



natural

natural
Development Server

Version 2.1.2

For VSE/ESA

This document applies to Natural Development Server Version 2.1.2 for VSE/ESA 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.

© Copyright Software AG 2001 - 2003.
All rights reserved.

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

Table of Contents

| | |
|--|----|
| Natural Development Server | 1 |
| Natural Development Server | 1 |
| Additional Terms | 2 |
| Additional Terms | 2 |
| Legal Notices | 3 |
| Legal Notices | 3 |
| Copyrights and Disclaimers of Included Third Party Products | 4 |
| Copyrights and Disclaimers of Included Third Party Products | 4 |
| HiHelpIndex, PHD Computer Consultants Ltd. | 4 |
| Spy-CD, PHD Computer Consultants Ltd. | 4 |
| Software AG License Agreement for Trial Versions of Software AG Software ONLY | 9 |
| Software AG License Agreement for Trial Versions of Software AG Software ONLY | 9 |
| Using this Documentation | 11 |
| Using this Documentation | 11 |
| Navigating through the Documentation | 11 |
| Navigation Bar | 11 |
| Navigator Facility | 11 |
| Searching this Documentation | 12 |
| Printing this Documentation | 12 |
| What's New? | 13 |
| What's New? | 13 |
| Product Enhancements and Corrections | 13 |
| Support of Object Type Function in SPoD Environments | 13 |
| Documentation | 13 |
| Dropped Functionality | 14 |
| End of Maintenance of Natural Development Server Versions | 14 |
| Introducing the Natural Development Server | 15 |
| Introducing the Natural Development Server | 15 |
| Purpose of a Natural Development Server | 15 |
| Remote Development Functions | 15 |
| Development Server File | 17 |
| Development Server File | 17 |
| Purpose of the Development Server File | 17 |
| Relations between FDIC and the Development Server File | 17 |
| Unique Development Server File | 17 |
| Natural Development Server on Mainframes | 18 |
| Natural Development Server on Mainframes | 18 |
| Development Server Concept | 18 |
| Front-End Stub NATRDEVS | 19 |
| Stub Description | 19 |
| Natural System Variables Used | 19 |
| Natural I/O Handling | 19 |
| Front-End | 20 |
| Transaction Processors | 20 |
| Gateway Module | 20 |
| Server Monitor | 20 |
| Product Interaction | 20 |
| Prerequisites | 22 |
| Prerequisites | 22 |
| General Prerequisites for NDV Installation | 22 |
| Prerequisites for NDV under SMARTS on VSE/ESA | 22 |
| Prerequisites/Restrictions for NDV CICS Adapter | 23 |

| | |
|---|----|
| Installing the Natural Development Server | 24 |
| Installing the Natural Development Server | 24 |
| Prerequisites | 24 |
| Content of the Development Server Distribution Tape | 24 |
| Installation Procedure | 24 |
| Copying the Tape Contents to Disk | 25 |
| Step 4: Create a Development Server Configuration File | 26 |
| Step 5: Create a SMARTS Sysparms file | 26 |
| Step 6: Load FDIC system file | 26 |
| Step 7: Assemble and link ADALNK | 26 |
| Step 8: Assemble and catalog for the Natural NDV interface | 26 |
| Step 9: Create the NDV server front-end module | 27 |
| Step 10: Link the Natural nucleus | 27 |
| Step 11: Delete Old Objects | 27 |
| Step 12: Load Natural objects, error messages and samples for NDV | 27 |
| Step 13: Copy DDMs and processing rules to FDIC | 27 |
| Step 14: Extend your SMARTS startup job by NDV-specific definitions | 28 |
| Step 15: NDV Clients must be defined to Natural Security | 28 |
| Configuring the Natural Development Server | 29 |
| Configuring the Natural Development Server | 29 |
| Configuration Requirements | 29 |
| Configuration File | 31 |
| Configuration Parameters | 31 |
| DBG_CODEPAGE | 31 |
| FRONTEND_PARAMETER | 32 |
| INITIAL_USERID | 32 |
| FRONTEND_NAME | 32 |
| TRACE_LEVEL | 33 |
| DEFAULT_PROFILE | 33 |
| SESSION_PARAMETER | 33 |
| HOST_NAME | 33 |
| PORT_NUMBER | 34 |
| Server Datasets | 34 |
| Operating the Natural Development Server | 35 |
| Operating the Natural Development Server | 35 |
| Starting the Natural Development Server | 35 |
| Terminating the Natural Development Server | 35 |
| Monitoring the Natural Development Server | 35 |
| Monitor Communication | 35 |
| Monitor Commands | 36 |
| Runtime Trace Facility | 36 |
| Trace Medium | 36 |
| Trace Configuration | 36 |
| Trace Level | 37 |
| Trace Filter | 37 |
| Natural Development Server Frequently Asked Questions | 39 |
| Natural Development Server Frequently Asked Questions | 39 |
| Introducing the Natural Development Server CICS Adapter | 45 |
| Introducing the Natural Development Server CICS Adapter | 45 |
| Purpose of the Natural Development Server CICS Adapter | 45 |
| Remote Development Functions | 45 |
| CICS Support | 45 |
| Product Interaction | 46 |
| Installing the Natural Development Server CICS Adapter | 48 |
| Installing the Natural Development Server CICS Adapter | 48 |
| Prerequisites | 48 |

- Installation Procedure 48
 - Step 1: Customize CICS 48
 - Step 2: Customize the Development Server 48
- Configuring the Natural Development Server CICS Adapter 50**
 - Configuring the Natural Development Server CICS Adapter 50
 - Configuration File 50
 - Configuration Parameters 50
 - RFE_CICS_TA_NAME 50
 - RFE_CICS_FE_NAME 50
 - RFE_CICS_TA_HOST 51
 - RFE_CICS_TA_PORT 51
 - RFE_CICS_TA_INIT_TOUT 51
 - RFE_CICS_KEEP_TA 51
 - RFE_CICS_TRACE 52
- NDV CICS Adapter Frequently Asked Questions 54**
 - NDV CICS Adapter Frequently Asked Questions 54

Natural Development Server

The documentation in the list below applies to the Natural Development Server plug-in (Product code NDV) under SMARTS on VSE/ESA and the optional Natural Development Server CICS Adapter.

SMARTS is an acronym for "Software AG Multi-Architecture Runtime System". It constitutes a runtime layer that allows POSIX-like applications to run on mainframe operating systems. Software AG products communicate with the operating system through the SMARTS layer.

Natural Development Server Documentation

- [What's New?](#)
- [Introducing the Natural Development Server](#)
- [Development Server File](#)
- [Natural Development Server on Mainframes](#)
- [Prerequisites](#)
- [Installing the Natural Development Server under SMARTS on VSE/ESA](#)
- [Configuring the Natural Development Server](#)
- [Operating the Natural Development Server](#)
- [NDV Server Frequently Asked Questions](#)

The following documentation applies in addition if you want to use a Natural Development Server in a CICS environment:

Natural Development Server CICS Adapter Documentation

- [Introducing the Natural Development Server CICS Adapter](#)
- [Installing the Natural Development Server CICS Adapter](#)
- [Configuring the Natural Development Server CICS Adapter](#)
- [NDV CICS Adapter Frequently Asked Questions](#)

Related Documentation:

- [Natural Single Point of Development documentation](#)
- [Natural for Windows documentation](#)

Additional Terms

The license does not grant you the right to sublicense, rent, assign or lease the software, in whole or in part, and you may not decompile, disassemble, modify, decrypt, extract or otherwise reverse engineer, or make further copies of the software, except as explicitly permitted by the License Agreement.

Also, the Product contains certain third party technology that is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. You must not use or redistribute the Product for such purposes.

Due to certain third party technology contained in the Product, the Product is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. You must comply strictly with all such regulations and, subject to the License Agreement, are responsible to obtain licenses to export, re-export, or import the software. The software may not be downloaded, or otherwise exported or re-exported (i) into, or to a national or resident of Cuba, Iraq, Iran, North Korea, Libya, Sudan, Syria or any country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nations or the U.S. Commerce Department's Table of Denial Orders. Use, duplication or disclosure by the United States government is subject to the restrictions as set forth in the Rights in Technical Data and Computer Software Clauses in DFARS 252.227-7013 (c) (1) (ii) and FAR 52.227-19 (c) (2) as applicable.

Legal Notices

This notice is valid for all products of the Adabas®, Bolero®, EntireX®, Natural® and Tamino® (the "Product") product line and associated Software AG products.

IMPORTANT: PLEASE READ BEFORE INSTALLING THE SOFTWARE

You are not allowed to install or use the Product without a corresponding license agreement.

If you have entered into a license agreement with one of Software AG's subsidiaries or distributors (the "License Agreement") the installation and use of the Product is subject to your acceptance of additional terms which are provided for you in the section Additional Terms.

If you have not entered into a License Agreement you must execute the license agreement with Software AG which is provided for you in the section Software AG License Agreement for Trial Versions of Software AG Software ONLY.

Copyright Notice

Adabas®, Bolero®, EntireX®, Natural® and Tamino® are Products of Software AG, a German corporation with its principal place of business at Uhlandstrasse 12, 64297 Darmstadt.

Software AG is the sole owner of all industrial property rights and copyright to the Products and accompanying user documentation or has the respective distribution rights. References made in or on the Products to the copyright or to other industrial property rights must not be altered, deleted or obliterated in any manner.

© Copyright Software AG, Uhlandstrasse 12, 64297 Darmstadt, Germany 2003. All rights reserved. For third party copyright notices please see the appropriate section below.

Trademark Notice

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

Links to Third-Party Sites

Some links in this area will let you leave Software AG's documentation. Linked sites are not under the control of Software AG and Software AG is not responsible for the contents of any linked site or any link contained in a linked site. Software AG is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of the content of the site by Software AG.

PLEASE READ THE FOLLOWING NOTICES BEFORE USING THE SOFTWARE!

Copyright Notices and Disclaimers of Included Third Party Products

Copyrights and Disclaimers of Included Third Party Products

- HiHelpIndex, PHD Computer Consultants Ltd.
 - Spy-CD, PHD Computer Consultants Ltd.
-

HiHelpIndex, PHD Computer Consultants Ltd.

Hi HelpIndex Licence Agreement

The intellectual property, software, documentation and images for Hi HelpIndex, Hi Lab, Hi Jump, Hi Brow, Hi Help and MakeHelpIndex are Copyright (c) 1996-2002 PHD Computer Consultants Ltd.

PHD does not agree to fix any apparent defects or limitations in the software. However if you notify PHD of the problem or have positive suggestions then PHD may change the software accordingly in the next release.

You must not re-use, decompile or otherwise mis-use the Hi HelpIndex software.

Unless agreed otherwise, this agreement is governed by English law.

NO WARRANTIES. To the maximum extent permitted by applicable law, PHD expressly disclaims any warranty for this software. This software and its documentation are provided "as is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties or merchantability or fitness for a particular purpose. The entire risk arising out of use or performance of this software remains with you.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES. To the maximum extent permitted by applicable law, in no event shall PHD or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profit, loss of business information, business interruption, or any other pecuniary loss) arising out of the use or inability to use this PHD product.

1.6a July, 2002

Spy-CD, PHD Computer Consultants Ltd.

INTRODUCTION

This is a legal agreement between you, the Licensee, and PHD Computer Consultants Ltd (PHD), United Kingdom.

The licence covers the use of the PHD Spy-CD and Spy-Server products.

This text lists the "Free", "Temporary", "Personal", "Standard", "Gold", "Professional Subscription" and "Site" licence terms.

A Source Code Escrow service for Spy-CD is available from ComponentSource. Contact ComponentSource for details of the escrow licence agreement. <http://www.componentsource.com/>

The Spy-CD software includes the following modules: Spy-CD Java applet Search Engine for CDs and product help systems. SpyLight Java applet word highlighter. Spy-Server Java servlet Search Engine for web sites. Spy-CD-Wizard set up tool that runs in most Windows platforms. Spynindex Java indexing application. PDF Scanner module for scanning PDF files. Hi Help help program for Spy-CD-Wizard.

The features available are listed on the Spy-CD and Spy-Server documentation Web sites.
<http://www.phdcc.com/spy-cd/> <http://www.spy-server.com/>

You may purchase Spy-CD licences either direct from PHD or through one of our outlets, as listed on the Spy-CD Web site.

The Spy-Server development kit is not included in the Spy-CD development kit. It is available separately.

You may try out and use Spy-CD and Spy-Server under the Free or Temporary licence terms. Otherwise you must purchase a suitable Spy-CD licence to cover your use of Spy-CD or Spy-Server.

COPYRIGHT

The intellectual property, software, documentation and images for Spy-CD, Spy-Server, SpyLight, Spy-CD-Wizard, Spynindex, PDF Scanner and Hi Help are Copyright (c) 1997-2002 PHD Computer Consultants Ltd.

You must not re-use, decompile or otherwise mis-use the Spy-CD software.

FREE LICENCE

The Free licence lets you use the Spy-CD development kit on one computer. You may distribute the Spy-CD runtime on any number of CDs and put the Spy-CD or Spy-Server runtime on any number of web sites.

The Spy-CD runtime only displays the first 50 files. The Spy-Server runtime only provides clickable links for the first 50 files.

There is no PDF file scanning capability.

TEMPORARY LICENCE

A Temporary licence is the same as a Free licence, except that it releases some or all limitations on the software for a temporary period, as indicated by the PHD sales staff. A Temporary licence is for internal evaluation purposes and Spy-CD must not be put on a web site or distributed on CD.

PDF SCANNER

The PDF Scanner module is enabled in all licences apart from the Free licence.

The PDF Scanner module must only be used in conjunction with the Spy-CD or Spy-Server development kit on one computer. The PDF Scanner module must not be used on a server. For use on a non-web server, a Site licence is required.

Licensed under U.S. Patent 4,558,302 and its foreign counterparts. PHD has been granted a licence from Unisys Corporation to use the LZW algorithm in PHD products.

The PDF Scanner module is NOT licensed for use on personal computers or workstations functioning as a server. The PDF Scanner software is NOT licensed for use in providing on-line financial transactions or any other on-line electronic commerce, or for operating an on-line service, bulletin board, Web, Intranet or Internet access service, or for distribution by such entities for use in communicating with their services. The PDF Scanner software may NOT be used for operating network access servers, Web, Intranet or Internet servers or for software providing for on-line or real time operation with a Web, Internet or Intranet server, or for providing for uploading or publishing web pages or related data and/or images to a web server and/or Internet server. Additionally, the PDF Scanner software may NOT be used for software development purposes and/or for

bundling or for any other combination with products of an OEM, VAR or developer.

This software uses the RSA Data Security, Inc. MD5 Message-Digest Algorithm from RFC 1321.

PERSONAL LICENCE

A Personal Licence permits you to put the Spy-CD runtime on 200 CDs and display a maximum of 200 files. You can put the Spy-CD or Spy-Server runtime on one web site searching a maximum of 200 files.

STANDARD LICENCE

A Standard Licence permits you to put the Spy-CD runtime on 750 CDs. You can also put the Spy-CD or Spy-Server runtime on one Web site.

GOLD LICENCE

A Gold Licence permits you to put the Spy-CD runtime on 3000 CDs. You can also put the Spy-CD or Spy-Server runtime on four Web sites.

PROFESSIONAL SUBSCRIPTION LICENCE

A Professional Subscription Licence permits you to put the Spy-CD runtime on any number of CDs for one year, and put the Spy-CD or Spy-Server runtime on ten Web sites. If you produce less than 10,000 CDs in the first year, then you may continue to produce CDs after one year until 10,000 CDs have been produced.

SITE LICENCE

A Site licence extends a Standard, Gold or Professional Subscription licence. A Site licence permits you to use Spy-CD-Wizard or Spynindex and the PDF Scanner module on the number of non-server computers stated in the product description, and on the number of non-web servers stated in the product description.

USING SPY-CD IN A PRODUCT

If you use Spy-CD as the full text search facility for a product, you must include one of the following sentences in your copyright notice: "The full text search facility is provided by Spy-CD, (c) 1997-2002 PHD Computer Consultants Ltd, www.phdcc.com." or "Portions (c) 1997-2002 PHD Computer Consultants Ltd" You may optionally include either the PHD or Spy-CD logo. Please contact PHD for these logos.

RECORD KEEPING AND AUDIT

You must keep careful records of the number of CDs that you distribute with Spy-CD on, and ensure that you have the correct licences. CDs that are not distributed do not count towards the total for licence purposes.

You must pay for an appropriate Spy-CD licence for each of your web sites, whether available to the public or for internal use.

PHD has the right, no more than once during any calendar year, to have an independent accountant inspect the relevant records on seven business days notice and during normal business hours to verify that the correct licences are held. If you are found to have unlicensed use of Spy-CD-Wizard or PDF Scanner, kept no records, or have under recorded the number of CDs distributed or web sites by 10%, then you shall pay the cost of the audit. If you require further licences, you will promptly pay for these together with interest of 12%, dated from when the licences were required.

If you have a Professional Subscription licence then you do not need to keep records of the number of CDs distributed unless counting CDs beyond the first year end, and PHD will not be able to demand an inspection in this respect.

BACKUP

You may have a single working copy of the Spy-CD software on each licensed computer and keep a single copy on a backup medium.

UPGRADES AND SUPPORT

Purchasing a Personal, Standard, Gold, Professional Subscription or Site licence entitles you to upgrades and support for 12 months from the date of purchase.

You may obtain upgrades from the Spy-CD and Spy-Server web sites as directed in the documentation. PHD will not inform you of all upgrades.

PHD will make a reasonable attempt to answer any relevant queries from users with a Personal, Standard, Gold, Professional Subscription or Site licence for 12 months from the date of purchase. If your supplier provides support then please contact them first.

PHD does not agree to fix any apparent defects or limitations in the software. However if you notify PHD of the problem or have positive suggestions then PHD may change the software accordingly in the next release. Title to all new or revised software will remain with PHD.

PURCHASING LICENCE IDS

PHD supplies the development kit in electronic form only on the PHD web site. When you purchase a licence you only receive a licence id. This will enable the full features of the software and enable use of the runtime as defined in this licence. You do not receive a physical copy of the software.

For purchases direct from PHD, the licence is invalid until PHD has received the agreed payment in full.

You may not assign or resell your Spy-CD licence or otherwise make Spy-CD available to others without the consent of PHD.

You can have a refund if you have a legitimate problem with the software within 30 days of purchase provided you confirm that you have destroyed all copies of the licence id.

REDISTRIBUTABLE FILES

You must put the Spy-CD or Spy-Server redistributable files on your Web site or CDs in their entirety: the archive file(s), all the class and image files and any others as directed in the documentation.

You may redistribute the current Spy-CD free version development kit, provided it is distributed in its entirety. Please inform PHD so we know who inform for updates.

TERMINATION

You may terminate this licence provided you delete all copies of the Spy-CD software and licence ids from your computers, software products, CDs and recall all CDs issued.

PHD may cancel this licence upon 30 days prior written notice if you are in breach of any material obligations hereunder and the breach is not remedied within the notice period.

GENERAL

All Spy-CD licences are non-exclusive.

This text represents the entire agreement. In the event that any terms of this licence are deemed invalid, the remaining terms will still be in force.

Unless agreed otherwise, this agreement is governed by English law.

NO WARRANTIES

To the maximum extent permitted by applicable law, PHD expressly disclaims any warranty for this software. This software and its documentation are provided "as is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties or merchantability or fitness for a particular purpose. The entire risk arising out of use or performance of this software remains with you.

To the best of PHD's knowledge, the software is Year 2000 compliant.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

To the maximum extent permitted by applicable law, in no event shall PHD or its suppliers be liable for any damages whatsoever (including, without limitation, damages for loss of business profit, loss of business information, business interruption, or any other pecuniary loss) arising out of the use or inability to use this PHD product.

4.x February 2002

Software AG License Agreement for Trial Versions of Software AG Software ONLY

IMPORTANT: PLEASE READ THIS SOFTWARE AG LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING THE SOFTWARE! THE TERMS OF THIS LICENSE AGREEMENT APPLY ONLY IF YOU HAVE NOT ALREADY EXECUTED A LICENSE AGREEMENT WITH A SOFTWARE AG SUBSIDIARY OR DISTRIBUTOR:

LICENSE AGREEMENT FOR TRIAL VERSION

By installing the enclosed software, you, as licensee, are entering into an agreement with Software AG, as licensor, the terms of which are set forth below. If you do not agree to the terms of this agreement, promptly return all media with the enclosed software and the accompanying documentation and other written materials to Software AG before installation.

THE LICENSE

With this agreement, Software AG grants you - free of charge - a non-exclusive license to use the enclosed software and accompanying documentation (Software) on a single computer, on a workstation or on a single terminal within a network for evaluation and testing purposes for a limited time period defined in the license key file. In no event may the Software be deployed or used for any commercial production purpose unless you acquire a commercial license from Software AG and pay the applicable license fees. You may not pass on copies of the Software to any third party or transfer the Software by electronic means to other computers via a network. For multi-user hardware systems or networks, a separate license is required for each user or each workstation. You have the right to make one copy of the Software solely for archival and backup purposes. You may not decompile, disassemble, modify, decrypt, extract or otherwise reverse engineer, or make further copies of the Software or parts thereof. This license agreement, with the downloaded authorisation key, is proof of your entering into this license agreement and you must retain it. This license does not grant you the right to sublicense, rent, assign or lease the Software, in whole or in part.

THIRD PARTY RESTRICTIONS

The Software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. You warrant that you will not use the Software for such purposes.

LICENSE VALIDITY

This license is limited to a period defined in the license key file, starting from the date of installation of the Software.

COPYRIGHT

Software AG is sole owner of the industrial property rights and copyright to the Software and accompanying user documentation or has the respective distribution rights. References made in or on the Software to the copyright or to other industrial property rights must not be altered, deleted or obliterated in any manner.

LIMITED WARRANTY

The Software is provided "as is" without any warranty whatsoever. You assume full responsibility for the selection of the Software to achieve your intended results and for the installation, use and results obtained from the Software.

UPDATES AND MAINTENANCE

This license does not grant you any right, license or interest in and to any improvements, modifications, enhancements or updates to the Software and documentation or other support services. Such services are only available for regular commercial licenses.

LIMITATION OF LIABILITY

Under no circumstances shall Software AG be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, work stoppage, loss of data or other financial loss) arising from the use of, or inability to use, this Software.

TERMINATION

Software AG reserves the right to terminate this agreement immediately for good cause, whereby good cause is understood as any gross breach of this agreement.

EXPORT

The Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. You must comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import the Software. The Software may not be downloaded, or otherwise exported or re-exported (i) into, or to a national or resident of Cuba, Iraq, Iran, North Korea, Libya, Sudan, Syria or any country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nations or the U.S. Commerce Department's Table of Denial Orders. Use, duplication or disclosure by the United States government is subject to the restrictions set forth in the Rights in Technical Data and Computer Software Clauses in DFARS 252.227-7013(c) (1) (ii) and FAR 52.227-19 (c) (2) as applicable. Any other form of transfer or use of the Software which exceeds the scope of this license as described is prohibited.

MISCELLANEOUS

The invalidity of any provision of this agreement shall not affect any other part of this agreement. This agreement represents the complete and exclusive statement concerning this license between the parties. No modification or amendment of this agreement will be binding on any party unless acknowledged in writing by their duly authorized representatives. This agreement shall be governed and construed by the laws of the Federal Republic of Germany.

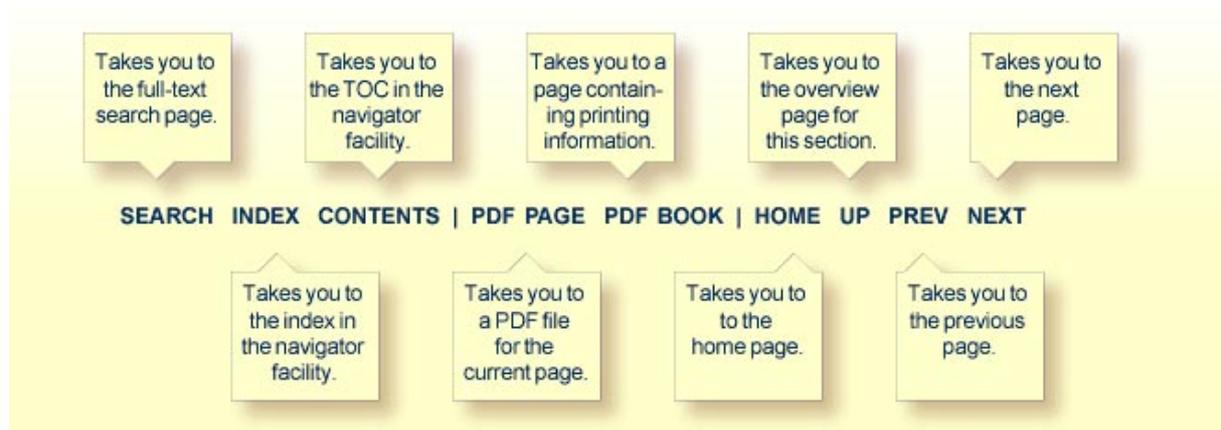
Using this Documentation

This document describes the features of the Software AG documentation that make it easy for you to **navigate** to the page you need, to **search** for keywords, and to **print out** relevant parts.

Navigating through the Documentation

Navigation Bar

In the top right-hand corner of each page you will find a navigation bar that helps you to navigate through the documentation. The navigation bar contains some or all of the following links:



Clicking the PREV link on the first page of a documentation module takes you up one level in the documentation hierarchy.

Clicking the NEXT link on the last page of a documentation module takes you back to the first page of the same module.

Navigator Facility

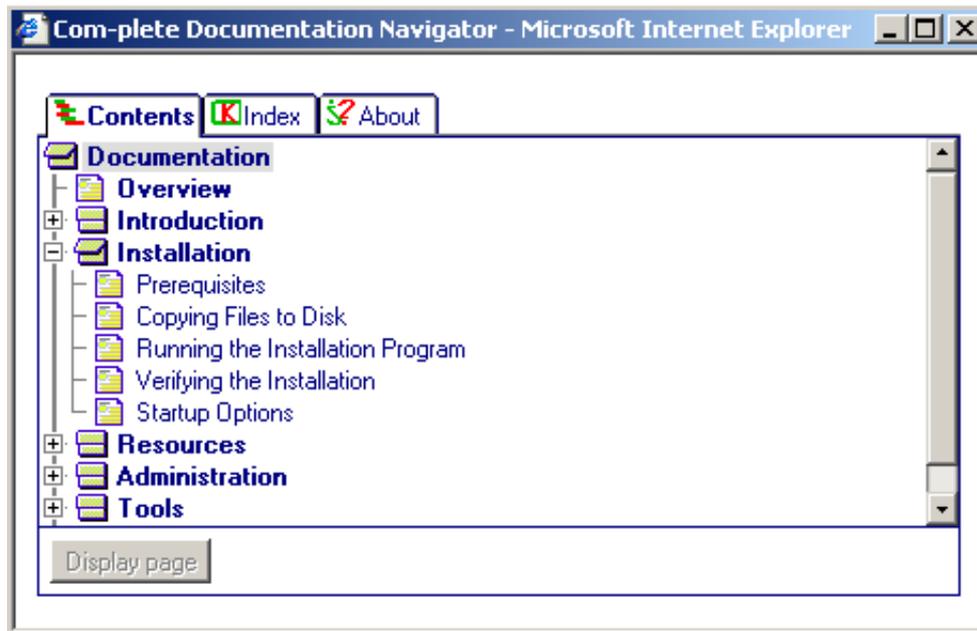
The navigator facility is a Java applet that runs in a separate window. You invoke it by choosing either INDEX (only available, if provided with the selected documentation set) or CONTENTS from the navigation bar on the HTML page.

- **INDEX** displays a list of index entries which help you find a particular location within the documentation. You can narrow the list down by entering a keyword or prefix in the input field. If you select an index entry from the list, the applet indicates the title(s) of the section(s) the index entry points to. You select a section title and then choose Display page.
- **CONTENTS** displays a collapsible/expandable, hierarchical tree view of the documentation, which helps you find your way through the documentation web. You can select in the tree view the document that interests you, then either choose the document title or choose Display page.

If you have been navigating in the documentation web, you can refresh the Tree Navigator display by choosing CONTENTS in the navigation bar. This will highlight your current location in the web.

Once the navigation applet is started, you can switch between the tree view (Contents) and the Index by selecting the appropriate tab, as illustrated in the figure below.

The navigator facility is a Java applet that displays a collapsible/expandable, hierarchical tree view of the documentation, which helps you find your way through the documentation web. You can select in the tree view the document that interests you, then either double-click on the document title or click on the 'Display page' button.



Searching this Documentation

Use the link SEARCH from the navigation bar to open the Full Text Search page.

The Full Text Search software is supplied by PHD Computer Consultants Ltd.

You can get detailed help for using the Spy-CD Full Text Search by clicking on the Help button on the Full Text Search page.

Printing this Documentation

This user documentation is also available in Adobe PDF (Portable Document Format) for convenient, high-quality printing. If you wish to use the PDF files, you must have appropriate PDF viewing software installed on your computer, for example, Adobe Acrobat Reader.

If Adobe Acrobat Reader is not already installed on your PC, you can download it by choosing:



Choose PDF PAGE in the navigation bar at the top of a page to open the corresponding PDF file in the Adobe Acrobat Reader.

If you choose PDF BOOK in the navigation bar at the top of a page, a list of available PDF books will be displayed. A PDF book contains the entire contents of the current documentation module, including a title page and a table of contents -- just like a regular book. Choose the required link to start the Adobe Acrobat Reader and open the corresponding PDF book. You can print either the entire book or just selected pages as required.

What's New?

This document describes the enhancements and corrections that apply to the release of Version 2.1.2 of the Natural Development Server (Product code NDV) under SMARTS on VSE/ESA.

The following topics are covered:

- Product Enhancements and Corrections
 - Support of Object Type Function in SPoD Environments
 - Documentation
 - Dropped Functionality
 - End of Maintenance of Natural Development Server Versions
-

Product Enhancements and Corrections

Version 2.1.2 of the Natural Development Server under SMARTS on VSE/ESA contains

- all ZAPs,
- INPL updates,
- early warnings and
- source changes

applied to the previous version as error corrections.

Support of Object Type Function in SPoD Environments

Natural Version 6.1 for Windows and UNIX enables the creation and execution of Natural objects of type Function. The source objects of type Function can be stored in a Natural system file on mainframe computers, but execution of such objects is not possible. Similar to source objects of type Dialog, the following functionality is supported:

- Sources of object type Function can be moved or copied to a remote mainframe environment by using copy-and-paste or drag-and-drop functionality.
- Objects of type Function existing in a remote mainframe environment are displayed in the tree view of the mapped environment.
- Remote editing and saving of objects of type Function is allowed.
- It is not possible to check, catalog and stow objects of type Function in a remote mainframe environment.

With Natural Development Server Version 1.1.5 or below, source objects of type Function existing in a remote mainframe environment are not displayed in the tree view of the mapped environment. Such objects cannot be moved or copied to a remote mainframe environment by using copy-and-paste or drag-and-drop functionality.

Documentation

The Natural Development Server documentation has been revised and updated.

See also the separate Natural Single Point of Development documentation.

Dropped Functionality

The System Management Hub is no longer used.

End of Maintenance of Natural Development Server Versions

Maintenance for Natural Development Server Version 1.1.3 ends on December 31, 2003.

Maintenance for Natural Development Server Version 1.1.4 ends on March 31, 2004.

Maintenance for Natural Development Server Version 1.1.5 ends on November 31, 2004.

Any extensions to the maintenance for Natural Development Server Version 1.1.3 or 1.1.4 will be published via ServLine24 (<http://servline24.softwareag.com>).

Software AG strongly recommends that you migrate to Natural Development Server Version 2.1.2 as soon as possible.

Introducing the Natural Development Server

This document describes the purpose and the functions of the Natural Development Server (Product code NDV).

The following topics are covered:

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

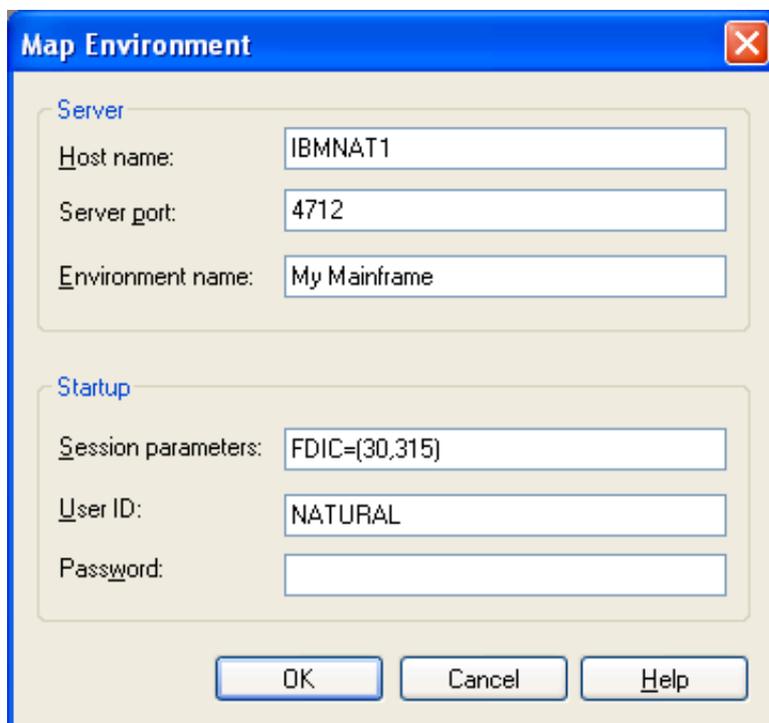
Purpose of a Natural Development Server

The Natural Development Server enables you to use the Natural Studio development environment to develop and test Natural applications in a remote Natural environment.

For more information, refer to the current Natural for Windows documentation and the Natural Single Point of Development documentation.

Remote Development Functions

A connection to an active Development Server can be established with the Map Environment function available in the Tools menu of Natural Studio. The Map Environment dialog is displayed for setting up the connection.



The host name defines the remote node name where the server is running (or the IP address of the server).

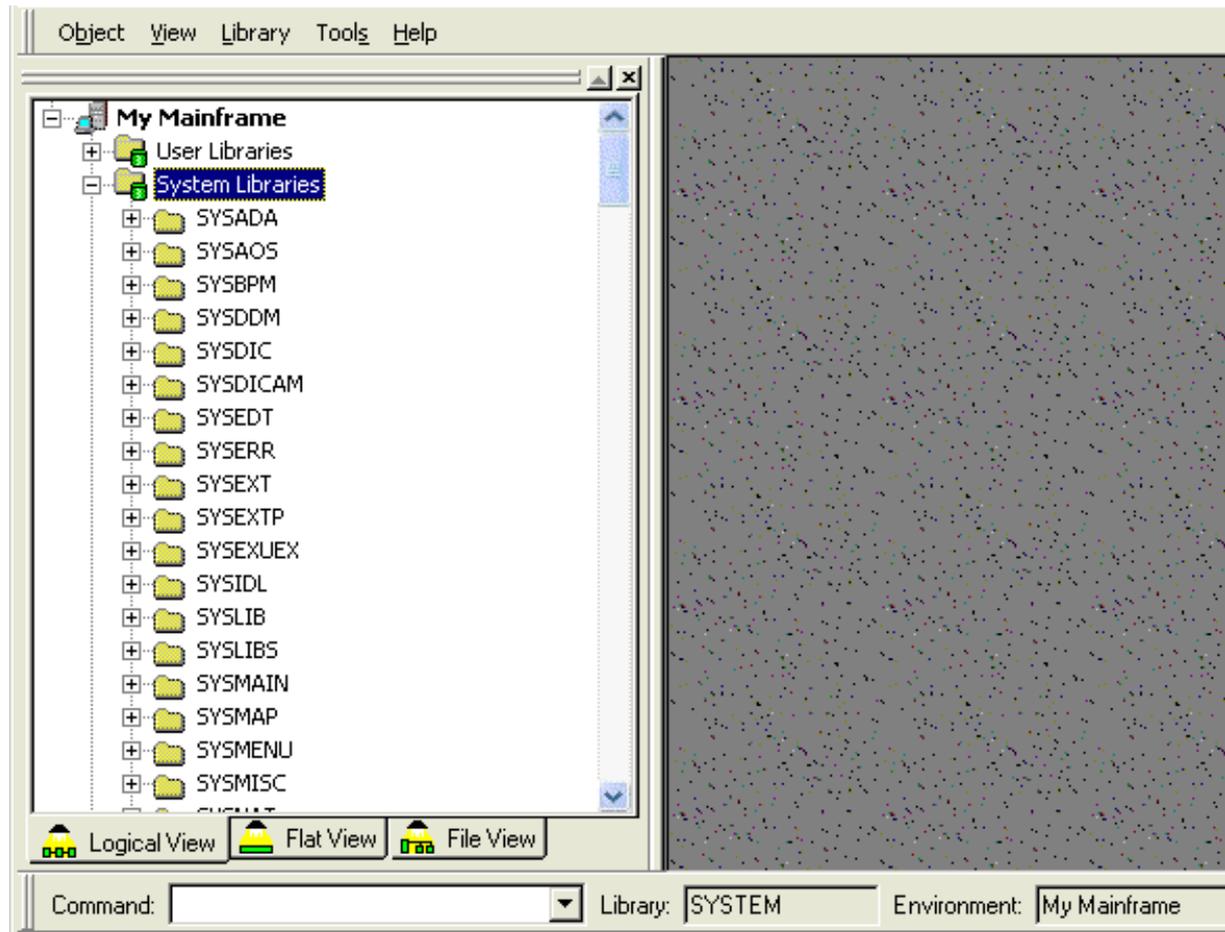
The server port defines the TCP/IP port number for the server.

The environment name can be used to give the addressed server a descriptive name.

You can specify additional Natural session parameters and a Natural Security password.

Both 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 executed a Map Environment command, Natural Studio opens a new tree view that represents the Natural objects of the connected Natural development environment, for example:



Note:

The node of a mapped server will reappear in the tree view when a new Natural session is started unless it is deleted with the corresponding context menu function.

You can use the entire functionality of Natural Studio to create, edit, store or execute Natural objects on the remote Natural environment.

You can map to multiple environments from one Natural Studio. Each mapped environment owns a Natural session on the Natural Development Server, even if you map multiple environments on the same server. However, only one mapped environment can be active at a time. The active environment is shown in the status bar. In addition, the corresponding node is highlighted in the tree view. It is this environment in which all remote commands will be executed.

You can change the active environment, by simply clicking on the desired node in the tree view.

Development Server File

This document describes the purpose and use of the Natural Development Server File, a central dictionary file which is used to store applications and the links to objects making up an application. It also holds object locking information in a Single Point of Development environment.

The following topics are covered:

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

Purpose of the Development Server File

As Natural stores its data in system files, the 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 as of Predict Version 4.2 or 4.3. This means that the central dictionary file can be also 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.

Attention: If Predict is used together with Natural Version 4.1.2 or together with NDV 2.1, Predict Version 4.3 is a prerequisite

Unique Development Server File

It is of vital importance that the various development environments 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.

To prevent the FDIC parameter from being overwritten when a Natural Development Server is mapped, you are strongly recommended to prevent the NTDYNP macro from being used to specify FDIC as a dynamic parameter.

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

Natural Development Server on Mainframes

This document describes the concept and the architecture of the Natural Development Server (Product code NDV) which is designed for use under SMARTS on VSE/ESA .

In addition, an optional Natural Development Server CICS Adapter is available that enables Natural Development Servers for OS/390 or SMARTS/VSE to be used with a CICS TP monitor.

The following topics are covered:

- Development Server Concept
 - Front-End Stub NATRDEVS
 - Transaction Processors
 - Front-End
 - Gateway Module
 - Server Monitor
 - Product Interaction
-

Development Server Concept

A Natural Development Server is a multi-user, multi-tasking application. It can host Natural sessions for multiple users and execute their requests concurrently.

The concept is based on the "serverized" Natural runtime system. Its architecture comprises a server front-end stub (development server stub NATRDEVS) that uses the Natural front-end to dispatch Natural sessions and to execute functionality within these sessions.

The Natural remote development server architecture basically consists of:

- Front-end stub
The stub NATRDEVS is launched to initialize a Natural Development Server, listens for incoming transactions and dispatches the appropriate Natural session for executing the transaction.
- Front-end
The front-end is called (together with the Natural runtime) by the front-end stub for session initialization/termination, request execution and session rollin/rollout.
- Gateway module
The module NATGWSTG provides for interaction between the Natural runtime and the front-end stub. NATGWSTG is linked to the Natural nucleus and is called by the Natural runtime to exchange the necessary request data.
- Transaction processors
Transaction processors are called by the front-end stub. The application logic of each individual transaction is implemented within a transaction processor.
- Server monitor
A monitor task allows the administrator to control the server activities, to cancel particular user sessions or to terminate the entire server, etc.

Front-End Stub NATRDEVS

The multi-user, multi-tasking, front-end stub NATRDEVS is launched to initialize a Natural Development Server.

- Stub Description
- Natural System Variables Used
- Natural I/O Handling

Stub Description

The task executing the server initialization (TMain) basically is the main listener which waits for incoming requests from the Remote Development Client. It owns a session directory to manage the multiple clients and their corresponding Natural sessions. TMain has the task to accept all incoming requests and to dispatch them to other subtasks (TWork). The process is as follows:

- First, a Map Environment command issued on the client side (in the Tools menu of Natural Studio) connects to TMain to establish a connection.
- Next, TMain inserts the client into its session directory, attaches a new TWork and passes the connection to TWork.
- TWork processes the request (indeed initializes a new Natural session if the client sends a CONNECT request) and replies to the client.
- After the reply, TWork listens on that connection for successive requests of that particular client. TWork remains active until the user on the client side (Natural Studio) switches the focus to a different environment (the local or a different mapped environment).
- If the user activates the environment again, TMain launches a new TWork that resumes the existing Natural session from the previous TWork.

That is, each client owns one subtask TWork on the Natural Development Server and multiple Natural sessions (one for each mapped environment). The subtask remains active as long as the mapped environment on Natural Studio is the currently active environment. Each Natural session remains active until the client disconnects/unmaps the corresponding environment. Consequently, a Natural session can be executed under different subtasks if the client switches among multiple environments.

Natural System Variables Used

Within a Natural Development Server session, the following Natural system variables are used:

- *TPSYS contains 'SERVSTUB',
- *DEVICE contains 'VIDEO' and
- *SERVER-TYPE contains 'DEVELOP'.

Natural I/O Handling

The Natural runtime allows I/O execution in the same way as in an online environment:

- A Natural Development Server intercepts the I/O and sends the 3270 data stream to Natural Studio.
- Natural Studio internally starts a terminal emulation window and passes the 3270 stream to that window.
- After I/O execution, the I/O data is sent back to the server.
- The front-end stub invokes the front-end to continue processing after I/O.

Front-End

OS/390 Batch: The Natural front-end required for a Natural Development Server is a Natural batch driver assembled with the option LE370=YES.

SMARTS: The Natural Version 3.1 front-end required for a Natural Development Server is a Natural Complete driver NCFNUC that is delivered with the Natural Development Server. You must not use the NCFNUC driver delivered with Natural Version 3.1 for Mainframes.

The Natural Version 4.1 front-end required for a Natural Development Server is the Natural Complete driver NCFNUC that is delivered with Natural Version 4.1 for Mainframes.

CICS: The Natural front-end required for executing the Natural sessions under control of CICS is the Natural remote front-end NATCSRFE that is delivered with the Natural Development Server. For more information, refer to the Natural Development Server CICS Adapter documentation.

Transaction Processors

The transaction processors are Natural programs in the library SYSLIB that process transactions (e.g. "save source", "get library list",) requested by the Natural Development Server client. The transaction processors are invoked by the front-end stub.

Gateway Module

The gateway module NATGWSTG must be linked to the Natural nucleus.

For CICS support, the Natural Development Server distribution tape contains a remote gateway module NATRGNVD. This module is responsible for transmitting the NDV-relevant data between a Natural Development Server and the Natural session running in CICS.

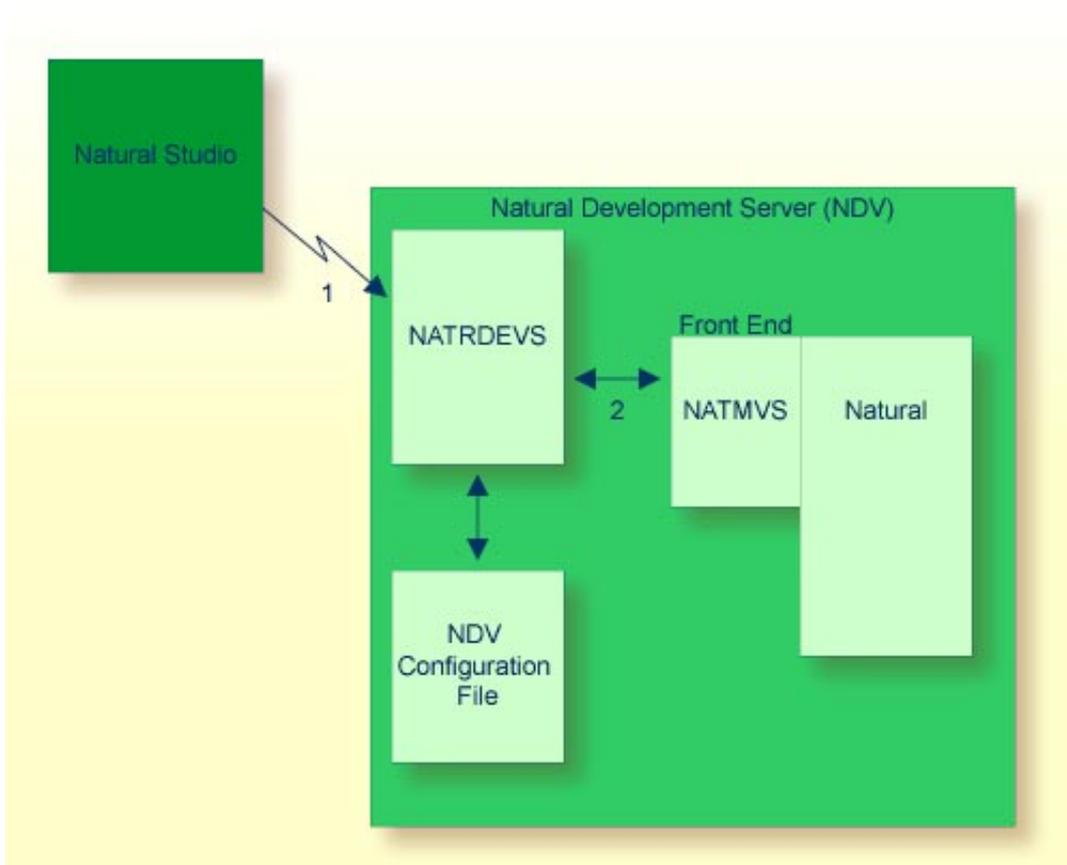
For more information, refer to the Natural Development Server CICS Adapter documentation.

Server Monitor

To enable the administrator to monitor the status of the Natural Development Server, a monitor task is provided which is initialized automatically at server startup. Using the monitor commands, the administrator can control the server activities, cancel particular user sessions, terminate the entire server, etc. See Operating the Development Server.

Product Interaction

The following figure illustrates the interaction of Natural Studio used as a remote development client with a Natural Development Server.



1. Natural Studio sends a remote development request to the Natural Development Server (NDV) using the port number specified with the NDV configuration variable PORT_NUMBER.
2. Natural Development Server dispatches the Natural session using the Natural front-end you have specified with the NDV configuration variable FRONTEND_NAME (NATMVS in this example).

Prerequisites

This document describes the prerequisites that apply when you install a Natural Development Server (Product Code: NDV) on a mainframe computer.

The following topics are covered:

- General Prerequisites for NDV Installation
 - Prerequisites for NDV under SMARTS on VSE/ESA
 - Prerequisites/Restrictions for NDV CICS Adapter
-

General Prerequisites for NDV Installation

- The current Natural Version for Mainframes must be installed.
- If Natural Version 3.1.6 is installed, Natural Version 3.1.6. Service Pack 8 and Zap NA46118 are required.
- If Natural Security Version 3.1.6 is installed, Natural Security Version 3.1.6 Service Pack 8 is required.
- If you are using Predict and you have to migrate to a Predict version specified under Natural and Other Software AG Products in the current Natural Release Notes, you are strongly recommended to migrate to the newer Predict version **before** you install the Natural Development Server.
The minimum version of Predict needed for the Natural Development Server Version 2.1 is Predict Version 4.3.1.

-



If you do not migrate to a Predict version specified under Natural and Other Software AG Products in the current Natural Release Notes **before** starting the Natural Development Server installation, you will have to define a new Natural system file (FNAT) and a new Development Server File (FDIC). The current Natural version for Mainframes and the desired additional products must have been loaded on the Natural system file FNAT before you start the installation of the Natural Development Server.

- The Software AG Editor must be installed. You are recommended to set the size of the editor buffer pool to 1024 KB.
If you are using SMA, the necessary modules are linked when the SMA parameter SAG-EDITOR is set.
If you are installing without SMA, see the Natural Installation Guide for Mainframes, Installing the Software AG Editor.
- The prerequisites required for the operation of a remote development client must be fulfilled in addition.
Natural Version 5 (restricted functionality) or Version 6 for Windows is required.
- Optionally:
 - Natural Construct Version 4.5.1 or higher.
If you have Natural Construct Version 4.5.1 installed, apply the latest Natural Construct Version 4.5.1 Service Pack to your Natural system file (FNAT).
 - Natural Engineer 4.4.1 or higher.

Prerequisites for NDV under SMARTS on VSE/ESA

In addition to the general prerequisites described above, the following operating-system-specific prerequisites apply:

- VSE/ESA must be installed.
Version as specified under *Operating/Teleprocessing Systems Required* in the current Natural Release Notes.
- SMARTS Version 2.7.1 Patch Level 5 (with the hotfix of Problem 242701 applied) or above must be installed (product code APS).
- Under Natural Version 3.1.6, NDV cannot access the editor work file. Therefore, you must specify the profile parameter EDPSIZE as described in the Natural Parameter Reference documentation.
- Under Natural Version 4.1.2, the Natural Complete/SMARTS Interface must be installed (product code NCF).

Prerequisites/Restrictions for NDV CICS Adapter

The Natural Development Server must have been installed on OS/390 or under SMARTS on VSE/ESA.

In addition, the following prerequisites and restrictions apply to the Natural Development Server CICS Adapter:

- CICS TS 1.3 or above.
- CICS TCP/IP and the CICS listener must be enabled. Refer to *CICS TCP/IP Socket Interface Guide*.
- Natural Version 4.1.2 and Natural CICS Interface Version 4.1.2 for Mainframes must be installed.
- Natural Development Server Version 2.1 must be installed on OS/390 in batch mode or under SMARTS on VSE/ESA.
- Natural must not be used with the Natural profile parameter ADAMODE set to 0, because this would cause an excessive number of Adabas user queue elements (UQE) per Natural session.

Installing the Natural Development Server

This document describes how to install a Natural Development Server (Product code: NDV) under the runtime environment SMARTS on VSE/ESA.

The following topics are covered:

- Prerequisites
- Content of the Development Server Distribution Tape
- Installation Procedure

Prerequisites

For details, refer to the section Prerequisites.

Content of the Development Server Distribution Tape

The installation tape contains the datasets listed in the table below. The sequence of the datasets and the number of library blocks needed are shown in the **Report of Tape Creation** that accompanies the installation tape.

| Dataset Name | Contents |
|-----------------|--|
| APS nnn .LIBR | Contains the load modules of the SMARTS Server. |
| NDV nnn .LIBR | Contains the load modules of the development server. See Natural Development Server on Mainframes. |
| NDV nnn .LIBJ | Contains Installation Job Control for customers who install without using System Maintenance Aid. |
| NDV nnn .DE11 | Contains instructions for deleting NDV Version 1.1 modules not used in NDV Version 2.1. |
| NDV nnn .INPL | Contains the transaction processor. See Natural Development Server on Mainframes. |
| NDV nnn .ERRN | Contains the error messages of the transaction processor. |
| NDV nnn .SYSF | Contains the FDT of the Development Server File (the layout is identical with PRD nnn .SYSF provided with a Predict version as specified under Natural and Other Software AG Products in the current Natural Release Notes). |
| NAT31601.LIBR | NDV requires the module NATGWSTG in the Natural load library. Since this module has not been delivered with Natural Version 3.1.6 for Mainframes, this correction library containing this module is provided. Future system maintenance (SM) levels and versions of Natural will already include the module NATGWSTG. |

The notation nnn in dataset names represents the version number of the product.

Installation Procedure

To install the Natural Development Server in the SMARTS environment, perform the following steps:

Copying the Tape Contents to Disk

If you are using System Maintenance Aid (SMA), refer to the SMA documentation (included on the current edition of the Natural documentation CD).

If you are **not** using SMA, follow the instructions below.

This section explains how to:

- Copy data set COPYTAPE.JOB from tape to library.
- Modify this member to conform with your local naming conventions.

The JCL in this member is then used to copy all data sets from tape to disk.

If the datasets for more than one product are delivered on the tape, the member COPYTAPE.JOB contains the JCL to unload the datasets for all delivered products from the tape to your disk, except the datasets that you can directly install from tape, for example, Natural INPL objects.

After that, you will have to perform the individual install procedure for each component.

Step 1 - Copy data set COPYTAPE.JOB from tape to disk

The data set COPYTAPE.JOB (file 5) contains the JCL to unload all other existing data sets from tape to disk. To unload COPYTAPE.JOB, use the following sample JCL:

```
* $$ JOB JNM=LIBRCAT,CLASS=0,                                     +
* $$ DISP=D,LDEST=(*,UID),SYSID=1
* $$ LST CLASS=A,DISP=D
// JOB LIBRCAT
* *****
*       CATALOG COPYTAPE.JOB TO LIBRARY
* *****
// ASSGN SYS004,NNN                                             <----- tape address
// MTC REW,SYS004
// MTC FSF,SYS004,4
ASSGN SYSIPT,SYS004
// TLBL IJSYSIN,'COPYTAPE.JOB'
// EXEC LIBR,PARM='MSHP; ACC S=lib.sublib'                       <----- for catalog
/*
// MTC REW,SYS004
ASSGN SYSIPT,FEC
/*
/&
* $$ EOJ
```

Where:

NNN is the tape address

lib.sublib is the library and sublibrary of the catalog

Step 2 - Modify COPYTAPE.JOB

Modify COPYTAPE.JOB to conform with your local naming conventions and set the disk space parameters before submitting this job:

Step 3 - Submit COPYTAPE.JOB

Submit COPYTAPE.JOB to unload all other data sets from the tape to your disk.

Step 4: Create a Development Server Configuration File

(Job I009 / Step 8410)

Catalogs the configuration file of the development server. For a description of the parameters, refer to Development Server Configuration.

The following parameters of the configuration file have to be defined. For the other parameters, the default values may be used:

| | |
|---------------|---|
| FRONTEND_NAME | Specify the name of the NDV server front-end module you generate in Step 13 . |
| PORT_NUMBER | Specify the TCP/IP port number under which the server can be connected. |

Step 5: Create a SMARTS Sysparms file

(Job I009, Step 8420)

Catalogs the member SYSPARMS for the SMARTS configuration file.

For detailed information on the SMARTS configuration file, refer to the SMARTS documentation, Configuration of the SMARTS Environment.

Step 6: Load FDIC system file

(Job I050, Step 8403)

If you do not use Predict at all or if you have not yet migrated to a Predict version specified under Natural and Other Software AG Products in the current Natural Release Notes, create the development server file, using the dataset NDV nnn .SYSF.

The layout of the Development Server File corresponds to the layout of the Predict Version 4.2 or above dictionary file.

Note: If you have a Predict version installed as specified under Natural and Other Software AG Products in the current Natural Release Notes, you can ignore this step.

Step 7: Assemble and link ADALNK

(Job I055, Step 8401)

The server environment requires a reentrant ADALNK module.

Link ADALNK using the ADALNKR module.

Step 8: Assemble and catalog for the Natural NDV interface

(Job I055, Step 8410)

If you install the Natural Development Server under Natural Version 3.1.6:

- Job I055, Step 8410, assemble and catalog the NCFNDVPM module.

If you install the Natural Development Server under Natural Version 4.1.1 or above:

- Job I055, Step 8415, assemble and catalog the NCFNDVPM module.

Step 9: Create the NDV server front-end module

(Job I060, Steps 8410, 8420/8425, 8430, 8440, 8450, 8460, 8470)

- Job I060, Step 8410, assemble and catalog the NDV NATPARM module.
- Job I060, Step 8420 link the NDV front-end.
If you install NDV under Natural Version 3.1.6.
- Job I060, Step 8425 link the NDV front-end
If you install NDV under Natural Version 4.1.1 or above.
- Job I060, Step 8430 link the NDV server.
- Job I060, Step 8440 link the NDV NATMONI module.
- Job I060, Step 8450 link the NDV NATMOPI module.
- Job I060, Step 8460 link the NDV NATSOCK module.
- Job I060, Step 8470 link the Natural Buffer Pool for NDV.

Step 10: Link the Natural nucleus

(Job I060, Step 0105)

- Job I060, Step 0105, link the Natural Nucleus with the module NATGWSTG.

Step 11: Delete Old Objects

(Job I061, Step 8440)

This step is required if NDV Version 1.1 was installed before. For a first time installation of NDV, skip this step.

Use the INPL command to delete the objects in dataset NDV nnn .DE11 from your Natural system file (FNAT). These modules are not used by the current NDV version.

Note that the parameter FDIC must have been set to point to your Natural Development Server File.

Step 12: Load Natural objects, error messages and samples for NDV

(Job I061, Steps 8450,8451,8452)

During NDV INPL, the assigned FDIC/FSEC file is initialized with NDV-specific information.

- Load objects from dataset NDV nnn .INPL onto your Natural system file (FNAT), using the INPL command. The parameter FDIC must have been set to point to your development server file.
- Load the error messages from dataset NDV nnn .ERRN using ERRLODUS.
- To use the tutorial (see First Steps with Natural Single Point of Development), load the sample programs from dataset NDV nnn .EXPL to your Natural system file.

Step 13: Copy DDMs and processing rules to FDIC

If you use a Predict Version 4.2 system file FDIC as development server file (FDIC), ignore this step.

If a Predict version as specified under Natural and Other Software AG Products in the current Natural Release Notes has not been installed or if you do not use a Predict Version 4.2 system file FDIC as Development Server File (FDIC), you have to copy the existing DDMs and processing rules to the Development Server File (FDIC),

using the copy function of the Natural utility SYSMAIN.

Step 14: Extend your SMARTS startup job by NDV-specific definitions

(Job I200, Step 8415)

Described in the section Development Server Configuration.

VSE Sample:

```
* $$ JOB JNM=NDVSRV,CLASS=C,DISP=L,LDEST=(,UID)
* $$ LST CLASS=A,DISP=H
// JOB NDVSRV --- NDV SERVER STARTUP ---
// OPTION PARTDUMP,NOSYSDMP,LOG
/* NDV server datasets -----
// DLBL NDVSRVT,'/SAGLIB/NDVCNFG/NDVSRV.TRC' writes trace to SAGLIB.NDVCNFG member NDVSRV.TRC
// DLBL NDVSRVC,'/SAGLIB/NDVCNFG/NDVSRV.P' location of NDV configuration file
// DLBL NDVSRVE,'SYSLST' NDV error output directed to job output
// DLBL NDVSRVO,'SYSLST'
// DLBL SYSPARM,'/SAGLIB/NDVCNFG/MSGQ.DMY' location of NDV Dummy file
// DLBL STDOUT,'CONSOLE' STDOUT directed to VSE console
// DLBL STDERR,'CONSOLE'
/* Libdef's -----
/*
// LIBDEF PHASE,SEARCH=(SAGLIB.NDVCNFG, +
SAGLIB.APS262, +
SAGLIB.ADA713)
/* *****
// UPSI 00000000
// EXEC TLINSP,SIZE=AUTO
* $$ SLI MEM=RJANPARM.P,S=SAGLIB.APS262
* $$ SLI MEM=PXANCONF.P,S=SAGLIB.APS262
* $$ SLI MEM=SYSPRM.P,S=SAGLIB.NDVCNFG NDV specific SMARTS configuration file
/*
// EXEC LISTLOG
```

Step 15: NDV Clients must be defined to Natural Security

If Natural Security (NSC) is installed:

- The NDV initial user ID (default ID is STARGATE) must be defined in Natural Security with a valid default library. Refer also to NDV configuration parameter INITIAL_USERID in the section Development Server Configuration. Alternatively, you can define the Natural profile parameter AUTO=OFF (automatic logon) for NDV.
- Each client user ID must be defined in Natural Security.

If the NDV initial user ID is not defined, the NDV server initialization aborts with a NAT0856 error message.

If an NDV client is not defined, the map environment returns an NSC error.

If you logon to the server from an NDV client, make sure that the user who is defined in Natural Security has a default library or a private library defined. Otherwise, the error message NAT0815 will occur.

Configuring the Natural Development Server

This document describes how to configure a Natural Development Server for SMARTS on VSE/ESA.

The following topics are covered:

- Configuration Requirements
 - Configuration File
 - Configuration Parameters
 - Server Datasets
-

Configuration Requirements

The Natural Development Server requires following SMARTS SYSPRM parameters:

| Parameter | Definition | |
|--------------|--|--|
| RESIDENTPAGE | The following members must be defined in the SMARTS resident area: NATRDEVS, NATSOCK, NATMONI, Natural front end (NCFNUC) and Natural nucleus (if you run using a split nucleus). | |
| SERVER | The following SERVER definitions are required for the Natural Development Server: | |
| | SERVER=(OPERATOR , TLINOPER , TLSPOPER) | The Operator Communications Server. |
| | SERVER=(POSIX , PAENKERN) | The POSIX Server. |
| | SERVER=(NATBPS31 , NCFBPS31 , 1 , 2048 , 2 , 512 , 4 , 1024) | The Natural local buffer pool. The module NCFBPS31 is delivered with Natural Version 3.1 for Mainframes. |
| CDI_DRIVER | CDI_DRIVER=(' TCPIP , PAACSOCK , MINQ=10 , MAXQ=20 ') | The SMARTS TCPIP Socket Driver for Connectivity Systems TCP/IP stack on VSE. MINQ/MAXQ defines the number of TcpIP listener tasks. |
| THSIZEABOVE | THSIZEABOVE=1024 | The storage above 16 MB that is available for each Natural Development Server subtask. This size must be large enough to keep the Natural tread, heap and stack of the Natural Development Server subtasks. If the Natural Development Server initialization fails with "NCF0003 Unable to allocate Natural thread", this parameter must be increased. |
| ADASVC | ADASVC= <i>nnn</i> | The Adabas SVC number of your Adabas installation. |

You can set the SMARTS SYSPRM parameters in the member RJANPARM.P in the SMARTS library, or you create a new SYSPARM member (e.g. in the Natural Development Server library). You have to concatenate this member with the RJANPARM.P definition in your SMARTS startup job.

SYSPRM Example for the Natural Development Server:

```

* ----- ADABAS PARMS -----*
ADACALLS=20 CALLS BEFORE ROLL
ADASVC=47 ADABAS SVC NUMBER
* ----- BUFFERPOOL PARMS -----*
BUFFERPOOL=(064,030,20,ANY)
BUFFERPOOL=(128,064,64,ANY)
BUFFERPOOL=(256,010,10,ANY)
BUFFERPOOL=(512,032,10,ANY)
BUFFERPOOL=(1K,032,32,ANY)
BUFFERPOOL=(6K,005,02,ANY)
BUFFERPOOL=(8K,016,16,ANY)
* ----- ROLLING PARMS -----*
ROLL-BUFFERPOOL=(048K,04,04,DS) ESA DATA SPACE
ROLL-BUFFERPOOL=(064K,04,04,DS) ESA DATA SPACE
ROLL-BUFFERPOOL=(128K,04,04,DS) ESA DATA SPACE
ROLL-BUFFERPOOL=(256K,04,04,DS) ESA DATA SPACE
ROLL-BUFFERPOOL=(800K,02,02,DS) ESA DATA SPACE
*
* ----- NDV Server to launch at startup -----*
* STARTUPPGM='NATRDEVS NDVS1
*
*
TASK-GROUP=(DEFAULT,6)
THREAD-GROUP=(DEFAULT,(DEFAULT,252,06,15,28,N))
*
THSIZEABOVE=1024
*
SERVER=(NATBPS31,NCFBPS31,1,2048,2,512,4,1024)
*
CDI_DRIVER=('TCPIP,PAACSOCK,MINQ=10,MAXQ=20')
*
RESIDENTPAGE=NATRDEVS
RESIDENTPAGE=NDVNCF31
RESIDENTPAGE=NATNUC31
RESIDENTPAGE=NATSOCK
RESIDENTPAGE=NATMONI

```

Configuration File

A configuration file is allocated to the DD-name `<serverid>C` (e.g. NDVS1C) or STGCONFIG alternatively.

The configuration file is a text file located on a dataset or or on a librarian member under VSE. It contains the server configuration parameters in form of a `keyword=value` syntax.

Configuration Parameters

The following configuration parameters are available:

```

DBG_CODEPAGE | FRONTEND_PARAMETER | INITIAL_USERID | FRONTEND_NAME |
TRACE_LEVEL | SESSION_PARAMETER | DEFAULT_PROFILE | HOST_NAME | PORT_NUMBER

```

DBG_CODEPAGE

This optional configuration parameter specifies which translation table is used by the remote debugger. By default, the remote debugger uses the code page IBM-1047 contrary to NDV which uses TABA1/2.

Possible values:

| | |
|------|---|
| USER | Use the Natural translation tables TABA1/2. |
|------|---|

Default Value none**Example** `DBG_CODEPAGE=USER`**FRONTEND_PARAMETER**

This optional configuration parameter contains additional Natural front-end parameters as specified in the Startup Parameter Area.

You can define multiple parameters. Each parameter specification is a pair of 8-character strings, the first containing the parameter keyword and the second the parameter value.

For further information, see the *Natural Operations for Mainframe documentation*, Natural in Batch Mode.

Default Value none**Example** `FRONTEND_PARAMETER="MSGCLASSX"`

The setting in the example specifies that the default output class for CMPRINT is "X".

INITIAL_USERID

At server initialization, the Natural Development Server creates a temporary Natural session to obtain the properties of the installed Natural environment.

This configuration parameter specifies the user ID to be used for this Natural session.

The specified value must not exceed 8 characters, otherwise it is truncated.

Default Value STARGATE**Example** `INITIAL_USERID=NDVINITU`**FRONTEND_NAME**

This configuration parameter specifies the name of the Natural front-end to be used to start a Natural session. The front-end resides on a PDS member.

Default Value none**Example** `FRONTEND_NAME=NAT315SV`

TRACE_LEVEL

See Trace Level details in the section Natural Development Server on Mainframe.

Default Value 0

Example TRACE_LEVEL=0x00000011

The setting in the example switches on Bits 31 and 27.

DEFAULT_PROFILE

This optional configuration parameter defines a default profile.

Specifying a parameter string in the Map Environment window of Natural Studio overwrites this default profile.

Default Value none

Example DEFAULT_PROFILE=RDEVS,10,930

The setting in the example defines that, if no parameters are defined in the Map Environment window, the session is started with the parameter PROFILE=(RDEVS10,930).

SESSION_PARAMETER

This optional configuration parameter defines session parameters that precede the parameter string either specified in the Map Environment window or defined by default by the configuration parameter DEFAULT_PROFILE.

Default Value none

Example SESSION_PARAMETER=FNAT=(10,930)

The setting in the example defines that every session on this Natural Development Server is started with the session parameter FNAT=(10,930) appended to the user-specified parameters or the definitions in DEFAULT_PROFILE.

HOST_NAME

This optional configuration parameter is necessary only if the server host supports multiple TCP/IP stacks.

If HOST_NAME is specified, the server listens on the particular stack specified by HOST_NAME, otherwise the server listens on all stacks.

Default Value none

Example HOST_NAME=node1

or

HOST_NAME=157.189.160.55

PORT_NUMBER

This configuration parameter defines the TCP/IP port number under which the server can be connected.

Default Value none

Example PORT_NUMBER=3140

Configuration File Example:

```
# This is a comment
SESSION_PARAMETER=profile=(stgqa,10,930) fuser=(10,32)
DEFAULT_PROFILE=DEFPROF
THREAD_NUMBER=2
THREAD_SIZE=700
FRONTEND_NAME=NATOS31L      # and another comment
PORT_NUMBER=4711
```

Server Datasets

The Natural Development Server requires the following datasets:

| | |
|----------|--|
| STGCONFG | Defines the server configuration file. |
| STGTRACE | The server trace output. |
| STGSTDO | The stdo dataset. |
| STGSTDE | The stde error output. |

Alternately, you can qualify each dataset name by the server ID. This is necessary if you would like to start different Natural Development Servers under a single SMARTS address space.

| | |
|--------|---|
| NDVS1C | Defines the server configuration file for the server NDVS1. |
| NDVS1T | The server trace output for the server NDVS1. |
| NDVS1O | The stdo dataset for the server NDVS1. |
| NDVS1E | The stde error output for the server NDVS1. |

Operating the Natural Development Server

This document describes how to operate a Natural Development Server in a SMARTS on VSE/ESA environment .

The following topics are covered:

- Starting the Natural Development Server
 - Terminating the Natural Development Server
 - Monitoring the Natural Development Server
 - Runtime Trace Facility
 - Trace Filter
-

Starting the Natural Development Server

A prerequisite is a running SMARTS address space that is configured to run NDV (see Development Server Installation under SMARTS on VSE/ESA).

Start the NDV server with the SMARTS console command `<msg-id> NATRDEVS <server-id>`, where *msg-id* is the message identifier assigned to the SMARTS partition and *server-id* is the name of your NDV server.

Example: 141 NATRDEVS NDVS1.

Note: If you qualify the NDV datasets by *server-id*, the *server-id* is restricted to a maximum length of 6 characters.

Alternatively you can automatically start NDV servers during SMARTS initialization using the SMARTS SYSPRM parameter STARTUPPGM. Specify `STARTUPPGM='NATRDEVS <server-id>'` within the SMARTS SYSPRM file.

Example: STARTUPPGM='NATRDEVS NDVS1'

Terminating the Natural Development Server

The Development Server can be terminated from within the Monitor Client NATMOPI (described in the Natural Operations for Mainframes documentation).

Monitoring the Natural Development Server

To enable the administrator to monitor the status of the Natural Development Server, a monitor task is provided which is initialized automatically at server startup. Using the monitor commands described below, the administrator can control the server activities, cancel particular user sessions, terminate the entire server, etc.

Monitor Communication

To communicate with the monitor, you can use the monitor client NATMOPI (described in the Natural for Mainframes Operations documentation).

Monitor Commands

The Natural Development Server supports the following monitor commands:

| Monitor Command | Action |
|---------------------------|---|
| ping | Verifies whether the server is active. The server responds and sends the string "I'm still up". |
| terminate | Terminates the server. |
| abort | Terminates the server immediately without releasing any resources. |
| set configvariable value | With the set command, you can modify server configuration settings. For example, to modify TRACE_LEVEL: <code>set TRACE_LEVEL 0x00000012</code> |
| list sessions | Returns a list of active Natural sessions within the server. For each session, the server returns information about the user who owns the session, the session initialization time, the last activity time and an internal session identifier (session-id). |
| cancel session session-id | Cancels a particular Natural session within the Natural Development Server. To obtain the session-id, use the monitor command list sessions . |
| help | Returns help information about the monitor commands supported. |

Runtime Trace Facility

For debugging purposes, the server code has a built-in trace which can be switched on, if desired.

The following topics are covered:

- Trace Medium
- Trace Configuration
- Trace Level

Trace Medium

A remote development server writes its runtime trace to a dataset allocated to the DD name STGTRACE or, alternatively, to *<server-id>T*.

The trace file is allocated (overwritten) at server initialization.

Trace Configuration

The trace is configured by a trace level which defines the detail of the trace. Once a trace is switched on, it can be restricted to particular clients or client requests by specifying a trace filter.

Every Natural session is equipped with a 32-bit trace status word (TSW) which defines the trace level for that session. The value of the TSW is set by a server configuration parameter. A value of zero means that the trace is switched off.

Trace Level

Each bit of the TSW is responsible for certain trace information. Starting with the rightmost bit:

| | |
|-----------|---|
| Bit 31 | Trace main events (server initialization/termination, client request/result). |
| Bit 30 | Detailed functions (session allocation, rollin/rollout calls, detailed request processing). |
| Bit 29 | Dump internal storage areas. |
| Bit 28 | Session directory access. |
| Bit 27 | Dump request/reply buffer EBCDIC. |
| Bit 26 | Dump request/reply buffer ASCII. |
| Bit 25-24 | Free. |
| Bit 23 | Request processing main events. |
| Bit 22 | Request processing detailed functions. |
| Bit 21 | Remote debugger main events. |
| Bit 20 | Remote debugger detailed functions. |
| Bit 19-16 | Free. |
| Bit 15 | Trace error situations only. |
| Bit 14 | Apply trace filter definitions. |
| Bit 13-08 | Free. |
| Bit 07-01 | Free. |
| Bit 00 | Reserved for trace-level extension. |

Trace Filter

In order to reduce the volume of the server trace output, it is possible to restrict the trace by a logical filter.

- The filter can be set with the configuration parameter TRACE_FILTER.
- The filter may consist of multiple keyword=*filtervalue* assignments separated by spaces.

The filter keyword is:

| | |
|--------|---|
| Client | Filters the trace output by specific clients. |
|--------|---|

The following rules apply:

- If a keyword is defined multiple times, the values are cumulated.
- The value must be enclosed in braces and can be a list of filter values separated by spaces.
- The values are not case sensitive and asterisk notation is possible.

Example:

```
TRACE_FILTER="Client=(KSP P*)"
```

Each request of the userid KSP and each request of the userids prefixed by a P are traced.

Natural Development Server Frequently Asked Questions

This document contains frequently asked questions concerning the Natural Development Server (NDV) under SMARTS on VSE/ESA.

The following topics are covered:

- Natural Development Server starts and terminates immediately
- Which dataset should I analyze to get error information?
- Trace output shows "Cannot load Natural front-end "
- Trace output shows "Transport initialization failed" "EDC8115I address already in use"
- Trace output shows "Error at: Template runtime connect"
- Definitions required in Natural Security
- I do not get a NAT0954 even if I specify DU=OFF
- Map Environment fails with a NAT3048
- Map Environment fails with Stub RC nn
- Special characters are not translated correctly
- How do I find out which hexadecimal value must be specified for TABA1/TABA2?
- The modifications of TABA1/TABA2 do not apply to sources listed in the remote debugger?
- Accessing work files
- I have problems when accessing DB2
- Are there any Natural profile parameter settings required for NDV?
- Sporadically I get a NAT7660 with socket code 0
- NAT9915 GETMAIN for thread storage failed

Natural Development Server starts and terminates immediately

At server initialization, the Natural Development Server

- allocates central control blocks,
- opens the datasets STGTRACE, STGSTDO, STGSTDE, STGCONFIG,
- obtains the configuration file,
- loads the Natural front-end,
- initializes the first Natural session and
- launches the TCP/IP listener task

If one of these steps fails, the server cannot continue and terminates immediately.

Analyze the trace output (STGTRACE) or the error output (STGSTDE) to find out the problem.

Which dataset should I analyze to get error information?

| | |
|----------|--|
| STGSTTE | <p>Contains only error output. Each record consists of 2-4 lines depending on whether it is a Natural error, a system error or an NDV stub error.</p> <p>Natural Error</p> <ol style="list-style-type: none"> 1. DayOfMonth Time TaskId UserId 2. TaskId NDV Error: error classification 3. Natural FrontEnd error or NATURAL runtime error 4. Natural error text <p>System Error</p> <ol style="list-style-type: none"> 1. DayOfMonth Time TaskId UserId 2. TaskId NDV Error: error classification 3. TaskId Sys Error: System error text <p>NDV stub error</p> <ol style="list-style-type: none"> 1. DayOfMonth Time TaskId UserId 2. TaskId NDV Error: error classification |
| STGTRACE | <p>Contains NDV trace information and error information. Each trace record contains</p> <p>DayOfMonth Time TaskId Trace information text</p> <p>The string PrintError in the Trace information text prefixes errors.</p> |
| STGSTO | Content of the configuration file allocated to STGCONFIG. |
| SYSOUT | Messages from LE runtime system. |

Trace output shows "Cannot load Natural front-end "

The Natural front-end specified by the NDV configuration parameter FRONTEND_NAME was not found in the load library concatenation.

Trace output shows "Transport initialization failed" "EDC8115I address already in use"

The TCP/IP port number specified by the NDV configuration parameter PORT_NUMBER is already in use by another process.

Trace output shows "Error at: Template runtime connect"

When a Natural Development Server initializes, it starts a Natural session using the session parameter defined by the NDV configuration parameter SESSION_PARAMETER. The profile definition of the NDV configuration parameter DEFAULT_PROFILE is appended.

If the initialization of the template session fails, the server terminates immediately. The original error can be found below the message "Error at:Template runtime connect".

Typical error situations could be:

- No Natural buffer pool defined
- Natural system file not accessible
- Profile parameter ITERM=ON
- NDV initial user ID not defined

Definitions required in Natural Security

- Each client must be defined in Natural Security if the Transition Period Logon flag in NSC is set to NO. Otherwise, your Map Environment fails with a NAT0873.
- You must define an NDV initial user ID (default ID is STARGATE) unless you run with profile parameter AUTO=OFF.
- Each user must have either a default library or a private library. Otherwise your Map Environment fails with

- a NAT1699.
- You must not specify a startup program that executes an I/O statement or stacks a LOGON, LOGOFF or RETURN command, because the program is executed whenever you change the focus to that library within the tree view.
- If you add a new user, you must specify a password for this user. Otherwise, the Map Environment fails with a NAT0838.

I do not get a NAT0954 even if I specify DU=OFF

The IBM Language Environment (LE) runtime option TRAP must be set to TRAP(ON,NOSPIE).

Map Environment fails with a NAT3048

Specify session parameter ETID=' '. If you have Natural Security, clear the ETID definition for that user.

Map Environment fails with Stub RC nn

Stub return codes are raised by the NDV front-end stub if it detects a logical processing error when dispatching the NDV request. The NDV trace output contains detailed information about the reason for the error.

The following stub return codes are possible:

| | |
|---|---|
| 1 | <p>Error during session reconnect (for future use).</p> |
| 2 | <p>Cannot create new session directory entry or subtask. If Natural Studio executes a Map Environment, the Natural Development Server allocates an entry in its session directory and creates a new subtask. If one of these actions fails, the Stub RC 2 is raised.</p> <p>Reason:</p> <ul style="list-style-type: none"> ● Region size (virtual storage below 16 MB) for the Natural Development Server is too small, ● Number of subtasks exceeds the limit specified by the OS/390 Unix System Services parameter MAXTHREADS. <p>Action: Increase region size or MAXTHREADS, or distribute the clients to several Natural Development Server. To save memory below 16 MB, you can also specify the ANY option of the LE parameter STACK (refer to Useful_LE_runtime options).</p> <p>The number of active tasks can be displayed using the OS/390 system command D OMVS ,PID=<i>process-id</i> (where <i>process-id</i> is the process id of the Natural Development Server).</p> <p>The value of MAXTHREADS can be displayed with D OMVS ,OPTIONS.</p> |
| 3 | <p>Cannot initialize new session. This error occurs if a storage allocation for internal NDV control buffers fails due to a lack of virtual memory above 16 MB.</p> <p>Reason: Virtual memory above 16 MB too small.</p> <p>Action: Increase the virtual memory above 16 MB, decrease the number of physical storage threads, configure NDV to use the Natural roll server, or distribute the clients to several Natural Development Servers.</p> |

| | |
|----|---|
| 4 | <p>Session execution failed. Internal error. The Natural Studio uses an invalid session identifier to process a request.</p> <p>Reason:</p> <ul style="list-style-type: none"> ● On Map Environment, the session ID already exists. ● The Natural session with specified ID is not initialized. <p>Action: Locate the defective session ID in the server trace file and cancel it using the monitor, or restart your Natural Studio.</p> |
| 5 | <p>I/O execution not allowed. In some situations, a Natural I/O is prohibited at the Natural Development Server.</p> <p>Reason:</p> <ul style="list-style-type: none"> ● I/O execution during LOGON request, ● I/O execution during execution of a transaction processor. <p>Action: Locate the I/O buffer in the server trace file to find out which I/O should be processed. Check for any startup program specified for the library you want to logon.</p> |
| 6 | Not applicable. |
| 7 | <p>Error during I/O execution. The Natural Development Server cannot finish a terminal I/O.</p> <p>Reason:</p> <ul style="list-style-type: none"> ● Virtual memory above 16 MB too small, ● I/O reply buffer send by Natural Studio is invalid. <p>Action: Increase the virtual memory above 16 MB. If the I/O reply buffer is invalid, contact Software AG support.</p> |
| 8 | Protocol element missing. Internal error, contact Software AG support. |
| 9 | <p>NDV not installed on Natural systemfile. Natural Development Server cannot execute the Natural module TRPRO located on library SYSLIB.</p> <p>Reason: The NDV modules are not loaded on the FNAT.</p> <p>Action: INPL the NDV modules.</p> |
| 10 | <p>LOGON command required. If you execute a program on the Natural Development Server that executes a LOGOFF (or a RETURN when no SETUP record is available), the logon library is undefined. In an online environment the Natural Security logon screen is displayed in this situation. Under NDV, the Natural session rejects all requests except a LOGON command. This applies only if Natural Security is installed. You can execute a LOGON command either by using the command line or by clicking on any library in your tree view.</p> |

Special characters are not translated correctly

The ASCII-EBCDIC translation for NDV uses the Natural translate tables TABA1/TABA2. These tables can be maintained at customer site. The translate tables can be modified as follows:

1. Modify source member NTTABA1/NTTABA2 on the Natural distribution library. Reassemble NATCONFIG and relink the Natural nucleus.
2. Specify the Natural session parameter TABA1/TABA2.

How do I find out which hexadecimal value must be specified for TABA1/TABA2?

Run the following program on your Natural for Windows locally.

```
#A(A1) = '{'  
WRITE A(EM=H)  
END
```

Output is **7B**.

Run the program on a mainframe (edit the program with the Natural mainframe editor).

Output is **75**, assuming that you use a German EBCDIC table. If you use a US EBCDIC table, the output will be C0.

Start your Natural Development Server session with TABA1=(75,7B) and TABA2=(7B,75).

The modifications of TABA1/TABA2 do not apply to sources listed in the remote debugger?

Specify the NDV configuration parameter `DBG_CODEPAGE=USER`.

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.

I have problems when accessing DB2

Ensure that your Natural Development Server is started with the configuration parameter `KEEP_TCB=YES`.

Are there any Natural profile parameter settings required for NDV?

The following Natural profile parameter values are required for NDV:

- `ETID=OFF` is required to allow multiple Natural sessions for each client.
- `DBCLOSE=ON` is required to remove database resources immediately after session termination rather than to keep them until they are removed due to a timeout.
- `ITERM=OFF` is required to continue with the Natural Development Server initialization even if session initialization errors occur.
- `AUTO=ON/OFF` has a different behavior under Natural Single Point of Development. In an online Natural environment, this parameter controls whether you are prompted for your user ID and password or if your user ID is treated to be a trusted user ID from the TP environment. With Natural Single Point of Development, you must always specify your user ID and password in the Map Environment dialog.

Sporadically I get a NAT7660 with socket code

The reason for that error is a queue overflow for incoming TCP/IP requests that results in an IPN214W error in the TCP/IP SYSLST output. The SMARTS SYSPARM parameter

`CDI_DRIVER=('TCP/IP', PAACSOCK, MINQ=nn, MAXQ=nn)` defines the minimum and/or maximum number of requests that can be queued by TCP/IP.

Increase the value of `MINQ` and ensure that `MAXQ` is greater than or equal to `MINQ`.

NAT9915 GETMAIN for thread storage failed

The Natural Front-End cannot allocate the Natural thread. Increase the SMARTS SYSPARM parameter THSIZEABOVE. The NDV configuration parameter THREAD_SIZE is obsolete under VSE.

Introducing the Natural Development Server CICS Adapter

This document describes the purpose and the functions of the Natural Development Server (NDV) CICS Adapter.

The following topics are covered:

- Purpose of the Natural Development Server CICS Adapter
 - Remote Development Functions
 - CICS Support
 - Product Interaction
-

Purpose of the Natural Development Server CICS Adapter

The Natural Development Server CICS Adapter is designed for a Natural Single Point of Development context where it enables the use of a Natural Development Server (NDV, running under OS/390 in batch mode or under SMARTS on VSE/ESA) within a CICS TP monitor environment.

See also

- Natural Single Point of Development
- Natural Development Server for OS/390
- Natural Development Server under SMARTS on VSE/ESA

Remote Development Functions

The Natural Development Server CICS Adapter enables you to execute a Natural Single Point of Development session within CICS.

In the Tools menu, Natural Studio offers you a function named "Map Environment". This function enables you to open a Natural session on a remote development server.

If you configure the remote development server for use in conjunction with the Natural Development Server CICS Adapter, this Natural session is not hosted by the remote development server, but it is dispatched remotely within a specified CICS region.

CICS Support

The CICS support is not implemented within the front-end stub NATRDEVS. For dispatching the Natural sessions in CICS, the development server continues to run in batch or under SMARTS. But it uses the remote front-end NATCSRFE that is delivered with the Natural Development Server to dispatch the Natural sessions in CICS. That is, depending on the installed front-end, a development server dispatches the sessions locally (NCFNUC for SMARTS or NATMVS for batch) or remote (NATCSRFE for CICS).

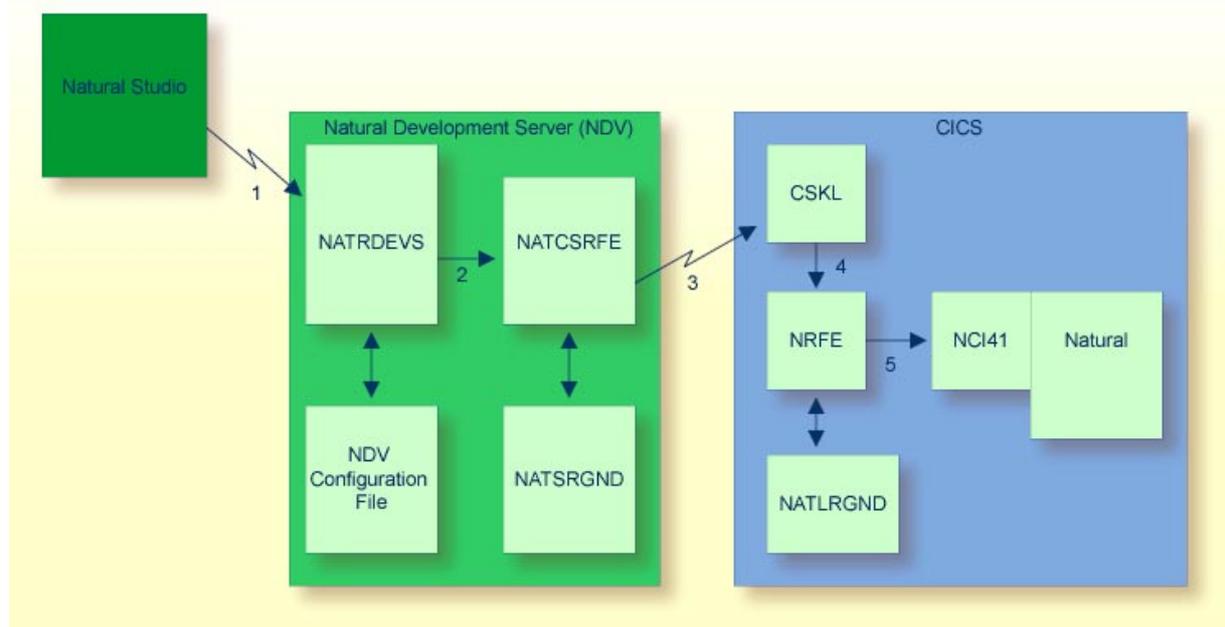
NATCSRFE in turn accepts the Natural request from NATRDEVS and transfers it to a configured CICS environment using the CICS Socket Interface. Within the CICS environment, a CICS Natural transaction is launched that processes the Natural request and returns the result. Thus it is not necessary to execute the entire development server under CICS. Only small working units (Natural requests such as "save source" or "get library list") are transferred to CICS for execution.

The Natural Development Server CICS Adapter comprises the following components:

| | |
|-------------------------------|--|
| NATCSRFE | The remote front-end called by the Natural Development Server to dispatch a Natural request. It is loaded into the development server address space. |
| NATCNRFE | The counterpart of NATCSRFE. NATCNRFE runs in the CICS address space. It is started by the IBM-provided standard listener of the CICS Socket Interface (refer also to <i>TCP/IP V3R1 for MVS: CICS TCP/IP Socket Interface Guide</i> and <i>TCP/IP for VSE/ESA VIR5 IBM Program Setup and Supplementary Information</i>). |
| NATSRGND/ NATLRGND | Transmits the NDV-relevant data between Natural Development Server and the Natural session running in CICS. NATSRGND must be loaded into the Natural Development Server address space and NATLRGND into the CICS address space. |

Product Interaction

The following figure illustrates the interaction between Natural Studio as a remote development client, the Natural Development Server and the CICS environment involved.



1. Natural Studio sends the remote development request to the Natural Development Server using the port number specified with the Natural Development Server configuration variable `PORT_NUMBER`.
2. The Natural Development Server dispatches the Natural session using the Natural front-end you have specified with the Natural Development Server configuration variable `FRONTEND_NAME`. Specify `NATCSRFE` in order to use the Natural Development Server CICS Adapter. For details, see the configuration parameter descriptions in the platform-specific document *Configuring the Natural Development Server*.
3. `NATCSRFE` transmits the request to the host/port specified with the Natural Development Server configuration variable `RFE_CICS_TA_HOST / RFE_CICS_TA_PORT`. You must configure the CICS-supplied standard listener `CSKL` to listen at this port.
4. `CSKL` launches the CICS transaction you have specified with the Natural Development Server configuration variable `RFE_CICS_TA_NAME` (`NRFE` in this example). This transaction must be defined to use the program `NATCNRFE`.

5. NATCNRFE finally dispatches the Natural session using the Natural CICS Front-End you have specified with the Natural Development Server configuration variable RFE_CICS_FE_NAME.

Installing the Natural Development Server CICS Adapter

This document describes how to install the CICS connection for a Natural Development Server (NDV) running under OS/390 in batch mode or under SMARTS on VSE/ESA.

The following topics are covered:

- Prerequisites
 - Installation Procedure
-

Prerequisites

For details, refer to the section Prerequisites.

Installation Procedure

To install the Natural Development Server CICS Adapter, perform the following steps:

Step 1: Customize CICS

(Job I005, Steps 8405, 8406, 8410, 8411)

The Natural Development Server load library must be defined in the CICS DFHRPL concatenation.

Customize the standard listener of the CICS socket interface using the CICS transaction EZAC.

Start the standard listener using the CICS transaction EZAO.

The following CICS resource definitions are required:

1. Define the CICS transaction for the remote front-end. This transaction name is an arbitrary name which must be defined in the NDV configuration parameter RFE_CICS_TA_NAME. This document uses the transaction name NRFE.
2. Define the programs NATCNRFE and NATRGNDV.

The sample JCL containing the following members defines all necessary CICS entries:

- NDV I005C
- NDV I009C
- NDV CONFC

Step 2: Customize the Development Server

In order to dispatch the NDV Natural sessions in CICS, you must adapt the configuration file of your development server running under OS/390 in batch mode or under SMARTS on VSE/ESA. For this purpose, two sample JCL members (NDV I009C and NDV CONFC) are available.

Refer to Configuring the Natural Development Server CICS Adapter and to Configuring the Natural Development Server.

Configuring the Natural Development Server CICS Adapter

This document describes how to configure the CICS connection for a Natural Development Server (NDV) running on OS/390 or under SMARTS on VSE/ESA.

The following topics are covered:

- Configuration File
 - Configuration of the NDV CICS Adapter
-

Configuration File

After the installation of the NDV CICS Adapter is complete, the configuration of the NDV CICS Adapter has to be done in the Natural Development Server configuration file of the corresponding Natural Development Server.

To enable the CICS Adapter, you have to specify the remote front-end module in the NDV configuration parameter `FRONTEND_NAME` (`FRONTEND_NAME=NATCSRFE`).

Configuration Parameters

The following CICS-relevant configuration parameters exist:

`RFE_CICS_TA_NAME` | `RFE_CICS_FE_NAME` | `RFE_CICS_TA_HOST` | `RFE_CICS_TA_PORT` |
`RFE_CICS_TA_INIT_TOUT` | `RFE_CICS_KEEP_TA` | `RFE_CICS_TRACE`

RFE_CICS_TA_NAME

This configuration parameter specifies the CICS transaction to be used for starting the remote front-end in CICS. This transaction must be defined in CICS and must refer to the program `NATCNRFE`. See also *Installing the NDV CICS Adapter*.

Default Value none

Example `RFE_CICS_TA_NAME = NRFE`

RFE_CICS_FE_NAME

This configuration parameter specifies the Natural CICS nucleus you have installed with the Natural Version 4.1 installation under CICS. This program must be defined in CICS.

Default Value none

Example `RFE_CICS_FE_NAME = NCI41NUC`

See also the *Natural Installation Guide for Mainframes*, *Installing the Natural CICS Interface*, *Customize CICS*.

RFE_CICS_TA_HOST

This configuration parameter specifies the TCP/IP address of the host the desired CICS is running. This parameter can be omitted if the development server and CICS are running on the same TCP/IP node.

Default Value The host address of the development server.

Example RFE_CICS_TA_HOST = node1 or RFE_CICS_TA_HOST = 157.189.160.55

RFE_CICS_TA_PORT

This configuration parameter specifies the TCP/IP port of the CICS supplied listener.

You can acquire this port number using the CICS supplied transaction EZAC. The CICS command EZAC DISPLAY LISTENER shows the definitions of the CICS standard listener.

Note:

This port number is not used in Natural Studio to map to a remote development server. This port number (and the RFE_CICS_TA_HOST definition) is used internally by the development server to communicate with the CICS region.

Default Value none

Example RFE_CICS_TA_PORT = 3010

RFE_CICS_TA_INIT_TOUT

If Natural Studio sends a request to a Natural Development Server that is configured to use the CICS remote front-end, the remote front-end launches a CICS transaction (NRFE) for processing the request. The CICS transaction in turn listens to the TCP/IP to receive the data from the development server required processing the request.

This configuration parameter specifies the timeout value (in seconds) a launched transaction waits until the expected request data arrives from the development server. If this timeout expires, the request aborts with a NAT9940 error.

Default Value 5

Example RFE_CICS_TA_INIT_TOUT = 20

Note: Do not define a value below 5.

RFE_CICS_KEEP_TA

For each request sent by the Natural Studio, NDV opens a TCP/IP connection to the CICS region and launches a CICS transaction (NRFE) for processing the request. With RFE_CICS_KEEP_TA=YES, the CICS transaction remains active for processing further requests of the same client. This saves the overhead for creating the TCP/IP connection and transaction initialization for successive requests, but consumes more resources within the CICS region due to waiting transactions.

The transaction wait time (for successive requests) is limited by RFE_CICS_TA_INIT_TOUT. That is, if the time slice between two successive requests exceeds the time specified by RFE_CICS_TA_INIT_TOUT, the CICS transaction and the TCP/IP connection is terminated independent of the RFE_CICS_KEEP_TA definition.

RFE_CICS_TA_INIT_TOUT=5 is a reasonable value to reuse transactions for multiple requests initiated by a single action in Natural Studio and to save CICS resources if Natural Studio waits for the next action of the user.

Default Value None

Example RFE_CICS_KEEP_TA = YES

RFE_CICS_TRACE

This configuration parameter specifies the trace level for the remote front-end.

The trace level is similar to the trace implemented for the development server. It is a bit string where each bit is responsible for a certain trace information:

| | |
|--------|---|
| Bit 31 | Trace main events (transaction initialization/termination, request processing). |
| Bit 30 | Detailed functions. |
| Bit 29 | Dump internal storage areas. |
| Bit 27 | Dump buffer header exchanged between development server and CICS. |
| Bit 26 | Dump entire buffer exchanged between development server and CICS. |
| Bit 25 | Dump the NDV relevant buffer only (Remote Gateway buffer). |
| Bit 23 | Trace error situations only. |
| Bit 07 | Activate trace in the development server region. |
| Bit 06 | Activate trace in the CICS region. |
| Bit 00 | Reserved for trace-level extension. |

The trace destination is the data set defined for STDOUT.

Default Value 0

Example RFE_CICS_TRACE = Dump main events and buffer header in the CICS region
0x02000011 (Bits 31 + 27 + 07)..

The following is a sample development server configuration file using the NDV CICS Adapter:

```
# the development server parameter
SESSION_PARAMETER= PROFILE=(NDV,10,930)
FRONTEND_NAME = NATCSRFE          # use the CICS Adapter front-end
PORT_NUMBER=4711                  # the port number used by Natural Studio

# the CICS Adapter parameter
RFE_CICS_TA_NAME = NRFE          # the CICS transaction for remote front-end
RFE_CICS_TA_PORT = 3010         # the port of the CICS listener
                                # no RFE_CICS_TA_HOST is defined. This requires
                                # that CICS runs on the same node as the
                                # development server
RFE_CICS_FE_NAME = NCI41NUC     # the name of the installed Natural CICS nucleus
RFE_CICS_TA_INIT_TOUT = 20     # transaction timeout is 20 seconds
```

Note:

The development server parameters `THREAD_NUMBER` and `THREAD_SIZE` are obsolete when the NDV CICS Adapter is used.

NDV CICS Adapter Frequently Asked Questions

This document contains frequently asked questions concerning the Natural Development Server CICS Adapter under VSE/ESA.

The following topics are covered:

- Under which CICS userid does the NDV transaction run within the CICS region?
 - I receive a NAT9940 (NAT9939) starting my NDV server.
-

Under which CICS userid does the NDV transaction run within the CICS region?

The NDV transaction (the NDV Natural session) runs under the CICS default userid specified in the CICS system initialization parameter DFLTUSER.

This is the same userid as your CICS standard listener (CSKL) uses.

I receive a NAT9940 (NAT9939) starting my NDV server.

The NAT9940 message in fact should be a NAT9939. This will be corrected with Patch Level 01 (NDV212PL01). The NAT9939 message indicates an error in the communication between the NDV server environment and the CICS environment. The general layout of the message is a text describing the error which may be followed by a condition code (CC), if applicable.

The most important NAT9939 errors are listed below. Many errors not listed here are internal errors.

- **ConfigError: missing or invalid.**
A mandatory configuration variable for the Natural Development Server CICS Adapter is not defined in the NDV configuration file.
- **Cannot bind Socket.**
The port specified with RFE_CICS_TA_PORT is not in a listen state on the node specified with RFE_CICS_TA_HOST. Probably the CICS TCP/IP standard listener is configured to use a different port or the listener is not running.
- **Timeout at connection establishment.**
The CICS transaction launched by the NDV server did not respond within the time specified in RFE_CICS_TA_INIT_TOUT. Examine CICS message log for potential messages regarding this transaction.
- **Partner closed connection.**
Unexpected abort of the connection by either of the partners (NDV server or CICS transaction). Examine CICS message log and NDV server trace for preceding error messages regarding this request.
- **Invalid reply on connection establishment.**
The CICS transaction launched by the NDV server did not initialize correctly. Examine CICS message log for potential messages regarding this transaction. Possible reason: The transaction defined with RFE_CICS_TA_NAME is not defined correctly within CICS.
- **Cannot load NDV Remote Gateway DLL.**
The remote gateway DLL NATSRGND/NATLRGND cannot be loaded within the NDV server/CICS region. Possible reason: Module cannot be found on load library concatenation or CICS PPT entry missing.
- **Cannot load NCI front-end.**
The Natural CICS front-end specified with RFE_CICS_FE_NAME cannot be loaded. Examine CICS message log error messages regarding this program. Possible reason: Module cannot be found on load library concatenation or CICS PPT entry missing.

