, binary variables with length ranging from 1 to 4 are interpreted and displayed as numerical values. Binary variables with length > 4 are displayed in alphanumeric representation.

  PC Binary variables are always represented as hexadecimal values.

- **Watchpoint for Array Elements**

  MF Watchpoints are supported for single array elements.

  PC It is possible to define a watchpoint for an entire array or a selected range.

- **Several Breaks (BP/WP) per Line Number of the Program**

  MF Multiple breaks or interrupts may arise for one and the same line number (because of multiple definitions of breakpoints or watchpoints).

  PC Stepmode, BP and WP settings together result in a maximum of one break per line number.

- **Breakpoints**

  MF Breakpoints can be defined for programs which are found in the current library or in any steplib.

  PC Debugging allows to define breakpoints for programs in any library (not necessarily current library or steplib).

- **BP-END**

  MF BP-END becomes active before the program is left. Normally, this is triggered by the END statement. However, if the END statement in the program code is preceded, e.g. by a FETCH statement, the BP-END condition will become effective already there.

  PC BP-END always refers to the END statement.

- **Different Handling of Watchpoints with Comparison Operators**

  MF The watchpoint becomes active when the watchpoint variable has changed and when the comparison condition is met.

  PC The watchpoint becomes active when the comparison condition for the watchpoint variable is met.

- **Leaving the Debugger**

  MF When you leave the Natural Debugger on mainframe, the program execution continues.

  PC Leaving the Debugger causes the program execution to be stopped.

- **Debugging of Programs which are Called through the Stack**

  MF Stacked programs can be debugged, when any breakpoint or watchpoint has been defined, but they cannot be entered automatically in stepmode..

  PC Programs can be entered automatically in stepmode.

Working without Natural Security

Although Natural Security is not a prerequisit to access development servers using a SPoD architecture, it is recommended that you use Natural Security in such an environment. Because, without Natural Security, it is not possible to check a user identification that is used in the "Map" dialog. As a consequence, the user would be able to map to each Natural environment with an invalid user ID.

Performance Considerations

The working situation displayed in the library workspace of Natural Studio is based on the representation of the **entire** user system files. The tree view window opens when the user connects to the Natural Development Server. For this, the entire system file has to be analyzed and the corresponding information has to be transferred from the Natural Development Server to the Natural Studio Client. In the case of very large system files, the build-up of the tree view window can be very time consuming. A status information displayed in the status bar keeps the user informed about the progress of the screen build-up operation. This is to avoid the impression that the connection to the Natural Development Server might be interrupted.

Tip:	Switch on the status bar using the View - Status Bar function of the Menu bar.
	Make sure that the transfer rate of your network is 10 Mbit/s at minimum.

In the default configuration of Natural Studio, all operations which result in a modification of the system file, for example, moving or copying objects, but also a SAVE or STOW command, will cause the tree view window contents to be refreshed, which can be very time consuming in the case of very large system files.

Tip:	Disable the automatic refresh function by choosing **Tools** > **Options** > **Workspace** and deactivating the function "Perform automatic refresh".
	The automatic refresh may then be activated if actually needed, for example, on a specific library node or for all user libraries, by activating the refresh function in the context menu.

Since the tree view of the application workspace displays only the objects that are linked to the application, the build-up of its tree view screen is consequently considerably faster, which is another advantage of using the application workspace.

Library Statistical Record

In a Natural Single Point of Development environment, either local Natural libraries are accessed or Natural Studio requests the library statistical data from the remote development server. In the local environment the data are stored persistently in the FILEDIR structure of the library. In the case of a Mainframe development server, Natural objects are stored in system files in the database and the requested statistical data of a library is not stored permanently up to Natural Development Server Version 1.1.3. Therefore for each client request, the data must be evaluated again. This causes a lot of Adabas calls on the development server and has a significant performance impact for large libraries.

In order to reduce the number of Adabas calls and to improve the performance, a statistical record has been introduced with Natural Development Server Version 1.1.4.

Concept

In a Natural Single Point of Development environment, for every library of the FUSER or FNAT system file, a library statistical record is created and maintained. The statistical record resides on the same system file where the library resides and contains for every library:

- Total number of objects
- Total number of all sources
- Total number of all cataloged objects
- Total number of objects for every object type
- Accumulated size of all sources
- Accumulated size of all cataloged objects
- Accumulated size of sources for every object type
- Accumulated size of all cataloged objects for every object type

Supported object types:

- Program
- Map
- GDA
- LDA
- PDA
- Subroutine
- Helproutine
- Subprogram
- Copycode (source only)
- Text (source only)
- Command Processor
- Dialog (source only)
- Class
- Error Message (source only)

When Natural Studio requests the statistics for a library the first time, the library statistical record is created and saved in the appropriate system file. Once the library statistical record has been built, all requests from Natural Studio will be satisfied by reading and sending the contents of the statistical record instead of rebuilding the complete library statistics.

When the user initiates an explicit refresh for a library, the statistical record is rebuilt completely.

Data Consistency

The library statistical record of a Mainframe development server is supported only in a Single Point of Development environment. The statistical record is always up to date if all system file modifications are initiated in this environment.

All commands or operations triggered by Natural Studio which will modify the system files, such as add new object or copy, move, delete or rename object will update the library statistical record on the development server.

In addition, the library statistical record is regenerated if an object list for the whole library is requested or the statistical record for a given object type is updated if an object list for this type is requested.

To ensure data consistency of the library statistical records of your FNAT and FUSER system files, you are strongly recommended to make changes on the same FNAT and FUSER system file used in a Single Point of Development environment exclusively in that environment.

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 library statistical data might become inconsistent. The same is true if you start a server application that directly changes the FNAT or FUSER system file.

Such inconsistencies may be resolved after the next regeneration (implicit rebuild via get object list or explicit refresh) of the library statistical record is forced.

Restrictions

Statistical records cannot be used for read-only system files. In this case, the old behavior is used.

Accessing Work Files

This topic is discussed in the Natural Operations for Mainframes documentation. Refer to Natural as a Server under OS/390, Print and Work File Handling with External Datasets in a Server Environment.

