

Natural

Configuration Utility

Version 6.3.13 for UNIX

October 2012

This document applies to Natural Version 6.3.13 for UNIX.

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

Copyright © 1992-2012 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

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

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

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Document ID: NATUX-NNATCONFIG-6313-20121005

Table of Contents

Preface	v
1 Starting and Terminating the Configuration Utility	1
Starting the Configuration Utility	2
Terminating the Configuration Utility	3
Performing a Function while Starting the Configuration Utility	3
2 Working with the Configuration Utility	5
Creating a New Parameter File	6
Opening a Parameter File	6
Locating and Changing a Parameter	7
Finding a Parameter	8
Saving the Changes	8
Deleting a Parameter File	9
Exporting a Parameter File	10
Importing a Parameter File	10
3 Overview of Configuration File Parameters	13
Global Configuration File	14
Local Configuration File	23
4 Overview of Profile Parameters	27
Database Management	28
Natural Execution Configuration	30
Natural Development Environment	44
Product Configuration	45
Client/Server	46
5 Invoking Natural with an Alternative Parameter File	49

Preface

This documentation describes the Configuration Utility which is used to modify global and local configuration files (for administrator use only) and to create or modify parameter files.

This documentation is organized under the following headings:

Starting and Terminating the Configuration Utility	How to start and terminate the Configuration Utility.
Working with the Configuration Utility	How to locate and change parameters, find parameters, and save changes. How to open, create, delete, export and import parameter files.
Overview of Configuration File Parameters	Information on the parameters that can be set in the global and local configuration files.
Overview of Profile Parameters	Information on the profile parameters that can be set in the parameter files.
Invoking Natural with an Alternative Parameter File	How to invoke Natural with your own customized parameter file.

See also: *Profile Parameter Usage* in the *Operations* documentation and *Parameter Reference*.



Notes:

1. Parameter files are upward-compatible, but not downward-compatible. Once you have created a parameter file in a higher version of the Configuration Utility and you try to read it with a lower version, an error message will be displayed. To be able to read the file, **export** the file to the higher version and then **import** this file to the lower version of the Configuration Utility.
2. Parameter files are not compatible on different platforms. Therefore, it is not possible to copy them from one platform to another (for example, from Windows to UNIX or OpenVMS). If you want to use a parameter file on a different platform, **export** the file and then **import** this file on the target platform.

1 Starting and Terminating the Configuration Utility

- Starting the Configuration Utility 2
- Terminating the Configuration Utility 3
- Performing a Function while Starting the Configuration Utility 3

Starting the Configuration Utility

The way you start the Configuration Utility depends on how the system has been configured at your site. For most installations, you start it as described below.

▶ **To start the Configuration Utility**


- Enter the following command at the UNIX system prompt:

```
natparm
```

The **Natural Configuration Utility** screen appears.

```
09/05/2007          Natural Configuration Utility          User: (Config)
                   V 6.3.4                               File: NATPARAM
+-----+-----+-----+-----+
|File      Edit      Configuration      Search      |
+-----+-----+-----+-----+

Natural Parameter File Administration
```

 **Note:** The **Configuration** menu is only shown in the **Natural Configuration Utility** screen if you have been defined as configuration administrator in the local configuration file.

The name of the parameter file which is currently active is shown in the top right-hand corner of the screen.

When "(Admin)" or "(Config)" is shown in the top right-hand corner of the screen (instead of a user ID), you have been defined as an administrator in the local configuration file.

For further information, see *Administrator Assignments* in the local configuration file.

Terminating the Configuration Utility

When you terminate the Configuration Utility and you have not yet saved your changes, a window appears asking whether you want to save now.

▶ To terminate the Configuration Utility

- From the **File** menu, choose **Exit**.

Performing a Function while Starting the Configuration Utility

You can perform several functions automatically when starting the Configuration Utility.

The following functions can be specified after the `natparm` command at the UNIX system prompt.

Function	Description
<code>exit</code>	Terminates the Configuration Utility.
<code>import=parameter-file</code>	Imports the specified parameter file.
<code>export=parameter-file</code>	Exports the specified parameter file.
<code>parm=parameter-file</code>	Opens the specified parameter file.
<code>save</code>	Saves a parameter file.
<code>save=new-parameter-file-name</code>	Saves a parameter file under a different name.

Multiple functions can be specified one after the other (see the examples below).

Examples

- Invoke the Configuration Utility and open the specified parameter file so that it can be edited immediately:

```
natparm parm=parameter-file
```

- Invoke the Configuration Utility, open the specified parameter file, and save it as a new file with the specified name:

```
natparm parm=parameter-file save=new-parameter-file-name
```

- Invoke the Configuration Utility, load and export the specified parameter file and terminate the Configuration Utility:

```
natparm parm=parameter-file export=parameter-file exit
```

- Invoke the Configuration Utility, import the specified parameter file and save it under the name of the imported file:

```
natparm import=parameter-file save
```

2 Working with the Configuration Utility

▪ Creating a New Parameter File	6
▪ Opening a Parameter File	6
▪ Locating and Changing a Parameter	7
▪ Finding a Parameter	8
▪ Saving the Changes	8
▪ Deleting a Parameter File	9
▪ Exporting a Parameter File	10
▪ Importing a Parameter File	10

Creating a New Parameter File

You can create your own parameter files and customize them to your requirements. You can then invoke Natural with one of these parameter files. See *Invoking Natural with an Alternative Parameter File* for further information.

Initially, each parameter file that you create contains Natural's default settings.



Tip: It is recommended that you create your own parameter file instead of modifying the default parameter file NATPARM which is delivered with Natural.

When you are not defined as an administrator, you can only modify your own parameter file.

▶ To create a new parameter file

- 1 From the **File** menu, choose **New**.

A new, unnamed parameter file is opened (a name is not shown at the top of the screen). The parameters in this file are set to the default values. You can set all parameters as required.

- 2 Assign a name to the file by **saving** it.



Note: You can also create a new parameter file by **saving** an existing file under a new name (administrator only).

Opening a Parameter File

If you want to modify a different parameter file, you have to open it as described below.



Note: Only an administrator can modify all parameter files.

▶ To open a parameter file

- 1 From the **File** menu, choose **Open**.

A list of all available parameter files is shown.

- 2 Select the required parameter file and press ENTER.



Note: When you select the first entry in the list, which is <OPEN>, you can enter the name of the parameter file.

The name of the opened parameter file is shown in the top right-hand corner of the screen. You can now modify this parameter file.

Locating and Changing a Parameter

You can locate a parameter in different ways:

- If you know in which parameter group and category of the **Edit** menu the parameter can be found, you can locate the parameter as described below.
- If you do not know the parameter group and category, you can search for the parameter. See [Finding a Parameter](#).

▶ To locate and change a parameter in the current parameter file

- 1 From the **Edit** menu, choose a parameter group (for example, **Natural Execution Configuration**).
- 2 From the resulting window, choose a category (for example, **Buffer Sizes**).

A window containing parameters is shown. The names of the Natural profile parameters are shown in parenthesis, behind the label of the corresponding option.

- 3 Set the parameter to the required value. See [Overview of Profile Parameters](#) for further information.



Note: When an input field is selected, the allowed values (or PF keys) are shown at the bottom of the screen. For several parameters it is possible to press a PF key to invoke a selection list.

- 4 Save your changes as described in [Saving the Changes](#).

▶ To locate and change a parameter in a configuration file

- 1 From the **Configuration** menu, choose either **Global Configuration File** or **Local Configuration File**.
- 2 From the resulting window, choose a category (for example, **Printer Profiles**).

A window containing parameters is shown.

- 3 Set the parameter to the required value. See [Overview of Configuration File Parameters](#) for further information.
- 4 Save your changes as described in [Saving the Changes](#).

Finding a Parameter

If you do not know in which category a parameter can be found, you can search for it.

▶ To find a parameter

- 1 Select the **Search** menu and press `ENTER`.

The **Search Parameter** window appears.

- 2 Specify the parameter name in the window and press `ENTER`.



Note: You can also leave the **Search Parameter** window empty and press `ENTER`. In this case, another window appears from which you can select the required parameter. It is also possible to select a category (for example, the **Printer Profiles** category which belongs to the global configuration file).

A window containing the parameter is shown.

Saving the Changes

Parameter files and configuration files are saved in different ways.

When you are not an administrator, you can only save your own parameter files. In this case, it is not possible to save a parameter file under a different name.

▶ To save the changes for the current parameter file

- 1 From the **File** menu, choose **Save**.

A window appears asking whether you want to overwrite the existing parameter file.

- 2 Choose "YES" to save your changes.

▶ To save the changes for the global configuration file

- 1 From the **Configuration** menu, choose **Global Configuration File**.
- 2 From the resulting window, choose **Save to Global Configuration File**.

▶ To save the changes for the local configuration file

- 1 From the **Configuration** menu, choose **Local Configuration File**.

- From the resulting window, choose **Save to Local Configuration File**.

▶ **To save the current parameter file under a different name**

- From the **File** menu, choose **Save As**.

A message appears at the bottom of the screen, prompting you to enter a name.

- Enter the name for the new parameter file.

Deleting a Parameter File

You can delete an existing parameter file.

When you invoke Natural with a parameter file that has been deleted, make sure to adapt the corresponding shortcuts. See *Invoking Natural with an Alternative Parameter File* for further information.

▶ **To delete a parameter file**

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

The following window appears.

```
+----- Delete Parameter File from... -----+
| File Name: *                               |
| Path.....:                               |
| $PARM_PATH                               |
+-----+
```



Note: The path in the above window cannot be changed.

- Enter the name of the parameter file that is to be deleted in the **File Name** field and press ENTER.



Note: You can also leave the asterisk (*) in the **File Name** field and press ENTER. In this case, a window appears and you can select the parameter file from a list.

A windows appears asking whether you really want to delete the parameter file.

- Choose "YES" to confirm the deletion.

Exporting a Parameter File

When you export a parameter file, a text file is generated which contains the values of the current parameter file. Such a text file can be used on Windows, UNIX and OpenVMS platforms; to use it there, you have to **import** it.

▶ To export the current parameter file

- 1 From the **File** menu, choose **Export**.

The following window appears.

```
+----- Export Parameter File to ... -----+
| File Name: NATPARAM                          |
| Path.....:                                  |
| $TMP_PATH                                    |
+-----+
```

The name of the current parameter file is proposed as the file name. You need not enter an extension. The extension "LST" will automatically be added to the file.

- 2 Optional. Specify another file name. You can also specify the path to another directory.
- 3 Press ENTER.

If you have not specified another directory, the file is stored in the Natural directory which has been defined for the temporary files. See also *Installation Assignments*.

Importing a Parameter File

You can import a parameter file which has previously been **exported**. The import generates a binary parameter file from the text file with the extension "LST" which has been generated during the export.

▶ To import a parameter file

- 1 From the **File** menu, choose **Import**.

The following window appears.


```

+----- Import Parameter File from... -----+
| File Name: *                                |
| Path.....:                                |
| $TMP_PATH                                  |
+-----+

```

- 2 Enter the name of the parameter file that is to be imported in the **File Name** field. If required, enter the path to this file. Press ENTER. You need not enter the extension.



Note: You can also leave the asterisk (*) in the **File Name** field and press ENTER. In this case, a window appears and you can select the parameter file from a list.

When a file with the same name does not yet exist, the parameter file is imported.

When a file with the same name already exists, you are asked whether you want to overwrite the existing file.

3 Overview of Configuration File Parameters

- Global Configuration File 14
- Local Configuration File 23

This chapter provides information on the parameters that can be set in the global and local configuration files.

```
+-----+
|File      Edit      Configuration      Search      |
+-----+-----+-----+
| Global Configuration File |
| Local Configuration File |
+-----+-----+-----+
```

The configuration files can only be modified by an administrator. They should only be changed with the Configuration Utility.

After a new Natural installation, the default administrator names are SAG and DBA. For these administrators, the **Configuration** menu is available. The administrator SAG or DBA can then add further administrators to the local configuration file as defined in the section [Administrator Assignments](#).

Global Configuration File

This parameter group contains the following categories:

- Database Management System Assignments
- Dictionary Server Assignments
- HTTP Client Parameter Settings
- Natural Version-independent Operating System Files
- NDV Environment Assignments
- Printer Profiles
- System Files

Database Management System Assignments

The parameters which apply to the database management system assignments are grouped under the following headings:

- DBMS Assignments
- DBMS Tracing
- SQL Assignments

- XML Assignments

DBMS Assignments

A table is shown which is used to define the database ID with which the database is mapped in the DDM.

When DBMS assignments have already been defined, they are shown in the table.

DBID

The database ID.



Note: If you are using DDMs where the database ID is set to "0", a separate entry where the database ID is also set to "0" must exist, with the corresponding database type specified. This entry is used at compilation time in order to generate the corresponding database statements into the generated program (GP).

Type

Since the types of all databases which are to be accessed by Natural must be defined in the global configuration file, specify one of the following values for each database ID:

Value	Description
ADA	Adabas database server (this is the default).
ADA2	Adabas database server. This database type is provided for use with Adabas as of version 6 on Open Systems and Adabas as of version 8 on mainframes. In particular, it supports Adabas LA fields, Adabas large object fields and Natural view sizes greater than 64 KB. See also <i>Adabas Database Management Interfaces ADA and ADA2</i> in the <i>Programming Guide</i> .
SQL	Any SQL database that can be accessed using Entire Access, which is Software AG's common interface to various SQL database systems.
XML	Tamino database server.



Important: You must not define a database type for a DBID which has already been assigned to a system file in the global configuration file (see [System Files](#)). If you do so, an error message will be issued at Natural startup, indicating an inconsistency in the system file setting and an error when reading the database assignments.

At compilation time, the profile parameter `UDB` is not used to determine the database type. `UDB` is only used at runtime to determine the database type. The database types must be the same at compilation time and at runtime.

Exception: For compatibility reasons, it is possible to use database type ADA at compilation time and ADA2 at runtime.

Parameter

Applies only to databases of the following types:

Type	Description
SQL	If your database type is SQL, specify the string (DBMS parameter) that is to be used to establish the connection to the database system with which you want to work. See <i>Natural and Entire Access</i> in the <i>Database Management System Interfaces</i> documentation for further information on how to access SQL-type database systems.
XML	If your database system is XML (Tamino), enter the URL of the Tamino database including the name of the collection which is to be accessed in this database. See <i>Natural for Tamino</i> in the <i>Database Management System Interfaces</i> documentation for further information on how to access Tamino databases.

DBMS Tracing

The settings for **Adabas Trace**, **SQL Trace** and **XML Trace** are for error diagnostics reasons. They should only be filled out when requested by Software AG support. Software AG support will provide valid values in this case.

SQL Assignments

The tables for the SQL assignments are grouped under the following headings:

■ SQL Date/Time Conversion

This table is used to specify the conversion masks for Entire Access.

As Natural has only one specific time format, you must decide how this format should be interpreted in the context of SQL database access. There are several possibilities, however, there is only one possibility per SQL-type DBID which can be specified here.

DBID

Cannot be edited. Entries for the SQL-type databases defined in the [DBMS Assignments](#) table (see above) are automatically provided.

Mask

The value specifies the configuration for Entire Access. It also specifies the format used to retrieve the SQL information for DATE, TIME and DATETIME into fields of Natural data format A. The mask should match the RDBMS-specific configuration for the DATE, TIME or DATETIME character string representation.

Date

This mask (usually a sub-string of the Mask value) specifies the character string representation into which the fields of Natural data format D are converted during update or retrieval of SQL DATE columns.

Time

This mask (usually a sub-string of the Mask value) specifies the character string representation into which the fields of Natural data format T are converted during update or retrieval of SQL TIME or DATETIME columns.

Remark

You can enter your remarks here, for example, to document how the SQL DATE and TIME character string representation is configured on the database site.

For database systems that support a common data type both for date and for time, the hash character (#) can be used in the mask to skip non-relevant portions of DATETIME strings. For example, the following settings of the masks may be appropriate for the Microsoft SQL Server which supports a common DATETIME data type for date and time values:

```
Mask = YYYYMMDD HH:II:SS
Date = YYYYMMDD
Time = #####HH:II:SS
```

- **SQL Authorization**

This table is used to specify user IDs and passwords for an automatic login to an SQL database.

DBID

Cannot be edited. Entries for the SQL-type databases defined in the [DBMS Assignments](#) table (see above) are automatically provided.

Auto Login

When selected, a non-interactive login is performed to the database by using the values from this table row. Furthermore, the evaluations of the environment variables SQL_DATABASE_LOGIN, SQL_DATABASE_USER, SQL_DATABASE_PASSWORD, SQL_OS_USER and SQL_OS_PASSWORD are suppressed.

Type

To define the kind of authentication to be performed for this DBID, specify one of the following values:

Value	Description
DB	Database authentication.
OS	Operating system authentication.
DB_OS	Both, database authentication and operating system authentication.

Please note that it depends on the database system which kind of authentication is available. Furthermore, there are restrictions concerning operating system authentication in Entire Access. See the Entire Access documentation for a more detailed description.

DB User

The user ID for database authentication.

DB Password

The password for database authentication.

OS User

The user ID for operating system authentication.

OS Password

The password for operating system authentication.

XML Assignments

A table is shown which is used to specify user IDs and passwords for an automatic login to a Tamino database.

DBID

Cannot be edited. Entries for the XML-type databases defined in the [DBMS Assignments](#) table (see above) are automatically provided.

Auto Login

When selected, a non-interactive login is performed to the database by using the values from this table row. Furthermore, the evaluations of the environment variables XML_DB_USER, XML_DB_PASSWORD, XML_HTTP_USER and XML_HTTP_PASSWORD are suppressed.

DB User

The user ID for Tamino database authentication.

DB Password

The password for Tamino database authentication.

HTTP User

The user ID for web server authentication.

HTTP Password

The password for web server authentication.

Dictionary Server Assignments

With dictionary servers, you can access free rules and automatic rules maintained in Predict once you have access to Predict on a mainframe or UNIX host.



Note: A dictionary server (also referred to as Predict server) can be any Natural RPC server running in the same environment that Predict is installed in.

You can assign so-called dictionary servers (a free rule server and an automatic rule server) to one common logical server name. It is possible to define more than one logical server name where each name represents a different pair of free rule server and automatic rule server. The servers to be assigned can be located on different nodes.

In the global configuration file, you define all dictionary servers that are available for selection in the parameter file. The name of the logical server that is to be used must then be defined in the parameter file. See [Remote Dictionary Access](#) in the *Overview of Profile Parameters*.

You can specify the following information:

Name

Specify the name for a logical dictionary server.

Free Rules

Specify the server name and node name for the server that is to be used for remote access to Predict free rules.

Automatic Rules

Specify the server name and node name for the server that is to be used for remote access to Predict automatic rules.

HTTP Client Parameter Settings

The following parameters are used to define the name and port number for the HTTP server.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
HTTP proxy address to use	PROX
HTTP proxy port number	PROXPORT
Do not use proxy for addresses	NOPROX
HTTPS proxy address to use	SSLPRX
HTTPS proxy port number	SSLPRXPT
Do not use SSL proxy for addresses	NOSSLPRX

Natural Version-independent Operating System Files

You can specify a directory common to all installed Natural versions, which is to contain operating system-specific work files and temporary files used by all these Natural versions. These files are required for synchronization purposes and must not be deleted or modified by a Natural Administrator.



Note: If you use the system directory for temporary files, ensure that no automatic procedures regularly delete the contents.

NDV Environment Assignments

When you choose this function, the currently defined ports are shown in a window. When you edit a port number (by entering "E" next to the port number and pressing ENTER), the environment assignments are shown. You can also choose <CREATE>, which is shown at the top of the window, to add environment assignments for further ports.

You can assign user-specific environments for any Natural development server (NDV). To do so, use the following syntax in the **Environment** column:

VARIABLE=VALUE[;VARIABLE=VALUE]...

Example:

```
+----- Environment settings for NDV port <1234> -----+
| C User      Environment                                     |
| USER1      MYWRKFILE=/natural/prog/user1;NDVSESSION=naturalins |
| USER2      MYWRKFILE=/natural/prog/user2;NDVSESSION=naturaltr;NAT_TRACE=1 |
| USER3      MYWRKFILE=/natural/prog/user3;NDVSESSION=naturalsql |
|                                                     |
```

In the above example, 3 user environments for the server started on port 1234 are defined. The environment variable `MYWRKFILE` defines the path of a Natural work file. This environment variable can then be used in the **Work Files** section of a parameter file to define the location of the work file, depending on the defined user.

The variable `NDVSESSION` specifies the Natural image to be loaded when a mapping to the server takes place. It is only known by the Natural development server.

Suppose Natural Studio is mapping to a Natural development server with port 1234 and user ID `USER2`: the server loads `naturaltr` as the Natural session and any Adabas call of a started application will be traced. Work files defined with `MYWRKFILE` will be written to the location `/natural/prog/user2`.

For information on how to configure a Natural development server under UNIX, see the Natural Development Server for UNIX documentation.

Printer Profiles

Printer profiles are used for printing additional reports, for hardcopies and for batch output generation. They recognize particular Natural field attributes and insert the appropriate control sequences (see below) as defined in the profile.

With the ability to translate Natural field attributes into escape sequences, you can control your printer in various ways by using the right profile name, and you can use the print features of a given device by using simple attributes in Natural programs.

Each profile that you define can be assigned to a Natural report number either statically by using the Configuration Utility (see **Device/Report Assignments**), or dynamically by using the `DEFINE PRINTER` statement within a Natural program.

You can specify the following information:

Profile name

Specify the name of a new printer profile. Or press PF2 to select a defined printer profile from a list.

External character set

Optional. An external character set must be defined, for example, if a printer's character set is different from the system character set, or if you wish to preserve Unicode data by outputting the data in UTF-8 format.

You can specify the name of the external character set. If the name is "UTF-8", the data are converted into UTF-8 format. Otherwise, any name can be used, but it must agree with the name used in the character translation table definition in the file *NATCONV.INI*. For example, if the internal character set name (as defined near the top of the *NATCONV.INI* file) is "ISO8859_1" and you define "FRED" as the name of the external character set in the printer profile, a section with the following name must be defined in the file *NATCONV.INI*, otherwise no character translation is performed:

```
[ISO8859_1 -> FRED]
```

Note that, in the non-UTF-8 case, the data are first converted from the in-memory format (UTF-16) into the system code page, prior to any *NATCONV.INI*-based translation being applied.

For further information on the *NATCONV.INI* file, see *Support of Different Character Sets with NATCONV.INI* in the *Operations* documentation.

Trigger / Leading command / Trailing command

You can specify whether commands to be triggered at job, page or field level.

Option	Description
Trigger	A triggering event controls the level on which specified printer control sequences are to be applied. The names of the available triggering events are listed below. Specify the name of a triggering event in the Trigger field to display the associated leading and trailing control sequences. You can also press PF2 to select a triggering event from a list.
JOB	Use this triggering event if you want your control sequences to apply to an entire print job. The specified control sequences will represent the job header and/or job trailer respectively.
PAGE	Use this triggering event if you want the control sequences to apply to each physical output page. The specified control sequences will then represent the page headers and/or page trailers respectively.
FIELD	Use this triggering event if you want the control sequences to apply to each field (regardless of its AD, CD or PM attributes).
AD and CD	These triggering events stand for specific attributes of the session parameters AD (attribute definition) and CD (color

Option	Description
	<p>definition). Use these triggering events if you want the control sequences to be applied at field level only. Any field in a Natural program with corresponding attributes will then cause these control sequences to take effect. See the <i>Parameter Reference</i> for details on these session parameters.</p> <p>PM=I Use this triggering event if you want the control sequences to apply to each field with an opposing print direction to that of the page. See the session parameter <i>PM</i> (print mode) and the terminal command <i>%V</i> (control of print mode) for more information.</p>
Leading command	The leading control sequence is inserted immediately before the triggering event (for example, to define a job header or to set attributes for field representation).
Trailing command	The trailing control sequence is inserted immediately after the triggering event (for example, to define a job trailer or to reset attributes previously set).

For each control sequence, you can specify the control characters in either alphanumeric or hexadecimal format.



Notes:

1. Blank characters must always be specified in hexadecimal format (^20). If they were entered directly, they would be confused with the blanks used to delimit tokens in the configuration file.
2. The escape character must always be specified in hexadecimal format (^1b) because this is a non-printable character.

Example

This example shows the PCL command for setting a proportional character spacing in both alphanumeric and hexadecimal format. The escape character "**^1b**", which must always be specified in hexadecimal format, corresponds to the decimal value 27.

Alphanumeric format:

```
^1b(s1P
```

Hexadecimal format:

```
^1b^28^73^31^50
```

Delete profile

Mark this field with "Y" if you want to delete the currently displayed profile.

System Files

A list of existing Natural system file assignments is shown.

The paths for all possible system files (that is: for all possible DBID/FNR combinations) are defined in the global configuration file. These must be valid paths which indicate the physical location of the system files on the disk.

You can specify the following information:

DBID

The database ID.

FNR

The file number.

Alias name

This feature applies when working in a remote development environment (SPoD) with Natural for Windows.

When you specify an alias name for a system file, this name is shown in Natural Studio's library workspace.

Path

The path to the system file.

The DBID/FNR combination for each defined system file can then be defined in the parameter file. See [System Files](#) in the *Overview of Profile Parameters*.

Local Configuration File

This parameter group contains the following categories:

- [Administrator Assignments](#)
- [Buffer Pool Assignments](#)

- Installation Assignments

Administrator Assignments

You can specify whether a user is to be a Natural administrator or a configuration administrator. The following combinations are possible:

```
+----- Administrator Assignments -----+
| C User Name   Administrator              |
|              NATURAL CONFIGURATION     |
| NAT1         Y           Y               |
| NAT2         Y           N               |
| NAT3         N           Y               |
| NAT4         N           N               |
+-----+-----+-----+-----+-----+
```

- When a user is defined as both Natural administrator and configuration administrator (Y/Y), the user can modify all parameter files as well as the global and local configuration files. When such a user invokes the Configuration Utility, "(Config)" is shown in the right-hand corner of the screen (instead of the user ID).



Note: Only Natural administrators can be defined as configuration administrators.

However, when a user is defined only as a configuration administrator (N/Y), the above information applies. This is handled as if the user has been defined as both Natural administrator and configuration administrator (Y/Y).

- When a user is defined only as a Natural administrator (Y/N), the user can modify all parameter files. The **Configuration** menu is not shown; thus, modification of the configuration files is not possible. When such a user invokes the Configuration Utility, "(Admin)" is shown in the right-hand corner of the screen (instead of the user ID).
- When a user is neither defined as a Natural administrator nor as a configuration administrator (N/N), the user can only modify his own parameter file. When such a user invokes the Configuration Utility, the user ID is shown in the right-hand corner of the screen.

Buffer Pool Assignments

A table is shown which is used to define buffer pools. See also *Natural Buffer Pool* in the *Operations* documentation.

You can specify the following information:

BP Name

The name of the Natural buffer pool. The name can be up to 8 characters long. The name of the default buffer pool is NATBP. See also BPID in the *Parameter Reference*.



Caution: Do not delete the default buffer pool NATBP, as it is possible that Natural may not function properly anymore.

BP Size

The size of the Natural local buffer pool. This setting is used during startup of the Natural buffer pool by the NATBPSRV server to create a global shared memory segment of the specified size.

Possible settings: 1 - 1024 MB.

Default setting: 1.

Maximum Users

The maximum number of users that can have simultaneous access to the buffer pool. This number determines the sizes of some internal tables stored inside the shared memory during startup.

Possible settings: 1 - 5000.

Default setting: 20.

Directory Entries

The number of directory entries. This value is used by the NATBPSRV server during the start of the buffer pool. If the maximum value has been reached, this value is automatically increased.

Only for the purposes of a read-only buffer pool, this number should be 10% larger than the maximum number of objects being loaded into such a buffer pool. The directory must not occupy more than 10% of the buffer pool size (**BP Size**). The size of a directory entry can be assumed to be about 1 kilobyte.

Possible settings: 10 - 999999.

Default setting: 10.

Shared Memory Key

The shared memory key for the IPC facilities that the buffer pool is using. Ask your UNIX administrator to supply the value for this key.



Important: The shared memory key must be unique across your entire UNIX system.

Semaphore Key

The semaphore key for the IPC facilities that the buffer pool is using. Ask your UNIX administrator to supply the value for this key.



Important: The semaphore key must be unique across your entire UNIX system.

Read Only

When set to "ON", this is a special buffer pool which only allows read access. In this case, the values defined for the maximum number of users and for the semaphore key are ignored.

For further information, see *Read-Only Buffer Pool* in the *Operations* documentation.

Alternate BP Name

For a read-only buffer pool, it is possible to define the name of an alternate buffer pool. The alternate buffer pool must also be a read-only buffer pool.

For further information, see *Alternate Buffer Pool* in the *Operations* documentation.

Installation Assignments

You can specify the following:

Path to parameter (PARM_PATH)

The location of the Natural parameter files.

Path to user profiles (PROFILE_PATH)

The location of the Natural user profiles.

Global configuration file (CONFIG_NAME)

The name and location of the global configuration file (default name is *NATCONF.CFG*).

Natural error file directory (ERROR_FILES)

The location of the Natural error files.

Natural I/O conversion table (NATCONV)

The name of the file which contains the character translation tables used with the internal character set ISO-8859-1. By default, this file is called *NATCONV.INI*. See *Support of Different Character Sets with NATCONV.INI* in the *Operations* documentation for further information.

Natural TMP directory (TMP_PATH)

The location of Natural temporary output.

Binary load path (NATEXTLIB)

The location of binary libraries of further Software AG products using Natural.

Terminal database (NATTCAP)

The name of the database which contains the descriptions of the terminal capabilities for each terminal type supported by Natural. See *Natural Termcap (NATTERMCAPI) Utility* in the *Utilities* documentation for further information.

Text file path (TXT_PATH)

The location of messages, choices and help texts used by the editors and by the Natural Termcap utility.



Note: When specifying a path, you can use `$NATDIR` and `$NATVERS`. When you press `TAB` after having specified a valid path, the path is automatically expanded.

4 Overview of Profile Parameters

- Database Management 28
- Natural Execution Configuration 30
- Natural Development Environment 44
- Product Configuration 45
- Client/Server 46

This chapter provides information on the profile parameters that can be set in the NATPARM parameter file (or an alternative parameter file).

File	Edit	Configuration	Search
	Database Management...		
	Natural Execution Configuration...		
	Natural Development Environment...		
	Product Configuration...		
	Client/Server...		

The individual parameters are divided into the parameter groups according to their functions.

This section lists all parameters that can be set with the Configuration Utility. However, if you are not an administrator, not all of these parameters are displayed. Some parameters can only be seen and thus set by users who are defined as administrators (see also *Administrator Assignments* in the local configuration file).

For a full description of all available profile parameters, see the *Parameter Reference*.

Database Management

This parameter group contains the following categories:

- General Parameters
- Adabas Specific
- Administrator DBMS Assignment
- User DBMS Assignment

See also: *Database Management System Assignments* in the global configuration file.

General Parameters

The following profile parameters apply to the supported database management systems.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Database updating	DBUPD
Execution of END/BACKOUT TRANSACTION	ET
ET at end of program	ETEOP

Adabas Specific

If Natural is used with Adabas, review the following profile parameters and, if necessary, adjust the default values to meet your specific requirements.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Adabas user identification	ETID
Multifetch	MFSET
Terminate when no ISN found with FIND statement	RCFIND
Terminate when no ISN found with GET statement	RCGET
Record hold processing	WH
Adabas OPEN/CLOSE processing	OPRB

Administrator DBMS Assignment

The following profile parameters are used to assign administrator-specific database management system settings.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Administrator logical files	LFILE
XA database list	XADB
Translation of file number	TF

User DBMS Assignment

The following profile parameters are used to assign user-specific database management system settings.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
User database ID	UDB
Database for transaction data	ETDB
User logical files	LFILE

Natural Execution Configuration

This parameter group contains the following categories:

- Batch Mode
- Buffer Sizes
- Character Assignments
- Command Execution
- Date Representation
- Device/Report Assignments
- Error Handling
- Field Appearance
- Limits
- Program Loading and Deletion
- Regional Settings
- Report Parameters
- Steplibs
- System Files
- System Variables
- Work Files

Batch Mode

The profile parameters which affect the batch mode behavior of Natural are arranged on the following pages:

- Channels
- Appearance
- Frame Characters

See also *Natural in Batch Mode* in the *Operations* documentation.

Channels

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Input data file name	CMOBJIN
Input data code page	CPOBJIN
Input commands file name	CMSYNIN
Input commands code page	CPSYNIN
Output file name	CMPRINT
Output code page	CPPRINT

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Natural log	NATLOG
Enable error processing	CC

Appearance

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Similar output	BMSIM
Display input data	ECHO
Display session-end message	ENDMSG
Display trailing blanks	BMBLANK
Display control characters	BMCONTROL
Display process time	BMTIME
Display window title	BMTITLE
Display Natural version	BMVERSION

Frame Characters

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Frame characters	BMFRAME

Buffer Sizes

Natural uses several buffer areas for the storage of programs and data. You may need to adjust their sizes in order to achieve maximum buffer efficiency.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Storage for sort program	SORTSIZE
Source area size	SSIZE
Work area size	USIZE
SAG editor buffer pool size	EDTBPSIZE
SAG editor logical files	EDTLFILES

See also [Buffer Pool Assignments](#) in the local configuration file.

Character Assignments

The following profile parameters are used to change default character assignments.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Terminal command character	CF
Clear key character	CLEAR
Decimal character	DC
Filler character	FC
Help character	HI
Input assign character	IA
Input delimiter character	ID
Thousands separator character	THSEPCH

Once a character has been defined to replace a default character, this character cannot be used as data.

Command Execution

The following profile parameters are used to control the execution of commands.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Enable command mode	CM
Enable terminal command %% and %	ESCAPE
Disable Natural commands	NC
Dynamic recataloging	RECAT

Date Representation

The following profile parameters are used to control the representation of dates.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Date format in output	DFOUT
Date format in STACK	DFSTACK
Date format in report titles	DFTITLE
Date format	DTFORM
Maximum year	MAXYEAR
Year sliding window	YSLW

Device/Report Assignments

These parameters are used to modify your screen and printer configurations as well as your report assignments.

See also the profile parameter `CMPRT nn` which is used for additional reports in batch mode.

Device Assignments

The device assignments are shown in a scrollable list of configurable logical devices (`VIDEO` and the logical printers `LPT1` to `LPT31`) as used in the `DEFINE PRINTER` statement. The following information can be changed:

Close mode

You can select a different close mode for each logical printer. Possible values are:

Close Mode	Description
Auto	Printers are automatically closed at the end of a program or when the execution of a called method is terminated. This is the default setting.
User	Closing of printers is controlled by the user. The printer is closed, if one of the following conditions is met: <ul style="list-style-type: none"> ■ a <code>CLOSE PRINTER</code> statement is executed, ■ a <code>DEFINE PRINTER</code> statement is executed, ■ the session terminates.

Notes:

1. Instead of using the device `TRANSFER`, it is recommended to use a logical printer device (one of `LPT1` to `LPT31`) where the device destination of the physical output device has been set to the value "E" (send data to an Entire Connection terminal). See also *Defining Printers* in the *Natural Connection* documentation.
2. If the device `TRANSFER` is used in the report assignments, the close mode for this device is determined as follows: The report number to which the device `TRANSFER` has been allocated is used as the device number (for example, for report number 2 the device `LPT2` is used). The close mode that is defined for this device is also used as the close mode setting for the device `TRANSFER`.

The following program example describes the user-controlled closing of printers. Note that the close mode must be first set in the parameter file.

```
DEFINE DATA
  LOCAL
  1 #OUT (A8)
END-DEFINE
*
#OUT := *PROGRAM
*
* Write to report with close mode "user"
WRITE (1) 'OUTPUT PROGRAM: ' #OUT
*
* Write to report with close mode "auto"
WRITE (2) 'OUTPUT PROGRAM: ' #OUT
*
* Close printer 1
* Printer 1 is user-controlled
CLOSE PRINTER (1)
*
END
```

Line Size

See the description of the LS parameter.

Page Size

See the description of the PS parameter.

Max. Pages

See the description of the MP parameter.

Physical Output Device

The path and/or name of the program, script (including any arguments) or file that should receive the print data. Programs and scripts can read the incoming print data via `stdin`.

For example, you can specify the standard print command of your printer spooler with all options as described below.

- For UNIX System V:

```
lp options
```

- For BSD-UNIX:

```
lpr options
```

Line size, page size and maximum page number should be compatible with your hardware printer assignments.

Device Destination

Determines how Natural should interpret the information in the **Physical Output Device** field.

Option	Description
F	If "F" is specified, this is assumed to be a file specification.
D	If "D" is specified, this is assumed to be a command or script.
E	If "E" is specified, all data that are written to this logical printer are sent to an Entire Connection terminal. That is: output data which normally appear on the screen are written into an NCD file. See also <i>Defining Printers</i> in the <i>Natural Connection</i> documentation.



Note: The following applies to the **Physical Output Device** field if the **Device Destination** field has been set to "F". As with work files, a file name can be defined by using environment variables. Any existing file of the same name at the specified location is normally overwritten, unless the entered file name is immediately prefixed by two right angle bracket characters (>>).

Report Assignments

The report assignments are shown in a scrollable list in which you can assign a Natural report number (report 1 to report 31) to a logical device name. The following information can be changed:

Override default report number

See the *Parameter Reference* for a description of the profile parameter MAINPR.

Device

For each report number, you can select another output medium. Possible values are (you can also press PF2 to select one of these values):

Value	Description
LPT1 to LPT31	Outputs the report data to the corresponding physical device (see the Physical Output Device field in the device assignments).
SOURCE	Outputs the report data to the source area. The user can then, for example, issue the EDIT command to open the output data in the editor and save it accordingly.
DUMMY	Report data are discarded.
INFOLINE	Report data are sent to the infoline. See also the description of the terminal command %X.

Report 0 is always implicitly assigned to VIDEO (that is, the output is shown on the screen). No other report number can be assigned to VIDEO.

Profile

In addition to the name of the logical device, you can assign a printer profile that has been defined in the [Printer Profiles](#) node of the global configuration file. All defined printer profiles can be selected by pressing F2. Select the blank entry if you do not want to use any of these profiles.

Error Handling

The following profile parameters are used to control error handling within Natural.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Error processing for PA/PF keys	IKEY
Display system error messages in full	MSGSF
Automatic REINPUT	REINP
Sound terminal alarm	SA
Sound bell on syntax error	SNAT
Zero division	ZD
Suppress message number prefix NAT	NOAPPLERR
Allow runtime interrupt	RTINT

Field Appearance

The following profile parameters influence I/O handling.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Control variable modified at input	CVMIN
Filler chars protected	FCDP
Enable lowercase	LC
Numeric fields entry	NENTRY
Overwriting protected	OPF
Zero printing	ZP
Print mode	PM

Limits

The following profile parameters are used to prevent a single program from consuming an excessive amount of internal resources.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Error on loop limit	LE
Processing loop limit	LT
Maximum number of DBMS calls	MADIO
Maximum number of program calls	MAXCL
Size of page data set	PD

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
System time delay	SD

Program Loading and Deletion

The following profile parameters are used to control the dynamic loading and deletion of programs.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Dynamic loading of non-Natural programs	CDYNAM
Allow dynamic parameter	DYNPARM
Release GDA in utility mode	FREEGDA
Read only access to system files	ROSY
Buffer pool search first	BPSFI
Error transaction program name	ETA
Program command line parameter	PRGPAR
Program to receive control after Natural termination	PROGRAM
Natural stack command line	STACK

Regional Settings

The following profile parameters are used to control the country- or region-specific settings of Natural.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Day differential	DD
Time differential	TD
User language	ULANG
Use UTF-8 format for sources	SUTF8
Retain source format	SRETAIN
Code page conversion error	CPCVERR
Default code page	CP
Substitution character	SUBCHAR
Display order of output data	DO

Report Parameters

The following profile parameters are used to control various attributes of Natural reports.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Page eject	EJ
Numeric edit mask free mode	EMFM
Line size	LS
Page size	PS
Spacing factor	SF
Default input terminal mode	IM

Steplibs

The following parameters are used to manage the steplibs.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
*STEPLIB	STEPLIB
Steplib Extension	For information on how to define additional steplibs and how to use the columns in this table, see <i>Additional Steplib Assignments</i> below.

Additional Steplib Assignments

You can define up to eight libraries that can be searched for objects which cannot be found in the current library. These additional steplibs are defined in a table. The steplib number is shown in the first column. The following information can be specified:

Name

The name of an existing library.

DBID

The database ID of the file system where the library is located.

FNR

The file number of the file system where the library is located.



Note: The following application programming interfaces are available for handling multiple steplibs: USR3025N (when Natural Security is not installed) and USR4025N (when Natural Security is installed).

System Files

The following profile parameters are used to specify the Natural system files.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
FNAT (Natural system file for system programs)	FNAT
FUSER (Natural system file for user programs)	FUSER
FSEC (Natural Security system file)	FSEC
FDIC (Predict system file)	FDIC
FDDM (Natural system file for DDMs)	FDDM

The physical location of each system file is defined in the global configuration file. See [System Files](#) in the global configuration file.

System Variables

The following profile parameters are used to adjust Natural system variables for the start of a Natural session.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Automatic logon	AUTO
Startup library	INIT-LIB
Startup program	STARTUP
User ID	USER

See also the *System Variables* documentation.



Note: You can use the command line parameters `NATVERS` (to specify the Natural version) and `PARM` (to specify a specific Natural parameter file) at session startup. These parameters can only be specified dynamically, therefore, they cannot be modified or viewed within the Configuration Utility.

Work Files

The following profile parameters can be used to specify work file settings.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Entire Connection protocol mode	ECPMOD
Entire Connection local NCF protocol	NCFVERS
PC support	PC
Max. work file number	WORK
Alternate sort work file names	TMPSORTUNIQ
Work file open on first access	WFOPFA
Work files	For information on how to define work files and how to use the columns in this table, see Work File Assignments below.

See also the profile parameter `CMWRK nn` which applies to batch mode.

See also *Work Files* in the *Operations* documentation.

Work File Assignments

The work file assignments are shown in a scrollable list containing work files with the numbers 1 to 32. The work file number is shown in the first column. When a work file name has already been defined for a work file number, this information (path and name) is shown in the **Name** column.

The following information can be changed:

Type

For each work file number, you can select another work file type. Possible values are (you can also press PF2 to select one of these values):

Work File Type	Description
Default	Determines the file type from the extension for upward compatibility.
Transfer	Used to transfer data to and from a PC with Entire Connection. This work file type represents a data connection between a Natural session on UNIX or OpenVMS and an Entire Connection terminal on a PC. The work file data is written in Entire Connection format on the PC.
SAG	Binary format.
ASCII	Text files with records terminated by a linefeed (LF). When you press PF4, you can change the properties for this work file type: a dialog box appears in which you can specify the following information:

Work File Type	Description
	<p>Code page You can specify the code page that is to be used for writing the work file (ICU code page name; IANA name recommended). When a code page is not defined here, the default code page defined with CP is used.</p>
<p>ASCII compressed</p>	<p>ASCII format where all trailing blanks are removed.</p> <p>When you press PF4, you can change the properties for this work file type: a dialog box appears in which you can specify the following information:</p> <p>Code page You can specify the code page that is to be used for writing the work file (ICU code page name; IANA name recommended). When a code page is not defined here, the default code page defined with CP is used.</p>
<p>Entire Connection</p>	<p>With this work file type, you can read and write (for example, with the statements READ WORK FILE and WRITE WORK FILE) directly to a work file in Entire Connection format on the local disk.</p> <p>Note: No transfer to a PC is possible. The Entire Connection terminal is not used in this process.</p>
<p>Unformatted</p>	<p>A completely unformatted file. No formatting information is written (neither for fields nor for records).</p> <p>When you press PF4, you can change the properties for this work file type: a dialog box appears in which you can specify the following information:</p> <p>Code page You can specify the code page that is to be used for writing the work file (ICU code page name; IANA name recommended). When a code page is not defined here, the default code page defined with CP is used.</p>
<p>Portable</p>	<p>Files which can handle dynamic variables exactly and can also be transported (for example, from a little endian machine to a big endian machine, and vice versa).</p>
<p>CSV</p>	<p>Comma-separated values. Each record is written to one line in the file. By default, a header is not written. The default character which is used to separate the data fields is a semicolon (;).</p> <p>When you press PF4, you can change the properties for this work file type: a dialog box appears in which you can specify the following information:</p> <p>Header When set to "ON", a header with the Natural field names is written to the file.</p> <p>Separator character You can select a different separator character. If you require a separator character which is not listed, you can enter it.</p> <p>Code page You can specify the code page that is to be used for writing the work file (ICU code page name; IANA name recommended). When a code page is not defined here, the default code page defined with CP is used.</p>

For information on the work file formats which result from a specific work file type, see *Work File Formats* in the *Operations* documentation.

Close mode

You can select a different close mode for each work file. Possible values are:

Close Mode	Description
Auto	<p>This is the default setting. Work files are automatically closed at the end of a program or when the execution of the first called method in a method call hierarchy is terminated and the first called method was called over COM.</p> <p>If the first called method was called locally (not over COM), the work files are not closed.</p> <p>Example: Method A (called over COM) invokes method B which in turn invokes method C. When the execution of method A is terminated, all work files are closed.</p>
User	<p>Closing of work files is controlled by the user. The work file is closed, if one of the following conditions is met:</p> <ul style="list-style-type: none"> ■ a CLOSE WORK FILE statement is executed, ■ a DEFINE WORK FILE statement is executed, ■ the session terminates.

The following program example describes the user-controlled closing of work files. Note that the close mode must be first set in the parameter file.

```

DEFINE DATA
  LOCAL
  1 #OUT (A8)
END-DEFINE
*
#OUT := *PROGRAM
*
* Write to work file with close mode "user"
WRITE WORK 1 'OUTPUT PROGRAM: ' #OUT
*
* Write to work file with close mode "auto"
WRITE WORK 2 'OUTPUT PROGRAM: ' #OUT
*
* Close work file 1
* Work file 1 is user-controlled
CLOSE WORK FILE 1
*
END
    
```

Attributes

You can define different attributes for each work file. You can specify one attribute for each of the following categories:

Category	Possible Attributes	Default Attribute	Description
Append mode	NOAPPEND or APPEND	NOAPPEND	Decides whether new data overwrites the current data in the work file or whether new data is appended at the end of the current data.
Keep/delete work file after close	KEEP or DELETE	KEEP	Decides whether the work file is kept after closing it or whether it is deleted.
Write byte order mark (BOM)	BOM or NOBOM	NOBOM	<p>Decides whether a byte order mark is written in front of the work file data.</p> <p>Only available for the work file types which write code page data: ASCII, ASCII compressed, Unformatted and CSV. For these work file types, the attribute BOM can only be set, if the code page UTF-8 is defined for the work file (see the description of the Type column).</p> <p>If a work file of another type is written or a code page other than UTF-8 is defined, the specification of the attribute BOM is ignored during runtime.</p> <p>See also <i>Work Files and Print Files on Windows, UNIX and OpenVMS Platforms</i> in the <i>Unicode and Code Page Support</i> documentation.</p>
Remove/keep carriage return	REMOVECR or KEEPPCR	REMOVECR	<p>Decides whether carriage return characters are removed when reading an ASCII work file or whether they are kept.</p> <p>Only available for the following work file types: ASCII and ASCII compressed. If one of the attributes is set for another work file type, it is ignored during runtime.</p> <p>Caution: Use KEEPPCR with care. ASCII format is only recommended for alphanumeric data. Binary data should not be processed with ASCII work files. When you use KEEPPCR, the work file record may include carriage return characters.</p> <p>The use of KEEPPCR only makes sense when reading ASCII work files which have been written on UNIX or OpenVMS. It does not make sense to use KEEPPCR with ASCII work files which have been written on Windows.</p>

Attributes cannot be defined for the work file type "Transfer".

You can enter a unique abbreviation for each attribute. The delimiter character can be one of the following: blank, comma(,) or semicolon (;). Example: "a, d" which stands for APPEND and DELETE.

You can only specify one attribute for each category. You can specify them in any sequence. When no attributes are specified in the table, the default values are used.

Name

Specify the location (path and name) of the work file.

See also *Defining Work File Names with Environment Variables* in the *Operations* documentation.

Natural Development Environment

This parameter group contains the following categories:

- [Compiler Options](#)
- [Environment Assignments](#)
- [Remote Debugging](#)

Compiler Options

The following profile parameters are used to set options for the Natural compiler.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Compiler output	CO
Interpretation of database field short names	DBSHORT
Dump generation	DU
Length/format specification	FS
Keyword checking	KCHECK
Structured mode	SM
Generate symbol tables	SYMGEN
Syntax error control	SYNERR
Dynamic thousands separator	THSEP
Translate quotation marks	TQMARK
MASK compatible with MOVE EDITED	MASKCME
Parameter checking for CALLNAT statement	PCHECK
Internal sign representation of format P	PSIGNF
Endian mode	ENDIAN
Generation of global format identifiers	GFID
Active cross reference	XREF

See also the description of the system command `COMPOPT`.

Environment Assignments

The following profile parameters are used to adjust Natural environment variables.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Other editor	EDITOR
Usage of editor ring buffer	EDTRB
Access to operating system from Natural	SHELL

Remote Debugging

The following profile parameters are used to allow for remote debugging.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Remote debugging	RDACTIVE
Node name	RDNODE
Port number	RDPORT

See also the *Debugger* documentation.

Product Configuration

This parameter group contains the following categories:

- [Entire Transaction Propagator](#)
- [Entire System Server](#)
- [Function Keys](#)
- [Web I/O Interface](#)

Entire Transaction Propagator

The following profile parameters are used in conjunction with Software AG's Entire Transaction Propagator.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
ETP database list	ETPDB
Size of ETP work area	ETPSIZE

Entire System Server

The following profile parameter is used in conjunction with Software AG's Entire System Server Interface.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
ESX database	ESXDB

Function Keys

The following profile parameter is used to assign values to PA, PF and CLEAR keys.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
CLR, PA1 to PA3, PF1 to PF24	KEY

Web I/O Interface

The following profile parameter is used for Unicode support with the Natural Web I/O Interface.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Web I/O Interface	WEBIO

Client/Server

This parameter group contains the following categories:

- [HTTP Parameters](#)
- [Remote Dictionary Access](#)

- [Remote Procedure Call](#)

HTTP Parameters

The following profile parameter is used to provide HTTP support.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Timeout for REQUEST DOCUMENT	RQTOUT

Remote Dictionary Access

The following profile parameter is used for remote dictionary access.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Name of logical dictionary server	USEDIC When a logical dictionary server has been defined in the global configuration file, you can specify its name here. See Dictionary Server Assignments in the global configuration file.

Remote Procedure Call

The profile parameters which apply to the Natural Remote Procedure Call (RPC) are grouped under the following headings:

- [RPC \(General\)](#)
- [RPC \(Client\)](#)
- [RPC \(Server\)](#)

RPC (General)

These profile parameters apply to both client and server.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Request buffer size (KB)	MAXBUFF
ACI version	ACIVERS
Code page	CPRPC

RPC (Client)

These profile parameters apply to the client only.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Automated remote execution	AUTORPC
Retry service on alternative server	TRYALT
Send-buffer compression	COMPR
Request timeout (sec)	TIMEOUT
Default server	DFS
Remote directory servers	RDS
Library for service directory	RPCSDIR

RPC (Server)

These profile parameters apply to the server only.

Option	See the <i>Parameter Reference</i> for a description of this profile parameter
Start session as RPC server	SERVER
Logon required for server request	LOGONRQ
Server name	SRVNAME
Server node	SRVNODE
Server user ID	SRVUSER
Server wait time (sec)	SRVWAIT
Number of reconnect attempts	SRVTRY
Wait time between attempts	
Server termination event	SRVTERM
Server commit time	SRVCMIT
Transport protocol	TRANSP
RPC trace level	TRACE
Trace on error	

5 Invoking Natural with an Alternative Parameter File

By default, Natural is invoked with the parameter file NATPARM. When you have **created** your own customized parameter file, you can also invoke Natural with this parameter file.

See also *Dynamic Assignment of Parameter Values* in the *Operations* documentation.

▶ **To invoke Natural with an alternative parameter file**

- Enter the following command at the UNIX system prompt:

```
natural PARM=parameter-file
```

