

Natural

Natural Connection

Version 9.3.2

September 2025

This document applies to Natural Version 9.3.2 and all subsequent releases.

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

Copyright © 1992-2025 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

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

Detailed information on trademarks and patents owned by Software GmbH and/or its subsidiaries is located at <https://softwareag.com/licenses>.

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

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software GmbH.

Document ID: NATUX-NNATCONNECTION-932-20250930

Table of Contents

Preface	v
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Main Functionality	5
Interactive Data Transfer	6
Integration	6
Security	6
3 Preparing to Use Natural Connection	7
Displaying Work File and Printer Settings	8
Defining Work Files	10
Defining Printers	10
Activating the PC Connection	11
Deactivating the PC Connection	12
4 Natural Statements	13
5 Processing Work Files and Nested Loops	15
Work File Format and Attributes	16
Maximum File Transfer Record Length for Natural Connection	16
Streaming	16
Dynamic Variables in READ WORK FILE	17
Nested READ Loops	17
Subsequent READ Loops	19

Preface

Natural Connection provides an access method required for transferring data, reports, Natural objects and sources from a Natural Linux computer to a PC.

The prerequisite for the use of Natural Connection is Entire Connection. For information on Entire Connection, refer to the relevant documentation.

This documentation is organized under the following headings:

Main Functionality	Describes the main functionality that is relevant for Linux.
Preparing to Use Natural Connection	Describes the tasks that must be accomplished to allow download and upload.
Natural Statements	Describes the Natural statements that are relevant for processing information between Linux and the PC by using Natural Connection.
Processing Work Files	Describes restrictions on the use of work file attributes, the support of work file formats and the impact of READ loops.



Note: Natural Connection is part of the Natural for Linux installation; it is installed automatically when Natural for Linux is installed.

1

About this Documentation

■ Document Conventions	2
■ Online Information and Support	2
■ Data Protection	3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

Product Training

You can find helpful product training material on our Learning Portal at <https://learn.software-ag.com>.

Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software GmbH resources.

Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Main Functionality

■ Interactive Data Transfer	6
■ Integration	6
■ Security	6

This chapter describes the main functionality provided by Natural Connection that is relevant for Linux.

For details regarding the features of the PC component, see the Entire Connection documentation.

Interactive Data Transfer

Data can be downloaded to and uploaded from the PC directly to/from an online program.

Data transfer and conversion to many formats (for example, Lotus 1-2-3, dBASE, ASCII, HTML, XML, Basic, DIF, Binary) is performed in one step. Sophisticated compression techniques are used.

Integration

Using Natural Connection, you can create integrated applications that use both Linux and PC resources.

For example, Natural creates files that you can then further process on the PC. In this way, graphics, spreadsheets, or binary data can be distributed to other users.

Linux applications can use PC printers.

Security

A key consideration for any information network is effective security and control. Natural Security on Linux provides comprehensive control facilities.

Detailed security profiles are assigned on a user-by-user basis to control access to data and programs.

Other significant objects secured are the individual keywords in the Natural syntax. This can restrict some users, for example, to retrieving data from Linux databases, while others have the full capability for retrieval and update of Linux data.

Natural Security also protects against excessive use of system resources by individual users. This is vital when PC users may be employing powerful Linux features for the first time.

As an additional safeguard, Natural Connection can also encrypt all data downloaded from Natural.

3

Preparing to Use Natural Connection

■ Displaying Work File and Printer Settings	8
■ Defining Work Files	10
■ Defining Printers	10
■ Activating the PC Connection	11
■ Deactivating the PC Connection	12

This chapter describes the tasks that must be accomplished to allow download and upload.

In order to download and upload data, a work file must be designated as a PC file.

To download reports, a printer must be designated as a PC printer.

Displaying Work File and Printer Settings

Work File Settings

To display your current work file settings, invoke the Configuration Utility and from the **Edit** menu, choose **Natural Execution Configuration > Work Files**. A screen similar to the following appears:

+----- Work Files -----+			
Entire Connection protocol mode.....	(ECPMOD)	ON	
Entire Connection local NCF protocol	(NCFVERS)	2	
PC support.....	(PC)	OFF	
Max. work file number.....	(WORK)	32	
Alternate sort work file names.....	(TMPSORTUNIQ)	OFF	
Work file open on first access.....	(WFOPFA)	ON	
		Close	
Number	Name	Attributes	Mode Type
-----	-----	-----	-----
1		Auto	Default
2		Auto	Default
3		Auto	Default
4		Auto	Default
5		Auto	Default
6		Auto	Default
7		Auto	Default
8		Auto	Default
9		Auto	Default
10		Auto	Default
11		Auto	Default
12		Auto	Default
13		Auto	Default
ON or OFF			

Printer Settings

To display the printer settings, invoke the Configuration Utility and proceed as follows:

1. From the **Edit** menu, choose **Natural Execution Configuration > Device/Report Assignments > Device Assignments**. A screen similar to the following appears:

Device Assignments								
Logical Device	Close Mode	Line Size	Page Size	Max Page	Physical Output Device			
VIDEO0	Auto	80	24	32767	VIDEO0			D
LPT1	Auto	80	66	32767	LPT1			D
LPT2	Auto	80	66	32767	LPT1			D
LPT3	Auto	80	66	32767	LPT1			D
LPT4	Auto	80	66	32767	LPT1			D
LPT5	Auto	80	66	32767	LPT1			D
LPT6	Auto	80	66	32767	LPT1			D
LPT7	Auto	80	66	32767	LPT1			D
LPT8	Auto	80	66	32767	LPT1			D
LPT9	Auto	80	66	32767	LPT1			D
LPT10	Auto	80	66	32767	LPT1			D
LPT11	Auto	80	66	32767	LPT1			D

Physical (D)evice, (E)ntire Connection, (F)ile. Use [PF2] for selection list

- From the **Edit** menu, choose **Natural Execution Configuration > Device/Report Assignments > Report Assignments**. A screen similar to the following appears:

Report Assignments		
Override default report number (MAINPR) 0		
Report	Device	Profile
0	VIDEO0	
1	LPT1	
2	LPT1	
3	LPT1	
4	LPT1	
5	LPT1	
6	LPT1	
7	LPT1	
8	LPT1	
9	LPT1	

See *Device/Report Assignments* in the *Configuration Utility* documentation for further information.

Defining Work Files

To download or upload from/to the PC, the corresponding work file must be defined as type `TRANSFER`. This can be done statically by an entry in the Configuration Utility or also during runtime with the `DEFINE WORK FILE` statement.

Definition in the Configuration Utility

From the **Edit** menu, choose **Natural Execution Configuration > Work Files**. You can define the corresponding work file number on the resulting screen. As work file type, `TRANSFER` must be defined.

If a download/upload takes place with the selected number of the work file, then the PC transfer is activated and an Entire Connection dialog box appears in which you specify the download destination on the PC or the file that is to be uploaded from the PC.

Definition during Runtime

If the work file shall get a new definition during runtime, then the Natural statement `DEFINE WORK FILE` must be defined before executing the download/upload sequence.

The work file type `TRANSFER` must be used. Example:

```
DEFINE WORK FILE 1 'PC' TYPE 'TRANSFER'
```

Defining Printers

To download data to a logical printer by writing reports, a static entry is required in the Configuration Utility which defines the setting of the logical printer.

The allocation of a report to this logical printer can then also be done during runtime with the `DEFINE PRINTER` statement.

Definition in the Configuration Utility

The following definition of the logical printer must always be done statically in the Configuration Utility in order to configure a printer that sends the output data to an Entire Connection terminal:

➤ to configure a printer:

- From the **Edit** menu, choose **Natural Execution Configuration > Device/Report Assignments > Device Assignments**. On the resulting screen, select the desired logical printer device (one

of LPT1 to LPT3) and set the device destination of the physical output device to "E" (send data to an Entire Connection terminal).

➤ **during runtime, a report is allocated to the previously defined logical printer device:**

- From the **Edit** menu, choose **Natural Execution Configuration > Device/Report Assignments > Report Assignments**. On the resulting screen, set the previously defined logical printer device (for example, LPT1) for the desired report number.

If a download takes place with the selected printer number, the PC transfer is activated and an Entire Connection dialog box appears in which you specify the download destination on the PC.

Definition during Runtime

The allocation of the desired report number to the previously defined logical printer device can also be done during runtime. The Natural statement `DEFINE PRINTER` must then be defined before executing the download sequence.

As an output parameter, the name of the previously defined logical printer device must be used (the following example uses the name "LPT1").

```
DEFINE PRINTER (7) OUTPUT 'LPT1'
```

Activating the PC Connection

To upload and download data to/from a PC, the PC connection must be active. You activate the PC connection from Linux.

➤ **To activate the PC connection**

- Use the Natural terminal command `%+`.

Or:

Invoke Natural with the dynamic parameter `PC=ON`.

Or:

Use the `SET CONTROL '+'` statement in a Natural program.

With the terminal command `%+`, you can also set an additional option:

Option	Description
%+N	The PC connection is activated. In addition, no field names are sent when downloading or uploading data.

If you attempt to upload or download data without the PC connection being activated, a message appears indicating that the PC connection is not active.

If you enter %+ and the PC connection is already active, a message appears indicating that the PC connection is already active.

Deactivating the PC Connection

➤ To deactivate the PC connection

- Use the Natural terminal command %-.

Or:

Use the `SET CONTROL ' - '` statement in a Natural program.

4 Natural Statements

This chapter lists the Natural statements required to process information between Linux and the PC by using Natural Connection.

For detailed information on these statements, refer to the relevant sections in the *Statements* documentation.

The Natural statements that apply to Natural Connection can be divided into the following groups:

Group	Statement
Transfer Data	DOWNLOAD PC FILE (synonym for WRITE WORK FILE)
	UPLOAD PC FILE (synonym for READ WORK FILE)
Download Reports	DISPLAY
	PRINT
	WRITE
Close a PC File	CLOSE PC FILE (synonym for CLOSE WORK FILE)

5

Processing Work Files and Nested Loops

■ Work File Format and Attributes	16
■ Maximum File Transfer Record Length for Natural Connection	16
■ Streaming	16
■ Dynamic Variables in READ WORK FILE	17
■ Nested READ Loops	17
■ Subsequent READ Loops	19

This chapter describes restrictions on the use of work file attributes, the support of work file formats and the impact of `READ` loops.

Work File Format and Attributes

Below are the restrictions that apply to the use of work file attributes:

- Accessing PC work files is restricted to a fixed record length of 1073741823 bytes or 32767 bytes when using the statement `WRITE WORK FILE VARIABLE`. Depending on the Entire Connection version installed on the PC, additional restrictions may apply as described below.

Maximum File Transfer Record Length for Natural Connection

The maximum record length supported for file transfer depends on the version of Entire Connection installed on the PC.

For Entire Connection up to Version 4.2, the maximum record length is limited by the number of bytes that can be displayed on the appropriate 3270 model. For example, for a 3270 Model 2 device the record length is $24 \times 80 = 1920$ bytes. Since all data buffers are enclosed by a header and trailer, the resulting net record length is 1887 bytes.

For Entire Connection Version 4.3.1, the maximum record length is limited to 32 KB - 1 byte = 32767 bytes.

As of Entire Connection Version 4.3.2 Fix Level 1 and Entire Screen Builder Version 5.2.1, the maximum record length is increased to 1 GB - 1 byte = 1073741823 bytes. But writing work files in variable format (`WRITE WORK VARIABLE`) is still restricted to a maximum record length of 32 KB - 1 byte.

Streaming

Entire Connection provides the option to transfer byte-streamed data that are non-record-oriented. A byte-streamed data transfer is activated when a `READ WORK FILE` or `WRITE WORK FILE` statement is coded with only one single operand of binary format.

Downloading and Uploading Binary Data

Binary data is usually object code or executable code that does not contain displayable or printable characters. To prevent standard character translations being performed during data transfer, Natural and Entire Connection use special methods for transferring binary data.

➤ To download binary data

- 1 Define a binary variable.
- 2 If the last block of downloaded data contains less data than the block size chosen, insert `X'FF'` at the position that marks the end of the binary data. (If you omit `X'FF'`, the rest of the last block will be filled with `X00`.)

➤ To upload binary data

- 1 Define a binary variable.
- 2 Remove `X'FF` from the last block. `X'FF` marks the end of the binary data.

Dynamic Variables in READ WORK FILE

If you define a dynamic variable of the format binary or alphanumeric as operand of a `READ WORK FILE` statement, when processing the corresponding `READ` loop, any resize operation on this variable will only be valid until the next `READ` is performed. While processing the `READ`, Natural resizes all dynamic variables to the size they had when the work file was opened. This is required in the open process which determines the record layout. The record layout is mandatory for processing the corresponding work file. The record layout is valid until the next close of the work file occurs.

Exception: An internal resize cannot be performed for inner loops if nested `READ` loops are processed on the same work file. See also the programming recommendations about nested loops below. If a dynamic variable of size 0 is used as the only operand of a `READ WORK FILE` statement, Natural issues the error NAT1500.

Nested READ Loops

Do not specify nested `READ` loops on one work file. The result of the inner loop(s) can be unpredictable if the operands of the inner loop do not correspond to the operands of the outer loop. The reason is that all records uploaded from the PC are processed in the format that was determined when the work file was opened in the outermost loop.

Below are example programs that demonstrate the unpredictable results the inner loop(s) of nested `READ` loops can have:

- [Example of Inner READ Loop](#)
- [Example of READ Loop and CALLNAT](#)

Example of Inner READ Loop

In the example program PCNESTED, during READ processing, another READ is performed:

```
/* PCNESTED
/*
DEFINE DATA LOCAL
  1 #REC1   (A) DYNAMIC
  1 #NUMBER (N10)
END-DEFINE
*
MOVE ALL 'TEST RECORD 1' TO #REC1 UNTIL 100
READ WORK FILE 1 #REC1
  READ WORK FILE 1 #NUMBER
  DISPLAY #NUMBER
END-WORK
END-WORK
END
```

Example of READ Loop and CALLNAT

In the example program PCMAIN and subprogram PCRSUB01, during READ loop processing, an external object is called:

```
/* PCMAIN
/*
DEFINE DATA
LOCAL
  1 RECL (A2000)
  1 REDEFINE RECL
    2 RECNR (N4)
  1 CO   (N4)
END-DEFINE
*
WRITE WORK 1 COMMAND
'SET PCFILE 2 UP DATA C:/TSTPCAM/PCMAIN.TXT'
READ WORK 2 RECL
  DISPLAY RECL (AL=72)
  CALLNAT 'PCRSUB01' RECL
END-WORK
END
```

Subprogram PCRSUB01:


```

/*Subprogram PCRSUB01
/*
DEFINE DATA
PARAMETER
  1 RECL (A2000)
LOCAL
  1 #CC1 (A20)
  1 #CC2 (N4)
*
END-DEFINE
READ WORK 2 RECL
  #CC1 #CC2
  DISPLAY #CC1 #CC2
END-WORK
END

```

Subsequent READ Loops

If a READ loop is terminated by a conditional `ESCAPE`, close the work file explicitly with the `CLOSE WORK FILE` statement so that the same work file can be processed in a subsequent READ in the same object.

Exception: You can omit the `CLOSE WORK FILE` if you need not read the file again from the beginning, and if the subsequent READ uses the same record layout as the preceding one.

Below is an example that demonstrates how to correctly code a program with two READ loops on one work file.

Example of Loop with ESCAPE and CLOSE

In the example program `PCESCAPE`, the work file is explicitly closed after the first READ loop has been terminated by `ESCAPE BOTTOM` so that the second READ loop must reopen the work file:

```

/*PCESCAPE
/*
DEFINE DATA
LOCAL
  1 #CC1          (A20)
  1 #CC2          (A40)
  1 #COUNTER      (I2)
*
END-DEFINE
READ WORK 2 #CC1
  DISPLAY #CC2
  ADD 1 TO #COUNTER
  IF #COUNTER GE 3
    ESCAPE BOTTOM

```

```
    END-IF  
END-WORK  
CLOSE WORK FILE 2  
*  
READ WORK 2 #CC2  
    DISPLAY #CC2  
END-WORK  
END
```