

Natural Development Server

Natural Development Server for UNIX

Version 8.4.1 for UNIX

October 2018

This document applies to Natural Development Server Version 8.4.1 for UNIX 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 © 2001-2018 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: NDV-NNDVDOC-841-20180720

Table of Contents

Preface	v
1 Introducing Natural Development Server	1
Purpose of Natural Development Server	2
Remote Development Functions	2
2 Development Server File	5
Purpose of the Development Server File	6
Relations between FDIC and the Development Server File	6
Unique Development Server File	6
3 Installing Natural Development Server under UNIX	7
4 Configuring Natural Development Server under UNIX	9
User-Specific Environments	10
NDV User Exit Routine	10
5 Operating Natural Development Server	13
Introduction	14
Starting the Development Server	14
Determining the Screen Size for Web I/O	15
Terminating the Development Server	16
Getting Help	16
NDV Configuration File	17
Trace	17
Error Messages	18
6 SPoD-Specific Limitations and Considerations	21
Limitations	22
Performance Considerations	24
Natural Documentation and Online Help	25

Preface

This documentation applies to Natural Development Server (product code NDV) for UNIX.

Natural Development Server for UNIX is released together with Natural for UNIX. It has the same version number as Natural for UNIX.

For information on changes, enhancements or new features in this version of Natural Development Server, see the *Release Notes* in the corresponding Natural for UNIX documentation.

Introducing Natural Development Server	Describes purpose and functionality of Natural Development Server which is used in conjunction with NaturalONE or Natural for Windows (as a client) in a Natural Single Point of Development (SPoD) environment.
Development Server File	Describes purpose and use of the Natural Development Server file, a central dictionary file that is structurally identical to the Natural system file FDIC.
Installing Natural Development Server	How to install Natural Development Server on UNIX.
Configuring Natural Development Server	How to configure Natural Development Server.
Operating Natural Development Server	How to operate Natural Development Server.
SPoD-Specific Limitations and Considerations	Describes the 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. In addition, this document includes hints which are important for the efficient use of the remote development facilities.

Related Documentation

- Natural Single Point of Development documentation
- NaturalONE documentation
- Natural for Windows documentation
- Natural for UNIX documentation

1 Introducing Natural Development Server

- Purpose of Natural Development Server 2
- Remote Development Functions 2

This chapter describes the purpose and the functions of Natural Development Server (product code NDV) which are used in conjunction with NaturalONE or Natural for Windows (as client) in a Natural Single Point of Development (SPoD) environment.

Purpose of Natural Development Server

Natural Development Server enables you to use NaturalONE or the Natural Studio development environment provided by Natural for Windows to develop and test Natural applications in a remote Natural UNIX environment running under the operating system UNIX.

For more information on NaturalONE and remote development, see

- [NaturalONE documentation](#) (describes the SPoD client side; how to manage offloaded Natural objects in the Eclipse workspace, and also how to modify them directly on a development server).
- [Natural Single Point of Development documentation](#) (general information).

For more information on Natural Studio and remote development, see

- [Natural for Windows documentation](#) or [Help system](#) (describes the SPoD client side; how to manage Natural objects directly on a development server).
- [Natural Single Point of Development documentation](#) (general information).

Remote Development Functions

- [Establishing a Connection between Client and Server](#)
- [Using the Remote Development Functionality](#)

Establishing a Connection between Client and Server

A connection to an active development server can be established by mapping it in the client (that is, in NaturalONE or Natural Studio). A dialog will be shown for setting up the connection in which you have to specify the following information:

Server

Host name	The host name defines the remote node name where the server is running (or the IP address of the server).
Server port	The server port defines the TCP/IP port number for the development server.
Environment name	The environment name can be used to give the addressed server a logical (descriptive) name. If this box is left blank, a default name will be created automatically.

Startup

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	Your user ID is automatically provided.
Password	If Natural Security is installed on the development server, specify the required password in this text box. Otherwise, leave this text box blank.

These settings are transferred to the selected Natural Development Server and evaluated to create an exclusive Natural session that is responsible for executing all development requests for that environment. Once you have successfully mapped a development server, the Natural objects of the connected remote development environment are shown in NaturalONE or Natural Studio.

Using the Remote Development Functionality

You can use the entire functionality of NaturalONE or Natural Studio to create, edit, store or execute Natural objects on the remote Natural environment. You can map to multiple environments from one NaturalONE or Natural Studio. Each mapped environment owns a Natural session on the Natural Development Server, even if you map multiple environments on the same server.

When you are working with NaturalONE, it is recommended that you work in the so-called "local mode". In local mode, the sources are no longer stored or modified directly on the development server. The central place for keeping the sources is now the Eclipse workspace which is connected to a version control system.

2 Development Server File

- Purpose of the Development Server File 6
- Relations between FDIC and the Development Server File 6
- Unique Development Server File 6

This chapter describes purpose and use of the Natural Development Server file, a central dictionary file that is structurally identical to the Natural system file `FDIC`.

Purpose of the Development Server File

As Natural stores its data in system files, Natural Development Server stores its data in the system file that is assigned to the Natural parameter `FDIC`, a logical system file which is called the “development server file”.

The development server file is used as a central dictionary file for storing Natural applications and the links to objects making up an application. It also holds object locking information. This information is not bound to certain groups of application developers, but has an impact on the entire application development of an enterprise. Therefore, this file should be available only once, to ensure that the application definitions and locking states are kept consistent.

Relations between FDIC and the Development Server File

The development server file layout corresponds to the file layout of the Natural system file `FDIC` used by Predict. This means that the central dictionary file can also be used to hold Predict data, but Predict is not a prerequisite for using the development server file. This enables you to use your existing application documentation in the application definitions of the remote development environment.

Unique Development Server File

It is of vital importance that the various remote development environments that can be mapped use a common and unique development server file.

Non-compliance with this requirement may give rise to inconsistencies in object locking and in the applications existing in the application workspace.

Under Natural Security

In a Natural Development Server that is protected by Natural Security, the use of another `FDIC` file in the application workspace is prevented if the application security profiles are activated. See also *Application Protection* in the *Natural Security* documentation.

3

Installing Natural Development Server under UNIX

Natural Development Server is delivered with the applicable version of Natural for UNIX.

For information on how to install Natural Development Server, see the *Installation* documentation for Natural for UNIX.

4 **Configuring Natural Development Server under UNIX**

- User-Specific Environments 10
- NDV User Exit Routine 10

User-Specific Environments

A Natural development server can be assigned user-specific environments. These environments can be set within the Natural Configuration Utility.

When a client maps to a server, the server is accessing the NDV environment section of the corresponding port inside the global configuration file and is looking for the corresponding user ID. If the user ID section is found, the corresponding environment will be created and the new Natural session will be started with the following environment:

- Server Environment
- User Environment

If no environment is available for the current user, only the server environment will be taken as the Natural session's environment.

The variable `NDVSESSION` is provided to control the starting of Natural sessions. With this variable it is possible to run a server with the ability to start different natural sessions depending on the mapped user.



Note: The `NDVSESSION` setting has a higher priority than the `-start` option of the server.

Suppose an NDV server is started with the option `-s=naturalsec` and a user has specified `NDVSESSION=mynatural`, then the session "mynatural" will be launched for this user.

For further information, refer to *NDV Environment Assignments* in the *Natural for UNIX Configuration Utility* documentation.

NDV User Exit Routine

Natural Single Point of Development provides the following user exit:

NDV-UX01	This exit is invoked before a Natural source object or a DDM is edited. It can be used to reject editing of certain sources. The source code of this exit is delivered in the library <code>SYSLIB</code> and named <code>NDV-SX01</code> *).
----------	---

*) The sources of these user exit routines are named `NDV-SXnn`, where `nn` denotes the number of the user exit routine.

➤ To make a user exit routine available

- 1 Copy the source code from `SYSLIB` into a user library.

- 2 Catalog it under the name `NDV-UXnn`.
- 3 Copy it back into the Natural system library `SYSLIB`.

The name of each user exit source is different from the name of the corresponding cataloged object. This guarantees that the object is not affected if the user exit source is overwritten by an installation update.

For further details, see the source code of the user exit routines `NDV-SXnn` in the Natural system library `SYSLIB`.

5 Operating Natural Development Server

■ Introduction	14
■ Starting the Development Server	14
■ Determining the Screen Size for Web I/O	15
■ Terminating the Development Server	16
■ Getting Help	16
■ NDV Configuration File	17
■ Trace	17
■ Error Messages	18

This chapter describes how to operate Natural Development Server in a UNIX-based environment.

Introduction

Natural Development Server for UNIX represents a listener process which waits for incoming Natural clients to connect. For every connection request, the listener starts a Natural process which will then communicate with the corresponding client. Natural Development Server can only be run in a valid Natural environment. For the currently applicable version of Natural for UNIX, refer to Empower at <https://empower.softwareag.com/>.

Starting the Development Server

The development server can be started as follows:

```
natdvsrv [start-options] [Natural-parameters]
```

where

-p[ort][=]<port>	Mandatory start option. Starts the server on the specified port.	
-c[lient][=]<type>	Optional start option. Defines the types of clients which are permitted to connect to the server. <type> can be one of the following:	
	NAT	Only Natural for Windows (Natural Studio) clients are permitted to connect to this server.
	ONE	Only NaturalONE clients are permitted to connect to this server.
	ANY	Both Natural for Windows (Natural Studio) and NaturalONE clients are permitted to connect to this server.
	If this start option is not specified, the server will use ANY as the default. If a client which is not permitted tries to connect to the server, Natural error NAT7742 will be issued, indicating that the current IDE is not permitted.	
	Note: If Natural Security is installed, the types of clients permitted to connect to the server can be controlled by setting the Natural Security option Natural client access .	
-s[tart][=]<image>	Optional start option. The Natural image to be used as the server session. Any valid Natural image can be used. If this start option is not specified, the server will use "natural" as the default session image.	
	For the currently applicable version of Natural for UNIX, refer to Empower at https://empower.softwareag.com/ .	

[<i>Natural</i> -parameters]	Optional. The Natural parameters to be used in conjunction with the session image. The syntax is identical to starting a Natural session with Natural parameters.
-------------------------------	---

Examples

```
natdvsrv -port=1234
```

starts the listener on port 1234, (implicitly) uses the image "natural" as the server session, and (implicitly) permits all types of clients to connect to the server.

```
natdvsrv -p=1234 -c=ONE
```

starts the listener on port 1234, (implicitly) uses the image "natural" as the server session, and permits only NaturalONE to connect to the server.

```
natdvsrv -p1234 -cNAT parm=myparm stack=(LOGON mylib)
```

starts the listener on port 1234, (implicitly) uses the image "natural" as the server session, and permits only Natural for Windows (Natural Studio) to connect to the server. When a client session connects, the listener starts Natural with the parameter module "myparm" and applies a logon to "mylib".

```
natdvsrv -p=1234 -s=naturalsql
```

starts the listener on port 1234, uses the image "naturalsql" as the server session, and (implicitly) permits all types of clients to connect to the server.

Determining the Screen Size for Web I/O

By default, a screen size of 80x24 is used for web I/O with the Natural Web I/O Interface. This is determined by the following variables:

Variable Name	Description
NWO_NCOLUMNS	The number of columns to be used. Default: 80.
NWO_NROWS	The number of rows to be used. Default: 24.

If a different screen size is to be used, you have to set these variables *before* starting the development server.

Example

```
#!/bin/sh
NWO_NCOLUMNS=100
export NWO_NCOLUMNS
NWO_NROWS=50
export NWO_NROWS
natdvsrv -port=4714 parm=myparm
```

This shell script starts a development server on port 4714 with a screen size of 100x50.

Terminating the Development Server

The development server can be terminated as follows:

```
natdvsrv -t[erminate][=]<port>
```

where *<port>* is the port number of the development server to be terminated.

Terminating the server will also terminate all Natural sessions started by the server.



Note: Alternatively, a development server session can be terminated via the `kill <process-id>` command.

Example

```
natdvsrv -t1234
```

terminates the listener and all Natural sessions launched by the listener.

Getting Help

```
natdvsrv [-h[e]lp]
```

displays the help information on the available options.

NDV Configuration File

When a new development server is started, the file *NDVSERVER.PRU* located in your installation directory, in the subdirectory *Natural\Prof*, will be accessed to add information about the server.

When a server is terminated, the following steps are performed:

1. A search of the specified port takes place in order to find the server's process ID. This process ID is then used to terminate the server.
2. The entry of the terminated server in the configuration file will then be removed.

Trace

A server trace can be established via the built-in trace macros.

 **Important:** In order to generate trace files for the listener and all started sessions, a plus sign (+) must be used as the trace file name. Otherwise, a new development server session will overwrite an existing trace file.

The name of the listener trace file is constructed as shown below:

```
natdvsrv_port
```

where

<i>port</i>	is the assigned port number.
-------------	------------------------------

The name of a session trace is constructed as shown below:

```
natdvsess_ipinfo_ pid_ port
```

where

<i>ipinfo</i>	is the client connection's IP address/name.
<i>pid</i>	is the process ID of the development server session.
<i>port</i>	is the port of the listener process.

Example

```
setenv NCTR_STATE 1
setenv NCTR_CONFIG path/nctr.cfg
setenv NCTR_OUTPUT /tmp/+
```

After a listener process was started on port 4712 and two clients have connected, the following trace files are generated into the directory /tmp/:

```
natdvsrv_4712
natdvsess_pc1_1055_4712
natdvsess_pc2_1045_4712
```



Note: Be careful when using the plus sign (+) notation. This could cause a lot of trace files to be generated.

Error Messages

When a development server is started or terminated, the following error messages can occur:

Error Number	Error Message	Action
66	The port number is missing.	Provide a valid port number.
67	The specified port number is already in use.	Use a port which is not yet occupied on the machine.
68	Invalid syntax [% <i>syntax</i> %] encountered.	The options used to start or terminate the server are used wrongly according to [% <i>syntax</i> %].
70	Invalid port number encountered.	Use a correct port number inside the valid range of 0 - 9999 and do not use other characters or digits.
71	Listen on specified port failed.	Try again with a different port number.
78	The specified server session [% <i>session-name</i> %] is not accessible.	Check that the specified session can be found via the PATH environment variable or that it is located inside \$NATBIN.
79	The port number [% <i>number</i> %] exceeds upper limit (99999).	Use a port inside the valid range.
80	Invalid combination of options [% <i>options</i> %] encountered.	Check that the specified options can be used together.
81	NDV server could not be terminated (Reason: % <i>reason</i> %).	Check the reason: <ul style="list-style-type: none"> ■ Check that the server with the specified port is running.

Error Number	Error Message	Action
		■ Check that you have the rights to terminate the server.
82	Error accessing file 'NDVSERVER.PRU'.	Check that the file is available in and that it has a valid content. If the file is corrupted, delete the file.

6 SPoD-Specific Limitations and Considerations

- Limitations 22
- Performance Considerations 24
- Natural Documentation and Online Help 25

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. In addition, this document includes hints which are important for the efficient use of the remote development facilities.

Editor Features With SPoD

You can use Natural's Single Point of Development with different versions of Natural on a variety of platforms. Depending on the server environment you are using together with Natural for Windows (client), the editors offer different features. For further information, refer to the section *Editor Features With SPoD* in the *Natural for Windows Editors* documentation.

Limitations

- [System Commands](#)
- [Terminal Commands](#)
- [Moving/Copying Error Messages](#)
- [Maps Containing GUI Elements](#)
- [Field Sensitive Maps](#)
- [Dependencies between XRef Evaluation and Predict](#)
- [Debugger](#)

System Commands

- [System Commands Entered Directly on the Development Server](#)

System Commands Entered Directly on the Development Server

All system commands which are not entered in the user interface of Natural Studio 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. This is the case when the `STACK TOP COMMAND` mechanism is used or when a system command is directly entered inside the terminal emulation window.

During the mapping phase any `STACK` commands entered in the text box **Session Parameters** are processed within Natural Studio and the corresponding Natural Studio windows are used.

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. Instead of the menu-driven user interface, the dialogs provided in Natural Studio are used.

Terminal Commands

Using terminal commands in a SPoD environment is only possible within the terminal emulation window. Entering terminal commands in the command line of Natural Studio is not possible.

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.

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 user input has been entered. Field sensitive maps can be moved or copied from the local environment to a remote environment. However, a field sensitive map cannot be tested or executed on a remote mainframe environment.

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.

Debugger

It is currently not possible to reliably debug a Natural application using ETID-based statements like `GET TRANSACTION DATA`.

The reason is that the debugger is started in a separate NDV server background session which cannot use the same ETID as applied in the NDV development session. The debug session always is using a generated ETID which differs from the ETID used by the application.

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. 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 100 Mbit/s at minimum.

Another possibility to reduce the amount of data read while mapping the environment is to supply filter definitions on system file or library level.



Tip: In the context menu of a system file and library node it is possible to apply filter definitions. Using these definitions on the client side, you can limit the number of libraries/objects displayed in the tree view.

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 a very time consuming process in the case of very large system files.



Tip: By default, the **Refresh** function is set to **Full automatic refresh**. Change the automatic refresh function by choosing **Optimized automatic refresh** or **No automatic refresh** 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.

Natural Documentation and Online Help

The following restrictions apply to the Natural documentation and the Windows-based online help when you are using a Natural Development Server (NDV) for remote development:

- The online help currently available with Natural Studio contains only the Natural for Windows documentation and the SPoD client documentation.
- Therefore this online help may describe Natural features which are not or not yet supported on the UNIX platform.
- Natural features that are available only on UNIX are missing.
- Particularly in the sections dealing with the Natural programming language, minor but important differences due to hardware platforms, operating systems, etc. may exist.

We ask you to refer to the Natural for UNIX documentation set for full details.

