Entire Connection

Terminal Emulation

Version 9.3.4

September 2026

ADABAS & NATURAL

This document applies to Entire Connection Version 9.3.4 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 © 1984-2026 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: PCC-TERMINAL-934-20260915

Table of Contents

Preface vii
1 About this Documentation 1
Document Conventions 2
Online Information and Support 2
Data Protection 3
I 5
2 Getting Started 7
Starting the Terminal Application 8
Elements of the Application Window 9
Shortcut Keys 21
Using Help 21
Quitting the Terminal Application 22
Startup Parameters 22
3 Working With Host Sessions 25
Starting a Host Session 26
Working in a Terminal Emulation Screen 29
Transferring Data Using the Clipboard 30
Setting the Options 33
Modifying the Font for a Host Session 37
Modifying the Color Scheme for a Host Session 38
Modifying the Key Scheme for a Host Session 40
Modifying the P-Key Scheme for a BS2000 Host Session 42
Changing the Translation Settings for a Host Session 43
Displaying Security Information for an SSL or SSH Session 44
Printing the Contents of a Terminal Emulation Screen 46
Defining the Default Orientation for Printing 47
Locking the Current Session 48
Switching to a Session in Another Terminal Application Window 48
Switching to the Configuration Manager 49
Switching to the Host Printer Manager 50
Terminating a Host Session 50
4 Task List 51
Executing a Task 52
Terminating a Task 55
5 Unattended Workstation 57
Scheduling Tasks 58
Starting and Stopping UA Mode 62
II Data Transfer 63
6 General Data Transfer Information 65
Natural Data Transfer 66
What can be Transferred? 66
Format Files 68
Supported File Formats 69

Supported Field Formats ... 70
Aborting Data Transfer ... 71
Auditing Data Transfer ... 71
7 Specifying a File Name ... 73
Downloading Data Using a Dialog Box ... 74
Uploading Data Using a Dialog Box ... 79
Specifying a File Name Using the SET Command ... 80
Creating a File Name Dynamically ... 82
8 Uploading and Downloading Data ... 83
Uploading and Downloading Encrypted Files ... 84
Downloading Fixed-Length Data Records ... 84
Downloading Variable Data Records ... 84
Downloading Reports ... 85
Downloading Binary Data ... 85
Downloading Data to dBase or Lotus with Labels ... 86
Downloading Data to HTML ... 86
Downloading Data to XML ... 86
Downloading Data to Excel ... 95
Downloading Numeric Data to ASCII Format ... 95
General Upload Information ... 95
Uploading Binary Data ... 96
Uploading Data with Labels from dBase or Lotus ... 96
Uploading HTML Files ... 96
Uploading XML Files ... 97
Uploading Excel Files ... 98
Uploading ASCII Data ... 98
Restrictions for Unicode Data ... 98
9 Transferring Natural Objects ... 99
Transfer Tasks ... 100
Executing a Transfer Task ... 102
10 Sending Commands to the PC ... 103
Asynchronous Processing ... 104
Synchronous Processing ... 104
III Procedure Files ... 107
11 Procedure Files ... 109
What is a Procedure File? ... 110
Executing a Procedure File ... 110
External Parameters ... 111
Addressing a Host Session from a Procedure File ... 111
Aborting or Terminating a Procedure File ... 112
Creating a Procedure File Using Learn Mode ... 112
Debugging a Procedure File ... 113
Conventions ... 115
Command Categories ... 117
Variables in Procedure Files ... 121

IV Variables 123
12 Dynamic Variables 125
List of Dynamic Variables 126
13 Global Variables 133
Global Variables with Value Type Integer 134
Global Variables with Value Type String 137
14 Local Variables 139
Local Variables with Value Type Integer 140
Local Variables with Value Type String 143
15 System Variables 147
Types of System Variables 148
Value Types for System Variables 149
List of System Variables 149

Preface

The following topics apply to the terminal application of Entire Connection.

Getting Started
Working With Host Sessions
Task List
Unattended Workstation
Data Transfer
Procedure Files
Variables

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 *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	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.softwareag.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.

I

- 2 Getting Started 7
- 3 Working With Host Sessions 25
- 4 Task List 51
- 5 Unattended Workstation 57

2 Getting Started

- Starting the Terminal Application 8
- Elements of the Application Window 9
- Shortcut Keys 21
- Using Help 21
- Quitting the Terminal Application 22
- Startup Parameters 22

This chapter tells you how to get started with the terminal application of Entire Connection.

Starting the Terminal Application

If you do not specify otherwise during installation, an Entire Connection folder automatically appears in the **All Programs** folder of the Start menu after Entire Connection has been installed. It contains the shortcuts for the Entire Connection components, including the terminal application.

To start the terminal application

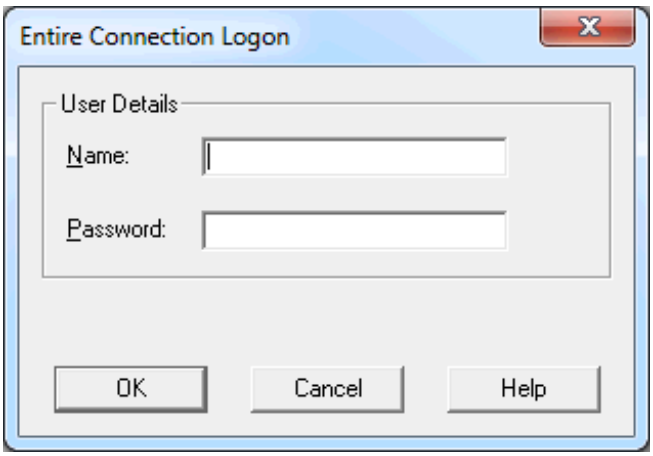
1 From the Start menu, choose **All Programs > Software AG Entire Connection** *n.n.n* **> Terminal**.

It is assumed that you log on as an administrator. Therefore, the following default settings are used:

- "SYSTEM" as the default name.
- "PASSWORD" as the default password.

It is recommended that you create a **shortcut** on your desktop.

If user information is required, the **Entire Connection Logon** dialog box appears and you must proceed as described below.



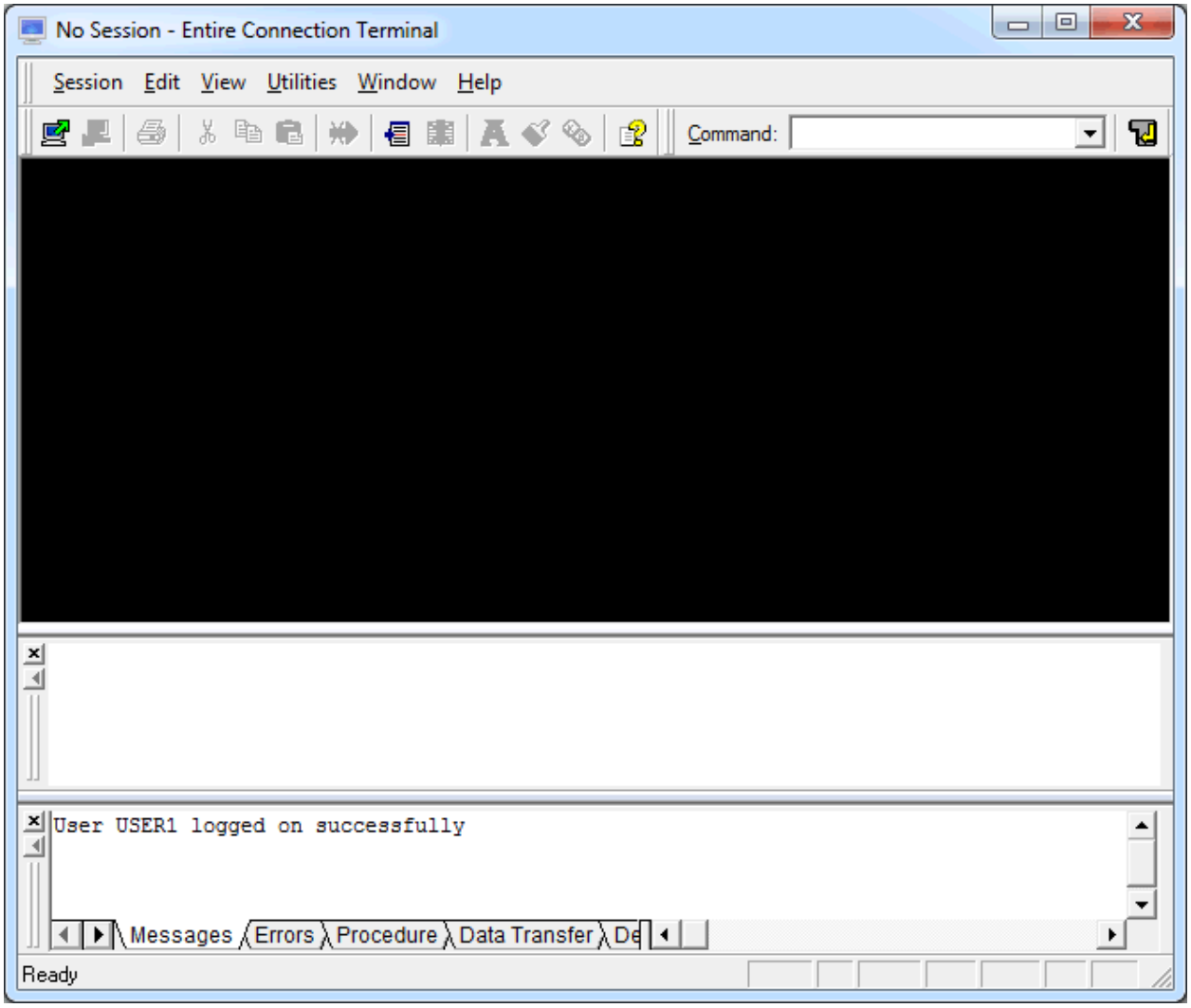
2 Enter the name that the administrator has assigned to you.

3 Enter your password (if required).

4 Choose the **OK** button.

The terminal application is now started with the rights that are defined for you.

Elements of the Application Window



When you start Entire Connection for the very first time, not all of the above elements are shown in the application window. These elements can be shown or hidden using the corresponding commands in the **View** menu.

The following topics are covered below:

- Title Bar
- Menu Bar
- Standard Toolbar
- Debug Toolbar
- Command Line
- Status Bar

- Terminal Emulation
- Output Window
- Input History
- Debug Windows
- Keypad

Title Bar

When a host session is active, its name is indicated in the title bar. Otherwise, "No Session" is shown.

Menu Bar

The following menus are available:

Session	Using the commands in this menu you can **start** or **end** a host session. You can also modify the **font**, **color** and **key scheme** used for the information shown in a host screen as well as **print** a host screen, or define BS2000 **P-keys**. In addition, you can change the **translation** settings for the current session, or display **security** information for the current SSL or SSH session.
Edit	Using the commands in this menu, you can **copy or cut** information from the host screen and paste it in a different location.
View	Using the commands in this menu, you can show or hide the various elements of the application window (such as the toolbar, keypad, output window or input history). When you exit the terminal application, these settings are stored in the Windows registry.
Utilities	Using the commands in this menu, you can invoke the **task list** or **unattended workstation**. You can also **cancel a data transfer**, **cancel a procedure file**, **lock the current session**, switch to the **Configuration Manager** or to the **Host Printer Manager**, or define the maximum number of lines in the **output window**.
Help	When you choose **Online Documentation** from this menu, you access the Entire Connection **documentation**. When you choose **About** from this menu, information about Entire Connection is shown. When you choose **Software AG on the Web** from this menu, you can check our web site for the latest information on products or events.

Note: The shortcut keys that are displayed next to a menu command are *not* available when the active key scheme uses them for different purposes.

Standard Toolbar

You can execute the most important Entire Connection functions using the standard toolbar.

Using the mouse, you can drag the standard toolbar to another position:

- for example, so that it is shown behind or below the command line, or to the left or right of the terminal emulation screen,
- so that it is shown in a window of its own:

You can move the window freely on your screen. You can move it back to the application window (e.g. back to its original position below the menu bar) so that it is no longer shown in a window. This process is called “docking”. To prevent docking, press CTRL while moving the window.

The toolbar buttons represent the following menu commands:

Open (**Session** menu)

Close (**Session** menu)

Print (**Session** menu)

Cut (**Edit** menu)

Copy (**Edit** menu)

Paste (**Edit** menu)

Cancel Transfer (**Utilities** menu)

Task List (**Utilities** menu)

Cancel Procedure (**Utilities** menu)

Font (**Session** menu)

Color (**Session** menu)

Keyboard (**Session** menu)

Online Documentation (**Help** menu)

To switch the standard toolbar display on and off

- From the **View** menu, choose **Toolbars > Standard**.

  When the standard toolbar is displayed in the application window, a check mark is shown next to the **Standard** command.

Debug Toolbar

You can execute an Entire Connection procedure file in debug mode using the buttons in the debug toolbar. This toolbar appears automatically when you start debugging. See *Debugging a Procedure File* for detailed information.

To switch the debug toolbar display on and off

- From the **View** menu, choose **Toolbars > Debug**.

  When the debug toolbar is displayed in the application window, a check mark is shown next to the **Debug** command.

Note: This command is only available as long as you are debugging a procedure file.

Command Line

You can execute the following directly from the command line:

- a command
- **a task**
- **a procedure file**

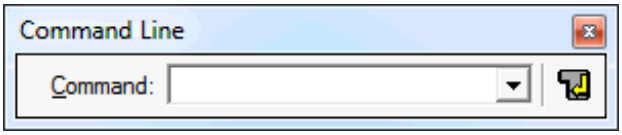
By default, the command line is shown to the right of the standard toolbar. The current working directory is now shown as tooltip in the text box of the command line.

Entire Connection saves each character string you enter in the command line. Each string can be up to 255 characters long. The drop-down list box contains your last 20 entries. You can select an entry and execute it once more.

When the mouse pointer is positioned on the command line, you can use the *right* mouse button or press SHIFT+F10 to invoke a context menu. Using the commands from this context menu, you can, for example, copy a text string to the command line or undo your last input.

Using the mouse, you can drag the command line to another position:

- for example, so that it is shown before or below the standard toolbar, or
- so that it is shown in a window of its own:



You can move the window freely on your screen. You can move it back to the application window (e.g. back to its original position below the menu bar) so that it is no longer shown in a window. To prevent docking, press CTRL while moving the window.

To switch the command line display on and off

- From the **View** menu, choose **Toolbars > Command Line** .

  When the command line is displayed in the application window, a check mark is shown next to the **Command Line** command.

To execute a command from the command line

1. Enter the command and, if applicable, all required parameters.

   For example:

   ```
   DISCONNECT
   ```

2. Press ENTER.

   Or:

   Choose the following button:

To execute a task from the command line

1. Enter the name of the task and, if applicable, all required parameters.

   For example, to execute a Natural source program (Natural must be active on the host):

```
DSOURCE RPT01 SAGXX File1
```

2 Press ENTER.

Or:

Choose the following button:

The program RPT01 in library SAGXX is downloaded to a PC file named "File1".

If you do not specify the parameters for DSOURCE in the command line, a dialog box appears and you must supply the required parameters (i.e. program, library and file name).

To execute a procedure file from the command line

1 Enter the command EXECUTE followed by a blank, the name of the procedure file and, if applicable, all required parameters.

For example:

```
EXECUTE LOGON
```

2 Press ENTER.

Or:

Choose the following button:

Status Bar

The status bar at the bottom of the application window is used to display system messages and help texts for the currently selected menu command or toolbar button.

To switch the status bar display on and off

- From the **View** menu, choose **Status Bar**.

  When the status bar is displayed in the application window, a check mark is shown next to this command.

Terminal Emulation

When communication with the host environment is established, terminal emulation is displayed in the application window. You can run several sessions with different hosts concurrently. For each session, a new terminal application window is invoked.

The following terminal types are emulated:

- 3270
- VT
- 9750

See *Working with Host Sessions* for further information.

Output Window

The output window is shown below the terminal emulation. It provides the following tabs:

- **Messages**
  Shows system messages (for example, whether logon was successful). For BS2000 communication, information from the system line is also shown here.
- **Errors**
  Shows all error messages that occur during the current terminal emulation session.
- **Procedure**
  Shows all messages that result from executing a procedure, except those that require user input.
- **Data Transfer**
  Shows all start, progress and end messages that result from transferring a file.
- **Debug Trace**
  Intended for debugging purposes under the supervision of your technical support.

When you select a tab, the corresponding pane is shown providing information on your recent actions. The appropriate pane is automatically shown, for example, when you execute a procedure file.

To switch the output window display on and off

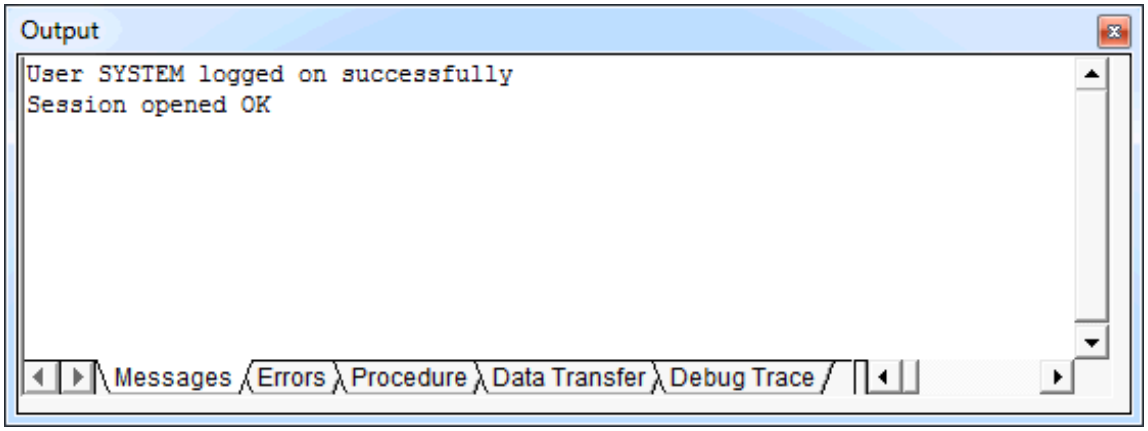
- From the **View** menu, choose **Output Window**.

  When the output window is displayed in the application window, a check mark is shown next to this command.

To modify the size of the output window, move the mouse pointer over the border between the output window and the window above it until the pointer changes, showing two arrows pointing

into opposite directions. Then drag the border using the mouse until the output window has the desired size.

Using the mouse, you can drag the output window to another position so that it is shown in a window of its own:



You can then move the output window freely on your screen. You can move it back to its original position in the application window. To prevent docking, press CTRL while moving the window.

When the mouse pointer is positioned on the output window, you can use the *right* mouse button or press SHIFT+F10 to invoke a context menu. This context menu provides the following commands:

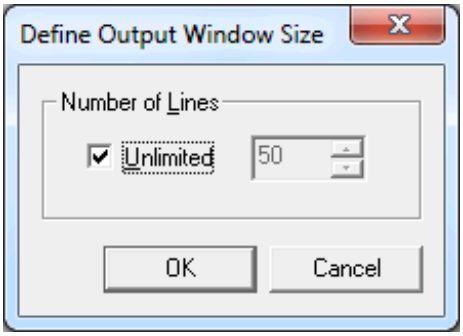
Copy	Copies the selected text to the Windows clipboard. If text has not been selected, the contents of the current pane is copied to the clipboard.
Clear	Clears the current pane so that no more text is shown.
Save to File	Writes the contents of the current pane to a file.
Select Pane	Enables keyboard users to go to a specific pane of the output window.
Enable Docking	When a check mark is shown next to this command, the output window can be docked at its original position in the application window. When a check mark is not shown, docking is not possible; a previously docked output window is shown in a window of its own.
Hide	Switches the output window display off. This corresponds to the **Output Window** command in the **View** menu.

You can define the maximum number of lines that are to be written to the output window. See below.

To define the maximum number of lines in the output window

1 From the **Utilities** menu, choose **Output Window Size**.

The following dialog box appears.



When the **Unlimited** check box is selected, the number of lines in the output window is not limited.

When the **Unlimited** check box is not selected, you can specify the maximum number of lines using the spin box.

2 Specify the required information in the dialog box.

3 Choose the **OK** button.

Input History

The window for the input history is shown below the terminal emulation. As soon as you press ENTER, your last character input is written to the input history. The input history contains your last 50 entries. Using the input history, you can execute a previously entered command once more or insert previously entered text in a field.

Identical input is only shown once in the input history. A triangle indicates the most often used entry.

The content of an invisible field (for example, a password) is not written to the input history. If you have entered information in more than one field, only the input in the current visible field is written to the input history. When you close the terminal application window, the input history is cleared.

This feature requires that the option **Enable recall feature and input history** has been enabled in the session properties of all required sessions. See the description of the **Terminal** property page in the *Overview of Object Properties* for further information. If it has not been enabled, it is only possible to open an input history file that was previously saved. You can then use the entries in this file to execute commands or insert text. New entries, however, are not written to the input history.

Note: This option is not available for communication with terminals of type VT.

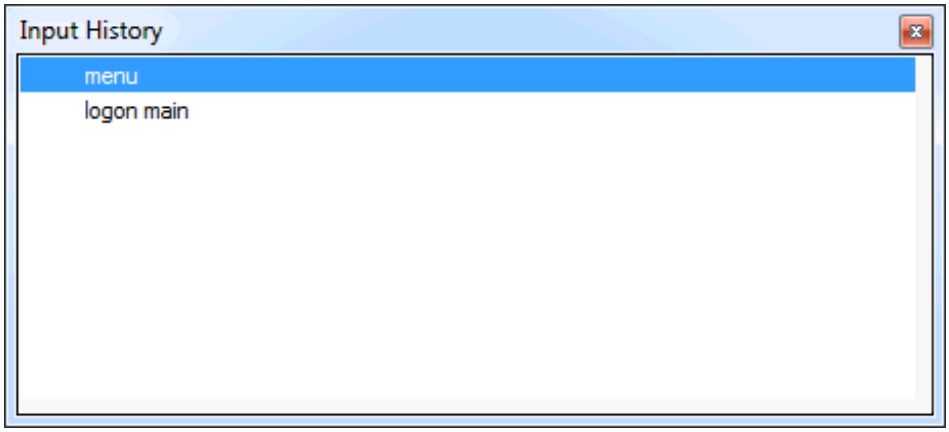
To switch the input history display on and off

- From the **View** menu, choose **Input History**.

When the input history is displayed in the application window, a check mark is shown next to this command.

To modify the size of the input history window, move the mouse pointer over the border between the input history window and the window above it until the pointer changes, showing two arrows pointing into opposite directions. Then drag the border using the mouse until the input history window has the desired size.

Using the mouse, you can drag the input history window to another position so that it is shown in a window of its own:



You can then move the input history window freely on your screen. You can move it back to its original position in the application window. To prevent docking, press CTRL while moving the window.

To execute an entry in the input history

- Double-click the desired entry.

  Or:

  Select the desired entry, click the right mouse button or press SHIFT+F10, and from the resulting context menu, choose **Execute**.

When the mouse pointer is positioned on the input history window, you can use the *right* mouse button or press SHIFT+F10 to invoke a context menu. This context menu provides the following commands:

Clear List	Clears the input history so that no more entries are shown.
Delete Entry	Deletes the selected entry from the input history.
Lock List	Locks the list so that new input is not written to the input history. This is helpful, if you have opened a file containing your own input history (see below). As long as the list is locked, the entries in the input history are shown with another color and certain commands in the context menu are not available.
Execute	Inserts the selected entry at the current position of the terminal emulation and sends it to the host.
Insert	Inserts the selected entry at the current position of the terminal emulation.
Sort by History	Displays the most recent entry at the top of the list.
Sort by Name	Displays the entries in alphabetical order.
Sort by Usage	Displays the most often used entries at the top of the list.
Save As	Writes the contents of the input history to a file. A dialog box appears prompting you to enter a file name.
Open	Opens an input history that was previously saved and thus overwrites the current entries. A dialog box appears prompting you to specify the name of desired file. You can also create your own file (for example, containing frequently-used commands) and then read this file into the input history window. Your file must be in ASCII format and each entry must be written to a line of its own. When the file is opened, all entries in this file are read into the input history. Exception: if the file contains blank lines, all entries after the first blank line are ignored.
Enable Docking	When a check mark is shown next to this command, the input history window can be docked at its original position in the application window. When a check mark is not shown, docking is not possible; a previously docked input history window is shown in a window of its own.
Hide	Switches the input history window display off. This corresponds to the **Input History** command in the **View** menu.

Debug Windows

When you start debugging a procedure file, two windows (a procedure window and a variable window) appear. See *Debugging a Procedure File* for detailed information.

To switch the procedure window display on and off

- From the **View** menu, choose **Debug Windows > Procedure**.

  When the procedure window is displayed in the application window, a check mark is shown next to the **Procedure** command.

To switch the variable window display on and off

- From the **View** menu, choose **Debug Windows > Variable**.

When the variable window is displayed in the application window, a check mark is shown next to the **Variable** command.

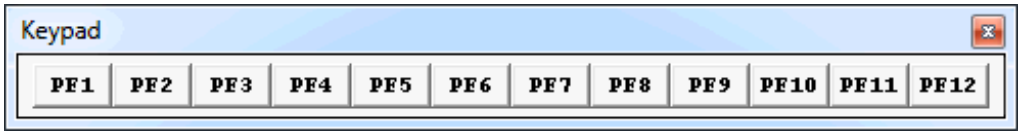
Note: These commands are only available as long as you are debugging a procedure file.

Keypad

A keypad contains buttons that represent host function keys. You can display different keypads.

Using the mouse, you can drag a keypad to another position:

- for example, so that it is shown below the toolbar, or to the left or right of the terminal emulation screen,
- so that it is shown in a window of its own:



You can move a keypad freely on your screen. You can move it to the application window (e.g. below the toolbar) so that it is no longer shown in a window. To prevent docking, press CTRL while moving the window.

To switch the keypad display on and off

- From the **View** menu, choose **Keypad** > *keypad-name*.

  When a keypad is displayed, a check mark is shown next to its name.

  Note: An option in the session properties defines whether the currently shown keypads are redisplayed the next time you open a session. For further information, see the description of the **General** page of the **Session Properties** dialog box in the *Overview of Object Properties*.

Shortcut Keys

When working with the keyboard, you can use the following shortcut keys:

Shortcut Key	Function
CTRL+SHIFT+1	Activate main window (terminal emulation).
CTRL+SHIFT+2	Activate window for input history.
CTRL+SHIFT+3	Activate output window.
CTRL+SHIFT+4	Activate procedure window (debug mode).
CTRL+SHIFT+5	Activate variable window (debug mode).
SHIFT+F10	Open context menu.

Using Help

The complete Entire Connection documentation is available as an HTML help file.

To access the documentation

- From the **Help** menu, choose **Online Documentation**.

  Or:

  Choose the following toolbar button:

To invoke context-sensitive help

- In a dialog box, choose the **Help** button.

Quitting the Terminal Application

When you quit the terminal application, the following is automatically executed:

- If a host session is active for which an automatic logoff procedure has been defined, the procedure is processed. The session is then closed (i.e. it is disconnected from the host).
- If a host session is active for which an automatic logoff procedure was not defined, the session is immediately closed.
- If a shutdown task has been defined, it is processed when the last terminal application is closed.

Once the above conditions have been checked and processed, the terminal application is closed. The settings in the **View** menu as well as position and size of the application window are stored in the Windows Registry. The next time, you start the terminal application, its application window is restored as it appeared when you last used it.

To quit the terminal application

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

  Or:

  Choose **Close** from the Control menu.

  Or:

  Click the corresponding standard button in the title bar.

To quit all terminal applications at once

- From the **Session** menu, choose **Exit All Terminals**.

Startup Parameters

The terminal application is started by invoking *Pcterminal.exe*.

If you create a shortcut for *Pcterminal.exe*, you can modify its properties in order to define the parameters with which the terminal application is to be started.

Note: This information applies only to shortcuts on your desktop (it does not apply to the shortcut in the Start menu or a copy of it).

The syntax for starting the terminal application is the following:

```
[drive:\path\]Pcctерminal.exe username password a b c d e f g h /NOSPLASH ↩
/SF:path-to-sharefile
```

You can enter up to 192 characters for the parameters.

User name and password are optional parameters. If a parameter is not required, you must specify an asterisk (*).

The optional parameters `a` through `h` stand for the following (you cannot specify an asterisk here):

Parameter	Description
`a`	A task or procedure name.
`b` through `h`	Seven parameters that can be passed to the task or procedure specified in parameter `a`.

Examples:

- Invoke the terminal application, prompt for user ID and password and then open the default session:

  ```
  Pcctерminal.exe * * te
  ```

- Invoke the terminal application, prompt for password, if required, and then open the default session:

  ```
  Pcctерminal.exe MYNAME * te
  ```

When you specify the option `/NOSPLASH`, Entire Connection's startup screen does not appear.

When you specify the option `/SF: path-to-sharefile`, you define the share file that is to be used. If the path to the share file contains blanks, you must put the path in quotes. Example:

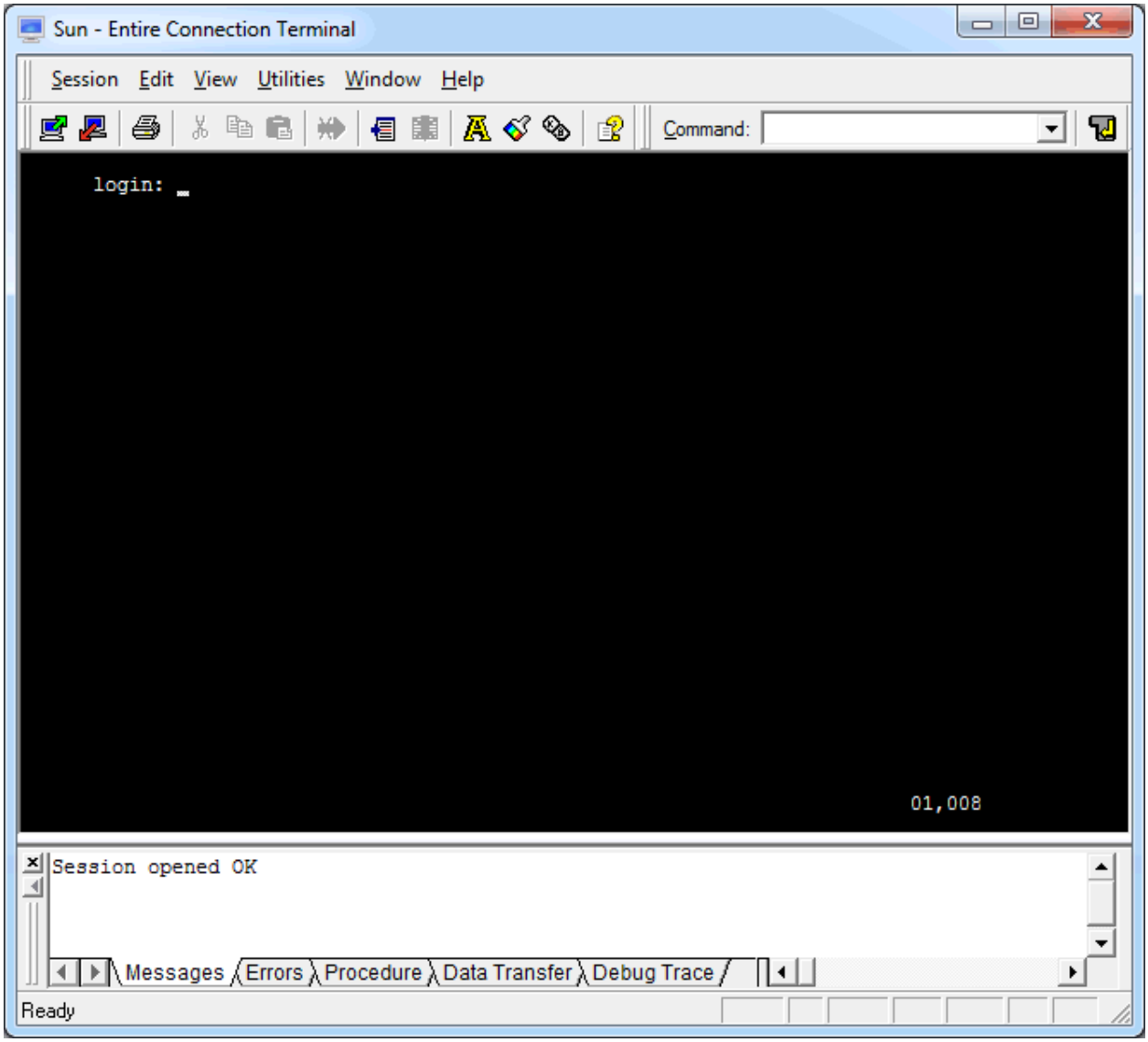
```
Pcctерminal.exe /SF:"C:\My Files\mysharefile.sag"
```

3 Working With Host Sessions

- Starting a Host Session 26
- Working in a Terminal Emulation Screen 29
- Transferring Data Using the Clipboard 30
- Setting the Options 33
- Modifying the Font for a Host Session 37
- Modifying the Color Scheme for a Host Session 38
- Modifying the Key Scheme for a Host Session 40
- Modifying the P-Key Scheme for a BS2000 Host Session 42
- Changing the Translation Settings for a Host Session 43
- Displaying Security Information for an SSL or SSH Session 44
- Printing the Contents of a Terminal Emulation Screen 46
- Defining the Default Orientation for Printing 47
- Locking the Current Session 48
- Switching to a Session in Another Terminal Application Window 48
- Switching to the Configuration Manager 49
- Switching to the Host Printer Manager 50
- Terminating a Host Session 50

Starting a Host Session

When you start a host session, terminal emulation is activated.



You can work with several host sessions at the same time. For one communication method, several sessions can be open at the same time. Each host session is shown in a separate terminal application window.

So that communication is successfully established with the host environment, all required communication parameters must be specified for the required session type. During communication with the host, the definitions in the session properties are used.

Messages from the host are shown in the status line of the terminal emulation screen (this is not the status line of the application window). The status line is located below the last line of a screen. For example, the status line for terminal model 2 is located in the 25th line.

Entire Connection uses the right corner of the status line in the terminal emulation screen to display the following information:

General

- **Cursor position**
  Example: the value 23,011 indicates that the cursor is located in row 23 and column 11.
- **Learn mode**
  If **learn mode** is active, an L is displayed in column 76. Learn mode is a trace option.
- **Trace status**
  If a trace command is active, a character indicating the corresponding trace command is displayed: B for `REC_BUFF`, S for `REC_SCR`, X for `REC_XFER`, and M for an active communication trace.
- **Procedure file processing**
  The character P in column 78 indicates that a procedure file is currently being processed for this session.
- **Insert mode**
  ^ in column 76 indicates that the insert mode is switched on.
- **Reveal mode**
  Shows field attribute information with the `REVEAL` command starting at column 38.

TN3270 Sessions

- **TN3270 session**
  If it is a TN3270 session, 4N is shown at the beginning of row 23.
- **Keyboard is locked**
  X() in row 23 indicates that the keyboard is locked.

VT Sessions

- **Debug trace is active**
  D indicates that the debug trace is active.
- **Uploading or downloading data**
  F indicates that Entire Connection is **uploading or downloading data**.
- **Keyboard is locked**
  X indicates that the keyboard is locked.

BS2000 Sessions

- **Send request**
  SAN indicates that the emulation sends a request. If request is not handled properly, the SAN is shown permanently.
- **Connection active**
  LTG indicates that the connection is active.
- **Keyboard detected**
  TAST indicates that the keyboard is usable.
- **Data exchange with BS2000**
  DIA indicates that screen data is transferred.
- **BS2000 trace active**
  Special BS2000 communication trace is active with **Option 2** active (*BS2.TRC*).

To start a host session

1 From the **Session** menu, choose **Open**.

Or:

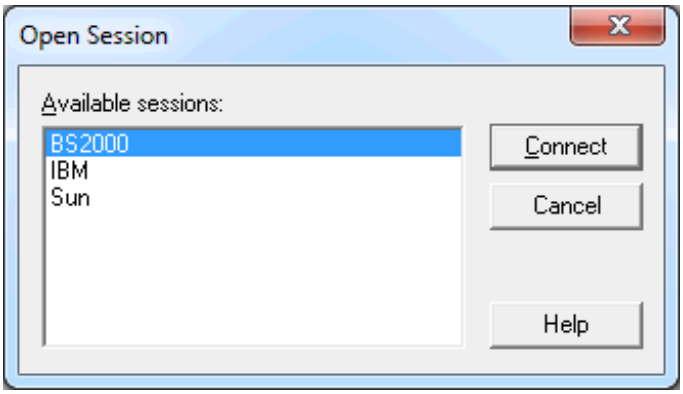
Press CTRL+O.

Or:

Choose the following toolbar button:

If only one session has been defined, this session is immediately started.

If more than one session has been defined, the **Open Session** dialog box appears and you must proceed as described below.



2 In the **Open Session** dialog box, select a session.

3 Choose the **Connect** button.

Notes:

1. When you open a session for which SSL has been enabled, your administrator may have defined that a dialog box is to appear in which you have to enter the password for the private key.
2. When you open a session for which SSH has been enabled, a dialog box appears in which you have to enter a user ID and password. In the communication parameters for a Telnet VTxxx session, on the **Security** page, you can configure the session in such a way that it uses the user ID and password from your user properties.
3. In the user properties, you can specify a default session which is automatically selected in the **Open Session** dialog box.

Working in a Terminal Emulation Screen

In a terminal emulation screen, you can use both mouse and keyboard. To change the cursor position, use the standard keys (for example, TAB or the arrow keys) or the mouse. To use a host function key, press the terminal emulation key assigned to this function key. To simulate the ENTER key, double-click any position in the terminal emulation screen. Otherwise, you can use your keyboard to enter data as normal.

Key Schemes

The key combinations that you can use (for example, for editing text) depend on the key scheme that is currently defined. See *Modifying the Key Scheme for a Host Session*.

The following functions are hardcoded for the following key combinations and will always be used if these key combinations are not defined otherwise in the current key scheme:

Key Combination	Description
CTRL+RIGHT-ARROW	Jump to the next word.
CTRL+LEFT-ARROW	Jump to the previous word.
CTRL+END	Jump to the end of the field.

Note: The shortcut keys that are displayed next to a menu command (e.g. CTRL+V for pasting text) are not available when the active key scheme uses them for different purposes.

Transferring Data Using the Clipboard

Using the commands from the **Edit** menu, you can copy or cut a terminal emulation screen (or part of it) and paste it somewhere else (for example, in another terminal emulation screen or in a PC application such as Microsoft Word). You can also copy or cut text from a PC application and paste it in a terminal emulation screen. Cut or copied data is stored on the Windows clipboard.

You must first select the desired text before you can use the **Cut**, **Copy** or **Append Copy** command.

The following rules apply for a terminal emulation screen:

- You can only cut text from unprotected fields. If protected fields have been selected, they are copied.
- When you use the **Cut** or **Copy** command, the content of the clipboard is deleted and replaced with the new text. If you want to add data to the clipboard, you must use the **Append Copy** command.
- When you use the **Append Copy** command, the content of the clipboard is not deleted. The selected text is inserted behind already existing text on the clipboard. When you then choose the **Paste** command, the whole content of the clipboard is inserted at cursor position.
- You can only paste text into unprotected fields. If the field is not long enough, it is filled up to its maximum size and the remaining text is inserted in the next unprotected field. Example: the text to be pasted is "DISPLAY" and the cursor is located in a two character long input field. In this case, only the first two characters ("DI") are pasted into the input field.

Unless indicated otherwise, the left mouse button is always used. The *right* mouse button is used to select the **Edit** menu commands from a context menu.

To select part of a terminal emulation screen (mouse usage)

1. Move the mouse pointer to the beginning of the text you want to select.
2. Press and hold down the left mouse button.

   This clears any previous selection.
3. Drag the mouse until all desired text is selected.
4. Release the mouse button.

   You can now cut, copy or append the selected text.

To select part of a terminal emulation screen (keyboard usage)

1. Use the arrow keys to move the cursor to the beginning of the text you want to select.

This clears any previous selection.

2 Press and hold down SHIFT.

3 Use the arrow keys to select the desired text.

4 Release SHIFT.

You can now cut, copy or append the selected text.

Note: When holding down CTRL+SHIFT and then using the arrow keys, all text to the end of the line (with RIGHT-ARROW) or to the bottom of the screen (with DOWN-ARROW) is selected.

To select the entire content of the terminal emulation screen

- From the **Edit** menu, choose **Select All**.

  Or:

  Press CTRL+A.

  A box outline is shown around the terminal emulation screen.

  You can now cut, copy or append the selected text.

To cancel the selection

- Click another screen position.

  Or:

  When using the keyboard, just select another part of the terminal emulation screen as described above.

  The existing selection is canceled.

To delete the selection

- From the **Edit** menu, choose **Clear**.

  Or:

  Press DEL.

  The selection is deleted (except protected fields). It is not transferred to the clipboard.

To copy the selection and append it behind existing text in the clipboard

- From the **Edit** menu, choose **Append Copy**.

  The selection is copied to the clipboard and appended behind already existing text.

To cut the selection and transfer it to the clipboard

- From the **Edit** menu, choose **Cut**.

  Or:

  Press CTRL+X.

  Or:

  Choose the following toolbar button:

  The selection is deleted and transferred to the clipboard. Exception: protected fields are not deleted; they are copied.

  Note: With VTxxx terminals it is not possible to cut text.

To copy the selection to the clipboard

- From the **Edit** menu, choose **Copy**.

  Or:

  Press CTRL+C.

  Or:

  Choose the following toolbar button:

To paste the content of the clipboard into a terminal emulation screen

1. Move the cursor to the position at which the text is to be inserted.
2. From the **Edit** menu, choose **Paste**.

Or:

Press CTRL+V.

Or:

Choose the following toolbar button:

The content of the clipboard is inserted at cursor position.

Setting the Options

Using the commands in the **Edit > Options** menu, you can set several options. These options are described below.

- Keep Insert Mode
- Block Mode Paste
- Field Mode Copy/Paste
- Single Line Paste (VT)
- Right Button Terminal
- Enable Mouse for Standard PF Keys
- Crosshair Cursor

For each session, the current settings of the options in the **Edit > Options** menu are stored when the session is closed. They are used again when the same session (with the same name) is opened once more.

Keep Insert Mode

This feature is only available for TN3270 and BS2000 sessions.

TN3270 and BS2000 terminals are normally in overwrite mode. If you want to insert characters, you have to press the INS key to activate insert mode. The insert mode is always reset to overwrite mode when you press ENTER, a PF key or any other key that sends the screen to the host.

However, when you enable **Keep Insert Mode**, the terminal is no longer reset to overwrite mode after a screen has been sent to the host.

To enable insert mode permanently

- From the **Edit > Options** menu, choose **Keep Insert Mode**.

When insert mode is enabled, a check mark is shown next to the **Keep Insert Mode** command.

Block Mode Paste

This mode is helpful when working with editors. When you select a block of text and paste it in the middle of a line, the block remains intact. The lines after the first line begin in the same column as the first line.

When block mode paste is not enabled for a TN3270 or BS2000 session, the lines after the first line start at the beginning of a line. When field mode copy/paste is enabled for these session types, block mode paste is ignored.

To enable block mode paste

- From the **Edit > Options** menu, choose **Block Mode Paste**.

  When block mode paste is enabled, a check mark is shown next to the **Block Mode Paste** command.

Field Mode Copy/Paste

This feature is only available for TN3270 and BS2000 sessions.

This mode is helpful if you want to copy the contents of several input fields in a screen to another set of input fields which has the same structure. During the paste operation, the target input fields (which can also be on a different screen) are filled field by field as they were copied from the source fields. Any non-input fields (protected fields and screen areas) that are part of your selection are ignored during the paste operation.

To enable field mode copy/paste

- From the **Edit > Options** menu, choose **Field Mode Copy/Paste**.

  When this kind of copy/paste mode is enabled, a check mark is shown next to the **Field Mode Copy/Paste** command.

To copy and paste the contents of input fields

1. Select the part of the terminal emulation screen from which you want to copy the input fields (see *Transferring Data Using the Clipboard*).
2. From the **Edit** menu, choose **Copy**.
3. Go to the screen on which you want to paste the copied input fields.
4. Select the first input field in which you want to start the paste operation.
5. From the **Edit** menu, choose **Paste**.

All target input fields are filled field by field with the contents of the source fields.

Single Line Paste (VT)

This feature is only available for VT sessions.

In single line paste mode, only the first line of the text from the clipboard is pasted. Pasting starts at the current cursor position and stops at the end of the line; further text in the line which is pasted from the clipboard is ignored. This feature is helpful when working outside of Natural, for example, on the Linux command line.

When block mode paste is enabled for a VT session, single line paste is ignored in this session.

To enable single line paste mode

- From the **Edit > Options** menu, choose **Single Line Paste (VT)**.

  When single line paste is enabled, a check mark is shown next to the **Single Line Paste (VT)** command.

Right Button Terminal

Within the same terminal emulation screen, it is possible to copy alphanumeric data into an input field using the mouse. To use this feature, you have to enable the right mouse button for terminal mode.

The data to be copied must not contain blanks and it must not contain non-alphanumeric characters. The copied data is always inserted at the current cursor position which may also be in the middle of an input field. If the input field is not long enough, the data are automatically truncated. Example: the copied data is "Display" and the cursor has previously been positioned at the beginning of a two character long input field. In this case, only the first two characters ("Di") are inserted in this field.

When you click on a period (.), only the period is copied to the input field.

Note: The context menu containing the edit commands is only shown when the right mouse button has not been enabled for terminal mode.

To enable the right mouse button for terminal mode

- From the **Edit > Options** menu, choose **Right Button Terminal**.

  When the right mouse button is enabled for terminal mode, a check mark is shown next to the **Right Button Terminal** command.

To copy data to an input field (single click)

1 Move the cursor to an unprotected field in which you want to enter data.
2 Click the required string with the right mouse button.

The string is copied to the input field.

To copy data to an input field and send it to the host application (double-click)

1 Move the cursor to an unprotected field in which you want to enter data.
2 Double-click the required string with the right mouse button.

The string is copied to the input field and is then immediately sent to the host application.

Enable Mouse for Standard PF Keys

It is possible to simulate a function key (PF key or PA key) with the mouse: you double-click the name of a defined function key on the screen in order to send the corresponding key code to the host. To use this feature, you have to enable the mouse for standard PF keys.

To enable the mouse for standard PF keys

- From the **Edit > Options** menu, choose **Enable Mouse for Standard PF Keys**.

  When the mouse is enabled for the standard PF keys, a check mark is shown next to the **Enable Mouse for Standard PF Keys** command.

Crosshair Cursor

This feature is only available for TN3270 and BS2000 sessions.

A crosshair cursor consists of two thin lines, one horizontal and one vertical, meeting at the lower left corner of the cursor. These lines help to see what is contained in the same line and column.

The crosshair cursor is displayed in addition to the regular cursor; it does not replace the regular cursor.

When the crosshair cursor is displayed, you can additionally define whether only the horizontal or the vertical line is to be shown, and you can fix the crosshair cursor at the current position.

Note: The color of the crosshair cursor can be changed in the **color scheme**.

To display the crosshair cursor

- From the **Edit > Options** menu, choose **Crosshair Cursor > Display Crosshair Cursor**.

  When the crosshair cursor is displayed, a check mark is shown next to the **Display Crosshair Cursor** command.

To display only the horizontal or vertical line of the crosshair cursor

- From the **Edit > Options** menu, choose **Crosshair Cursor > Only Horizontal**.

  Or:

  From the **Edit > Options** menu, choose **Crosshair Cursor > Only Vertical**.

  Only one of these two commands can be active at one time. Choosing **Only Horizontal** automatically disables **Only Vertical**, and vice versa.

  When only the horizontal or vertical line is displayed, a check mark is shown next to the corresponding command.

To fix the crosshair cursor at the current position

- From the **Edit > Options** menu, choose **Crosshair Cursor > Fix at Position**.

  When the crosshair cursor is fixed, a check mark is shown next to the **Fix at Position** command.

Modifying the Font for a Host Session

You can define another font to be used for the current session directly from the terminal application. Your new font definition is written to the share file. It is not necessary to invoke the Configuration Manager for this purpose.

Note: When you (the general user) modify the font of a public session, a private session is automatically created. This private session has the same properties as the public session. From now on, you continue working with the new private session.

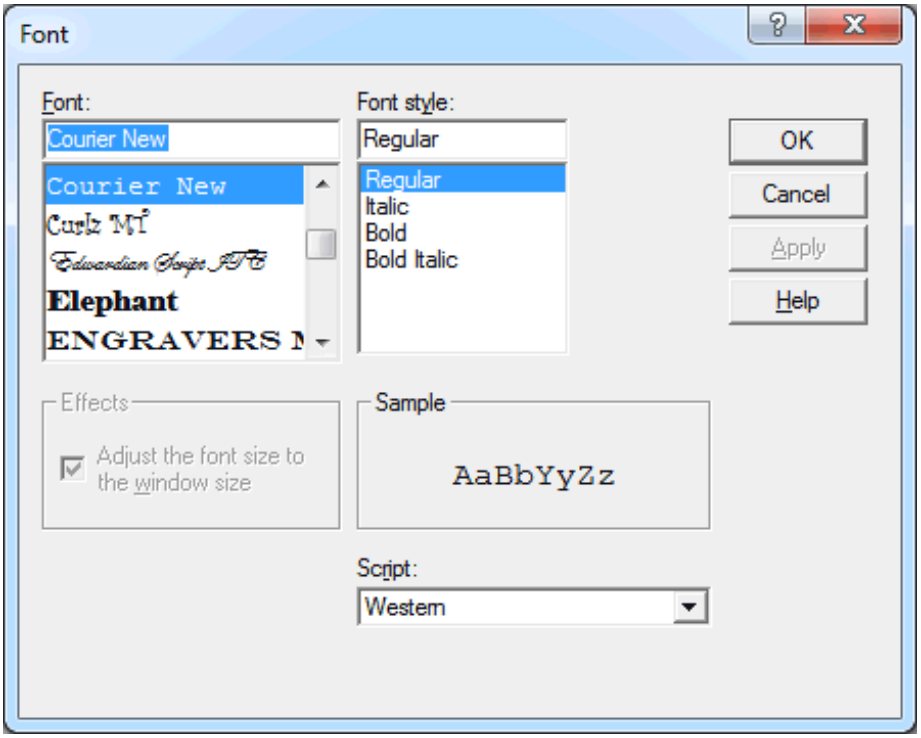
To modify the font

- From the **Session** menu, choose **Font**.

  Or:

  Choose the following toolbar button:

The **Font** dialog box appears:



See the *Overview of Object Properties* for information on this dialog box.

Modifying the Color Scheme for a Host Session

You can define another color scheme to be used for the current session directly from the terminal application. Your modifications are written to the share file. It is not necessary to invoke the Configuration Manager for this purpose.

Note: When you (the general user) modify the color scheme of a public session, a private session is automatically created. This private session has the same properties as the public session. From now on, you continue working with the new private session.

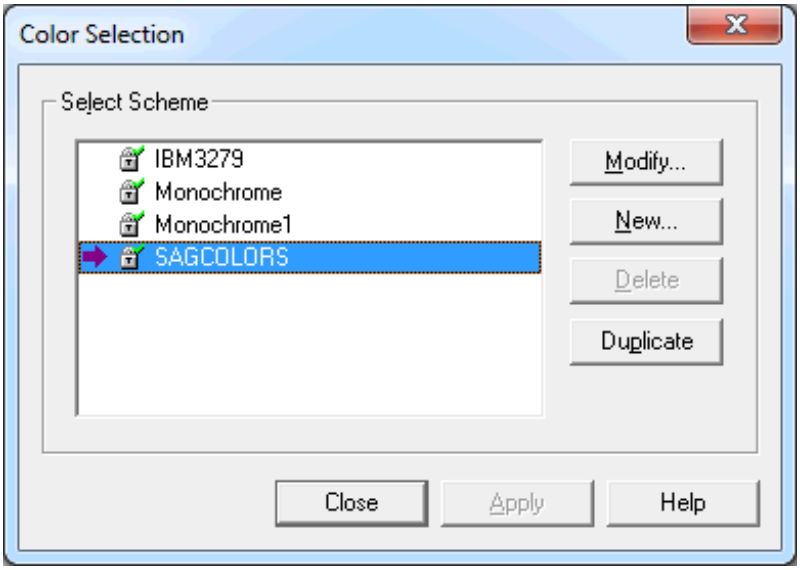
To activate another color scheme

1 From the **Session** menu, choose **Color**.

Or:

Choose the following toolbar button:

The **Color Selection** dialog box appears:



The active color scheme is indicated by an arrow. When the lock symbol next to the color scheme name shows a green check mark, the color scheme can be modified by a general user. When it shows a red cross, it cannot be modified by a general user; it can only be duplicated in this case.

2 Select the color scheme you want to activate.

3 Choose the **Apply** button to save your changes.

4 To close the dialog box, choose the **Close** button.

Command Buttons

Besides activating another color scheme, you can also create, modify, delete or duplicate a color scheme directly from the terminal application. To do so, use one of the following command buttons in the **Color Selection** dialog box:

Modify	Modifies the selected color scheme. You can only modify the color scheme which is currently active. If you select another color scheme, you are asked whether you want to make the selected color scheme active in order to continue.
New	Creates a new color scheme. Specify a name for the new color scheme in the resulting dialog box and choose the **OK** button. You are then asked whether you want to make the new color scheme active in order to modify it.
Delete	Deletes the selected color scheme. You are asked to confirm the deletion. Any session using this scheme will need to be updated manually. **SAGCOLORS** or **VTColors** cannot be deleted.

Duplicate	Creates a copy of the selected color scheme. Specify a name for the copy in the resulting dialog box and choose the **OK** button. You are then asked whether you want to make the copy active in order to modify it.

For detailed information on the dialog box which appears when you create, modify or duplicate a color scheme, see the *Overview of Object Properties*.

Modifying the Key Scheme for a Host Session

You can define another key scheme to be used for the current session directly from the terminal application. Your modifications are written to the share file. It is not necessary to invoke the Configuration Manager for this purpose.

Note: When you (the general user) modify the key scheme of a public session, a private session is automatically created. This private session has the same properties as the public session. From now on, you continue working with the new private session.

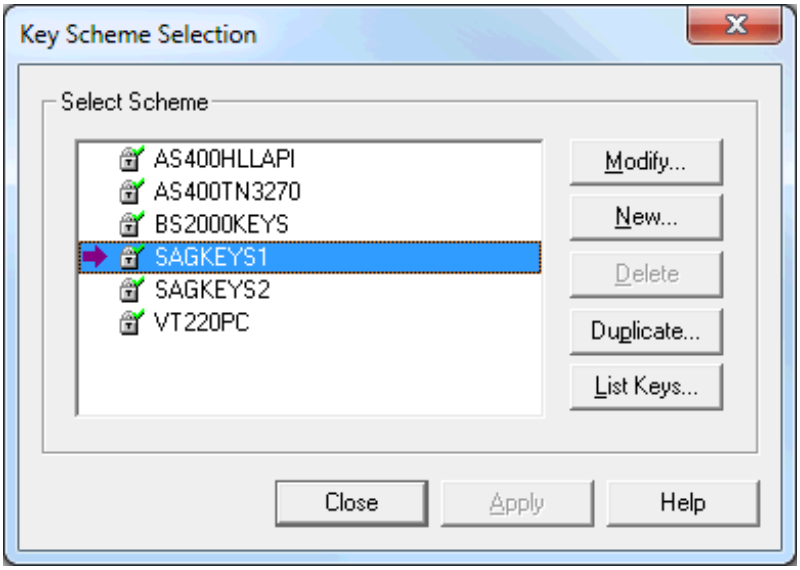
To activate another key scheme

1 From the **Session** menu, choose **Keyboard**.

Or:

Choose the following toolbar button:

The **Key Scheme Selection** dialog box appears:



The dialog box lists the available key schemes for all session types. Make sure to select a key scheme which makes sense with the currently connected session type.

The active key scheme is indicated by an arrow. When the lock symbol next to the key scheme name shows a green check mark, the key scheme can be modified by a general user. When it shows a red cross, it cannot be modified by a general user; it can only be duplicated in this case.

2 Select the key scheme you want to activate.

3 Choose the **Apply** button to save your changes.

4 To close the dialog box, choose the **Close** button.

Command Buttons

Besides activating another key scheme, you can also create, modify, delete or duplicate a key scheme directly from the terminal application. To do so, use one of the following command buttons in the **Key Scheme Selection** dialog box:

Modify	Modifies the selected key scheme. You can only modify the key scheme which is currently active. If you select another key scheme, you are asked whether you want to make the selected key scheme active in order to continue.
New	Creates a new key scheme. Specify a name for the new key scheme in the resulting dialog box, make all required modifications and choose the **OK** button. You are then asked whether you want to make the new key scheme active.
Delete	Deletes the selected key scheme. You are asked to confirm the deletion. Any session using this scheme will need to be updated manually. **SAGKEYS1** cannot be deleted.
Duplicate	Creates a copy of the selected key scheme. Specify a name for the copy in the resulting dialog box, make all required modifications and choose the **OK** button. You are then asked whether you want to make the copy active.

List Keys	Lists the currently defined keys for the selected key scheme. It is possible to save the current key scheme to a file and to print the current key scheme.

For detailed information on the dialog box which appears when you create, modify or duplicate a key scheme, or when you list the currently defined keys, see the *Overview of Object Properties*.

Modifying the P-Key Scheme for a BS2000 Host Session

P-keys (programmable keys) are only available for sessions of type BS2000 TCP/IP.

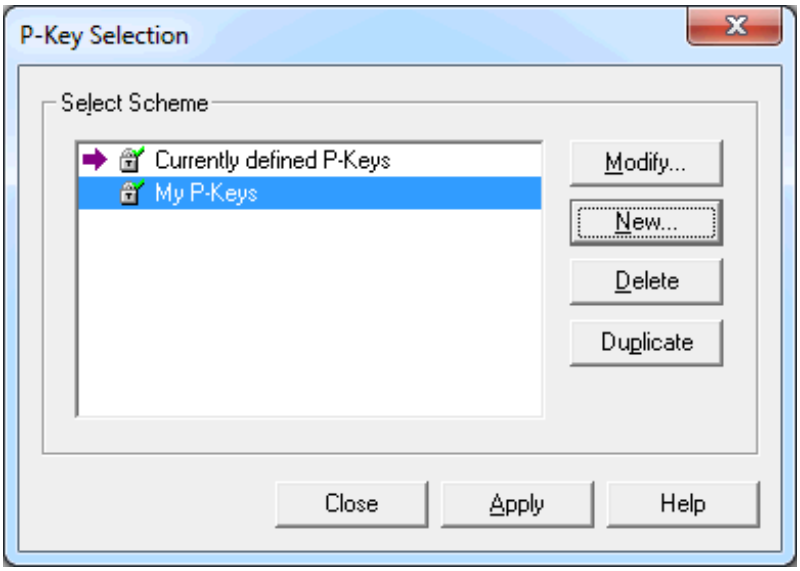
You can define another P-key scheme to be used for the current BS2000 session directly from the terminal application. Your modifications are written to the share file. It is not necessary to invoke the Configuration Manager for this purpose.

Note: When you (the general user) modify the P-key scheme of a public BS2000 session, a private session is automatically created. This private session has the same properties as the public session. From now on, you continue working with the new private session.

To activate another P-key scheme

1 From the **Session** menu, choose **P-Key**.

The **P-Key Selection** dialog box appears:



The P-key scheme with the name **Currently defined P-Keys** is always available. Its key definitions may change dynamically, depending on the application that is active in the current

session. When you modify this P-key scheme, your modifications may be lost when you change to another application.

The active P-key scheme is indicated by an arrow. When opening the **P-Key Selection** dialog box, the arrow always points to the P-key scheme which has been assigned in the Configuration Manager, or to the P-key scheme with the name **Currently defined P-Keys** if a P-key scheme has not been assigned in the Configuration Manager.

When the lock symbol next to the P-key scheme name shows a green check mark, the P-key scheme can be modified by a general user. When it shows a red cross, it cannot be modified by a general user; it can only be duplicated in this case.

2 Select the P-key scheme you want to activate.

3 Choose the **Apply** button to save your changes.

4 To close the dialog box, choose the **Close** button.

Command Buttons

Besides activating another P-key scheme, you can also create, modify, delete or duplicate a P-key scheme directly from the terminal application. To do so, use one of the following command buttons in the **P-Key Selection** dialog box:

Modify	Displays/modifies the selected P-key scheme.
New	Creates a new P-key scheme. Specify a name for the new P-key scheme (can be up to 16 characters long) in the resulting dialog box and choose the **OK** button.
Delete	Deletes the selected P-key scheme. You are asked to confirm the deletion. Any session using this scheme will need to be updated manually. The P-key scheme with the name **Currently defined P-Keys** cannot be deleted.
Duplicate	Creates a copy of the selected P-key scheme. Specify a name for the copy in the resulting dialog box and choose the **OK** button.

For detailed information on the dialog box which appears when you create, modify or duplicate a P-key scheme, see the *Overview of Object Properties*.

Changing the Translation Settings for a Host Session

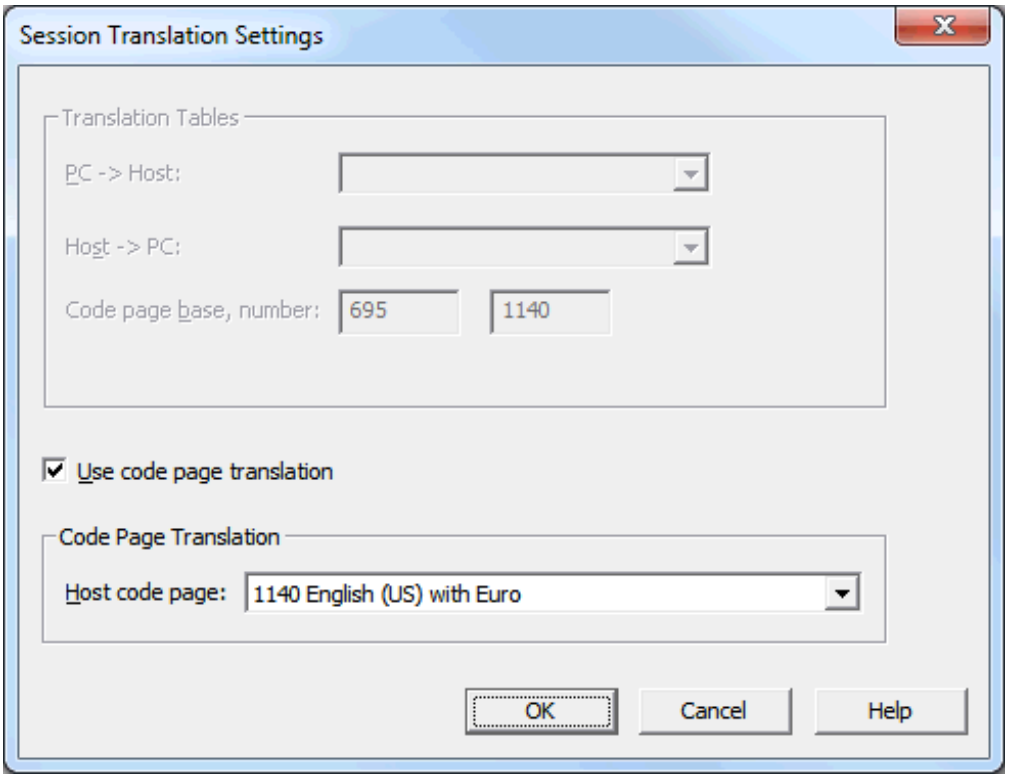
You can change the host code page or the translation tables for the current session. The new translation settings become active for the next screen that is received from the host.

Note: Code page translation is only available for the session type TN3270(E).

To change the translation settings

1 From the **Session** menu, choose **Translation**.

The **Session Translation Settings** dialog box appears.



2 Select the appropriate host code page or translation tables.

For detailed information on the options in this dialog box, see the description of the **National** property page of the session properties in the *Overview of Object Properties*.

3 Choose the **OK** button to activate the new settings.

Displaying Security Information for an SSL or SSH Session

Only available for sessions of type TN3270 for which SSL has been enabled, and for sessions of type Telnet VTxxx for which SSH has been enabled.

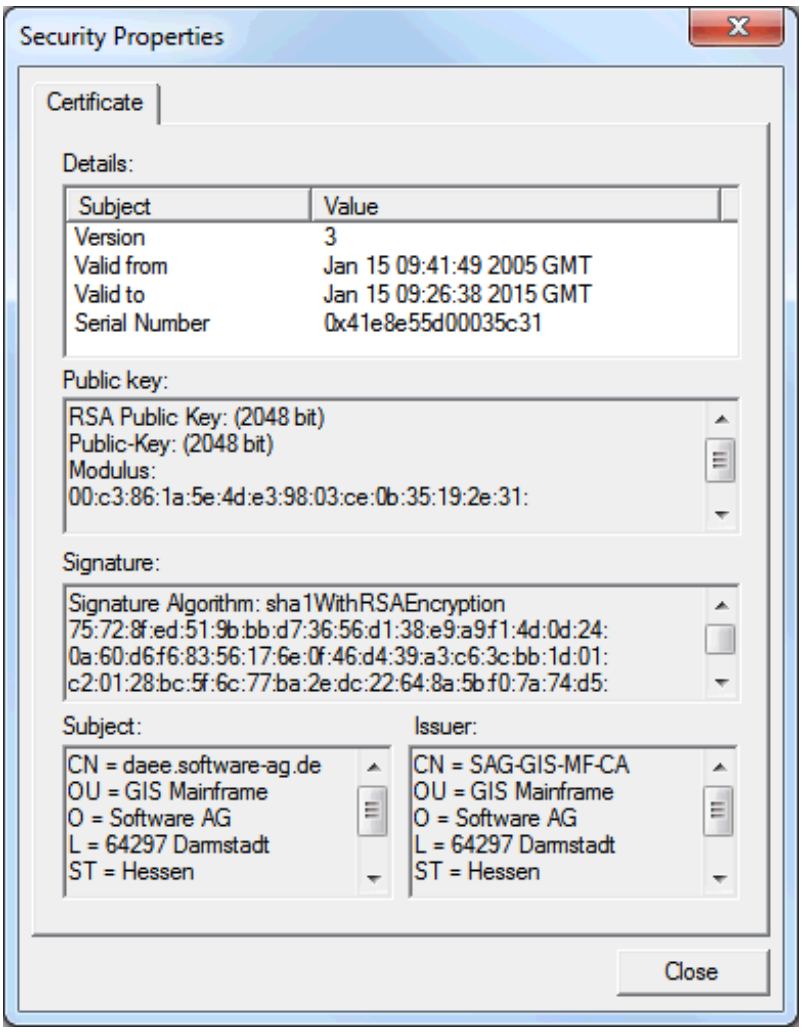
To display security information

- From the **Session** menu, choose **Security**.

Or:

Choose the following icon in the status bar which is shown when security has been enabled:

The **Security Properties** dialog box appears. The following example applies to an SSL session and shows information on the server certificate.



Printing the Contents of a Terminal Emulation Screen

You can print the contents of the terminal emulation screen on a printer that is defined under Windows.

To set up another printer

- From the **Session** menu, choose **Print Setup**.

  The **Print Setup** dialog box appears in which you can connect to another printer.

To preview the screen to be printed

1. From the **Session** menu, choose **Print Preview**.

   The print preview window appears.
2. Optionally: use the **Zoom In** and **Zoom Out** buttons to view the information in this window. Or choose the **Print** button to invoke the **Print** dialog box.
3. To return to the terminal emulation screen, choose the **Close** button.

To print the screen contents using a dialog box

1. From the **Session** menu, choose **Print**.

   Or:

   Press CTRL+P.

   Or:

   Choose the following toolbar button:

   The **Print** dialog box appears.
2. Choose the **OK** button to print the current screen.

To print the screen contents directly, without displaying a dialog box

- Press the PRINT SCREEN key.

  The contents of the current terminal emulation screen are immediately printed. The current printer settings are used.

You can also print directly from the application that is currently running on the host. You simply have to issue this application's print command. See the description of the Host Printer Manager for further information.

See also: *Downloading Data to a File or Printer Using a Dialog Box*.

Defining the Default Orientation for Printing

You can specify that the contents of a terminal emulation screen is always printed with a specific orientation: landscape, portrait or half page.

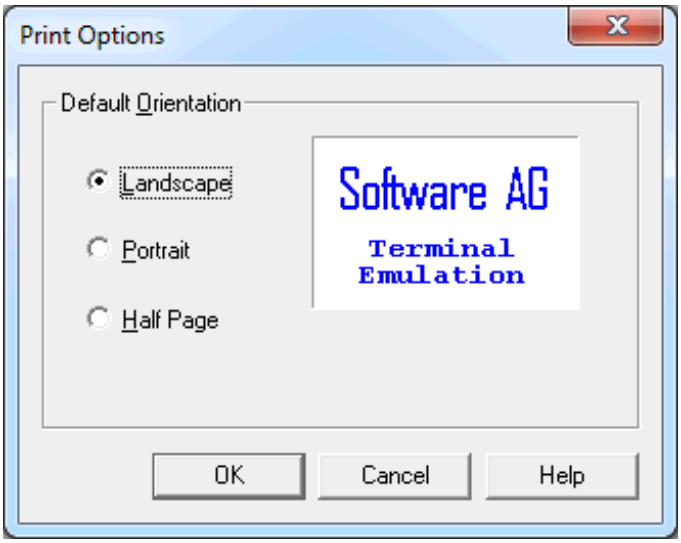
Half page means that the screen is printed in portrait orientation on the upper half of the paper. This leaves the lower half of the paper, for example, for handwritten notes.

Before you define the default orientation for the first time, the orientation is as defined with the printer settings. Your new default orientation only applies to Entire Connection. It overwrites the orientation (portrait or landscape) defined in the printer settings.

To define the default orientation

1 From the **Session** menu, choose **Print Options**.

The **Print Options** dialog box appears:



2 Select the desired orientation.

An example of the selected orientation is shown to the right of the option buttons.

3 Choose the **OK** button.

Locking the Current Session

When you lock a session, terminal emulation is no longer displayed and "locked" appears in the title bar. As long as you do not unlock the session, nobody can work with it.

To lock the session

- From the **Utilities** menu, choose **Lock/Unlock Session**.

  When the session is locked, a check mark is shown next to this command.

To unlock the session

1. From the **Utilities** menu, choose **Lock/Unlock Session**.

   The **Entire Connection Logon** dialog box appears. It shows the user name with which you are currently logged in.
2. Enter your password (if required).
3. Choose the **OK** button.

Switching to a Session in Another Terminal Application Window

Using the commands in the **Window** menu, you can switch to another session which you have previously opened (these sessions are also called active terminals). Each session is displayed in a separate terminal application window.

Up to ten active terminals can be shown as menu commands in the **Window** menu. The complete list of active terminals is always shown in the dialog box which you can also invoke from the **Window** menu. The active terminals are shown in the order in which they have been opened.

The names of the sessions are shown as menu commands or as entries in the dialog box. If a terminal does not contain an active session, "No Session" is shown. The current terminal is marked with an asterisk (*).

To switch to another terminal using the names in the menu

- From the **Window** menu, choose the name of the required session.

  Or:

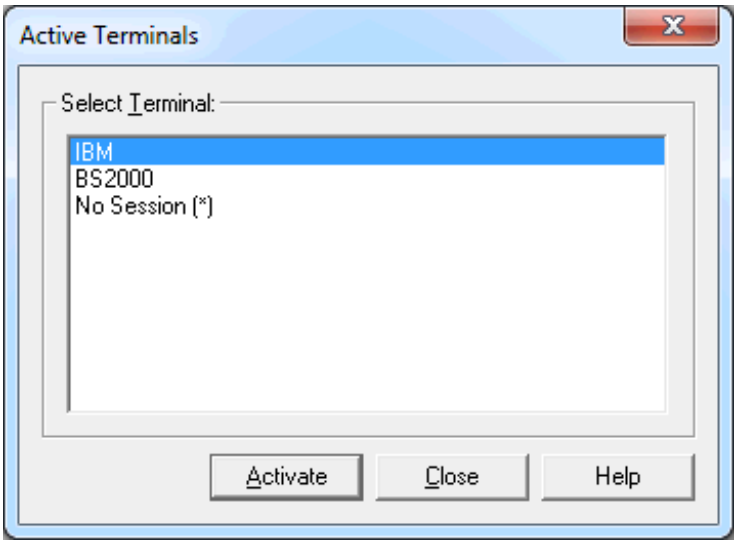
Use the key combination which is displayed next to the menu command, for example SHIFT+ALT+2 to open the second terminal in the list.

Note: The key combinations shown in the **Window** menu are always available. It is also possible to use a shorter key combination, without SHIFT (for example, ALT+2). However, this shorter key combination can only be used if it has not been defined otherwise in the current key scheme.

To switch to another terminal using a dialog box

1 From the **Window** menu, choose **Windows**.

The **Active Terminals** dialog box appears.



2 Select the required terminal.

3 Choose the **Activate** button.

Switching to the Configuration Manager

You can switch directly to the Configuration Manager.

To switch to the Configuration Manager

- From the **Utilities** menu, choose **Configuration Manager**.

When the Configuration Manager has not yet been started, it is invoked. When it is already running, its window is activated.

Switching to the Host Printer Manager

Only available to an administrator (default name: SYSTEM), and only when host printer LU support for Entire Connection has been installed.

You can switch directly to the Host Printer Manager.

To switch to the Host Printer Manager

- From the **Utilities** menu, choose **Host Printer Manager**.

  When the Host Printer Manager has not yet been started, it is invoked. When it is already running, its window is activated.

Terminating a Host Session

Before a host session is terminated, a logoff procedure, if defined, is executed. A logoff procedure contains, for example, all keyboard input required to logoff from a host session. The logoff procedure is defined in the session properties.

To terminate a host session

- From the **Session** menu, choose **Close**.

  Or:

  Choose the following toolbar button:

  Or:

  Enter the command `DISCONNECT` in the command line.

  This terminates the host session. The terminal application window, however, remains open, displaying "No Session" in its title bar.

4 Task List

- Executing a Task 52
- Terminating a Task 55

Using this utility, you can execute a task or **procedure file**.

A task is created and maintained using the Configuration Manager. For detailed information, see *Tasks* in the *Overview of Object Properties*.

Executing a Task

You can execute a task (except Software AG tasks) in the following ways:

- Select the task from the task list (see below).
- Enter the name of the task and, if applicable, all required parameters in the **command line**.
- **Execute a procedure file** which executes a task using the `EXECTASK` command.
- Press the key or key combination that is defined for the task.
- Choose the task from a user menu.

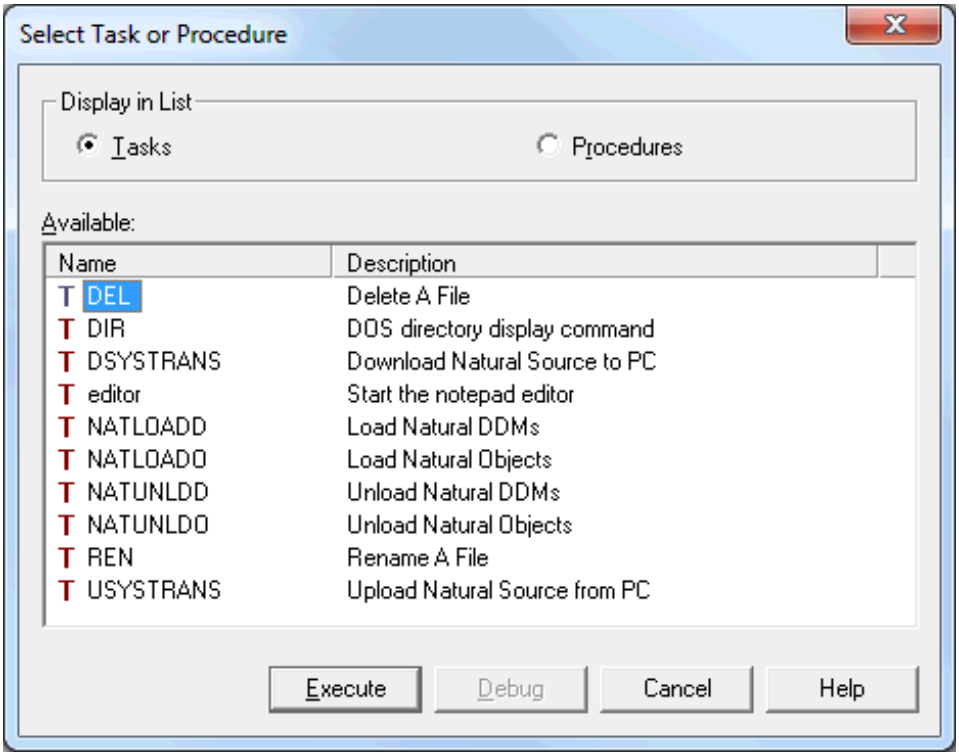
To execute a task or procedure file using the task list

1 From the **Utilities** menu, choose **Task List**.

Or:

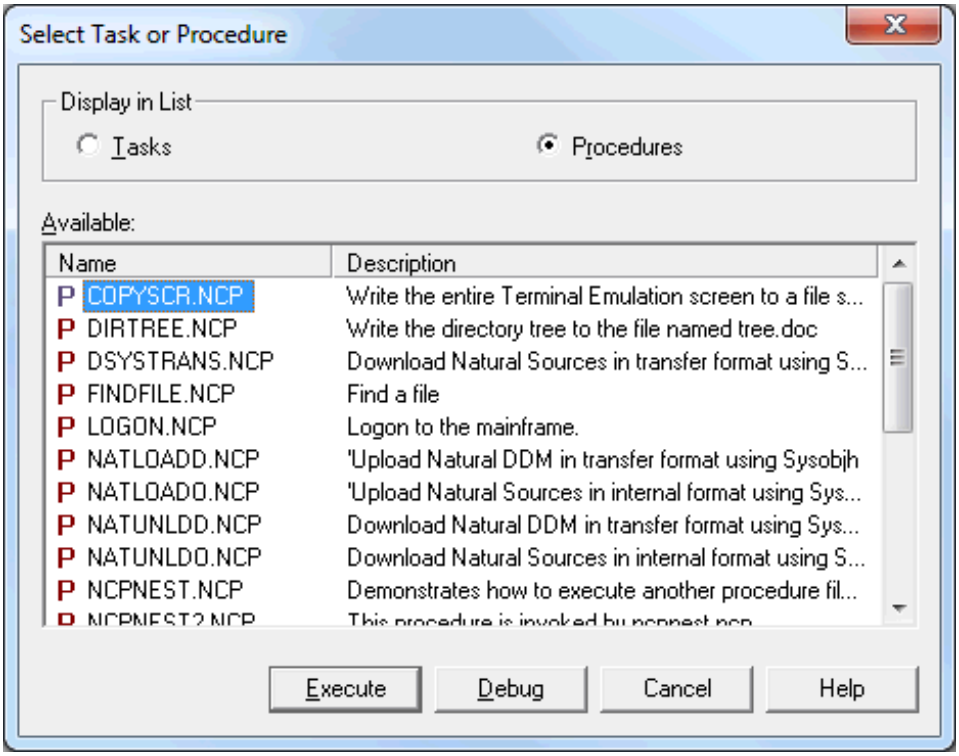
Choose the following toolbar button:

The **Select Task or Procedure** dialog box appears.



The dialog box contains a list of all tasks and procedure files that are allowed for the user, including the **transfer tasks** for Natural objects. By default, the list of available tasks is shown first.

2 To display the list of all available procedure files, choose the corresponding option button.

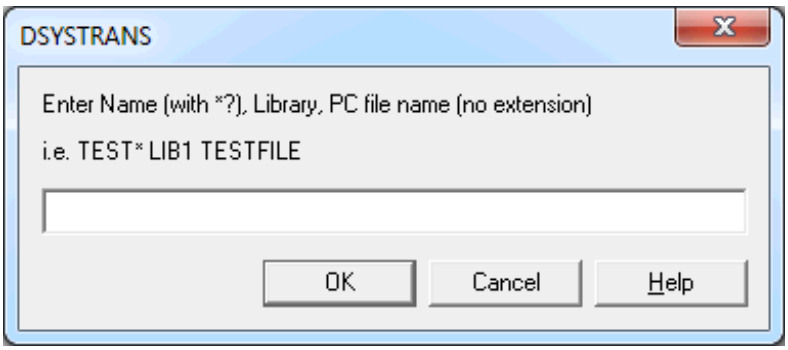


The **Debug** button only applies to procedures. See *Debugging a Procedure File* for further information.

3 Select the desired task or procedure file.

4 Choose the **Execute** button.

The selected task or procedure file is started. When parameters are required, a dialog box appears, for example:



5 Enter the parameters in the dialog box and choose the **OK** button.

See *Transferring Natural Objects* for further information.

Terminating a Task

A task is automatically terminated upon completion.

You can also cancel the task during execution. If a dialog box appears during the execution of a task, you can choose the **Cancel** button to abort the task. In all other cases, the cancelation method depends upon the type of task.

This type of task	is canceled as follows
Procedure file task	Choose **Cancel Procedure** from the **Utilities** menu or choose the corresponding toolbar button. See also: *Aborting or Terminating a Procedure File*.
Application task	Use the application-specific command. For example, in a Windows application, choose **Exit** from the **File** menu.
DOS task	Enter a DOS command (for example, press CTRL+C).

5 Unattended Workstation

- Scheduling Tasks 58
- Starting and Stopping UA Mode 62

Using the Unattended Workstation (UA), you can schedule tasks to be executed without supervision.

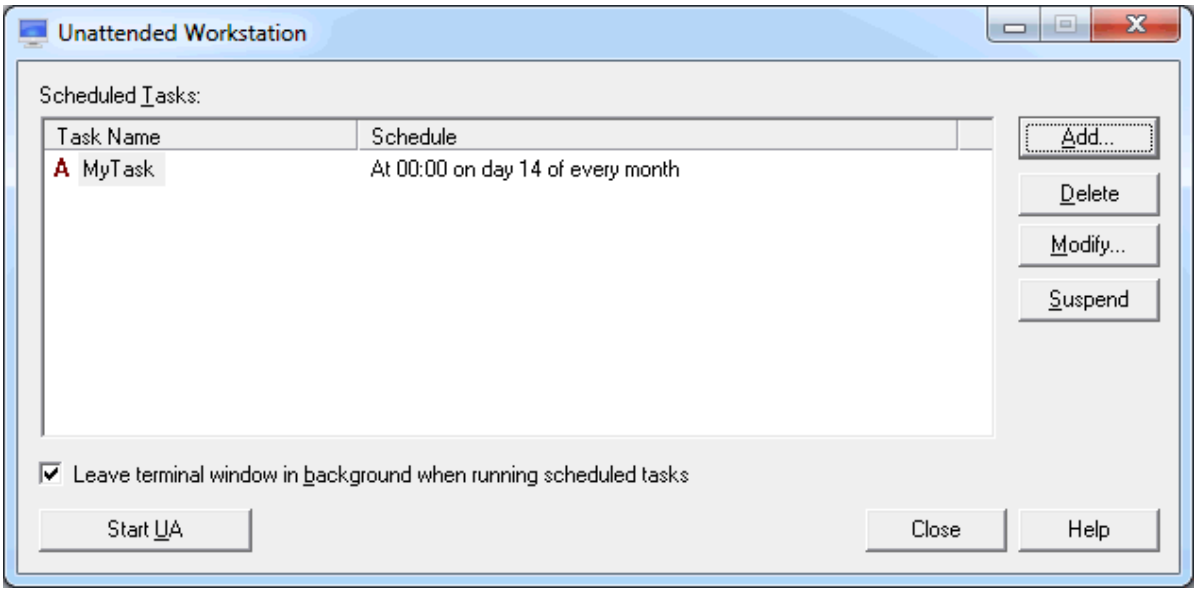
Scheduling Tasks

Each task that you want to schedule must already be defined as a task in the Configuration Manager.

To invoke the unattended workstation

- From the **Utilities** menu, choose **Unattended Workstation**.

  The **Unattended Workstation** dialog box appears. For each scheduled task, it shows the name, scheduled processing time and the current status.



The task status is indicated by the character to the left of the task name. It can be one of the following:

Status	Description
A	Active. The task is waiting to be processed.
S	Suspended. The task has been deactivated.
N	Never. The task was scheduled for processing, but at the defined processing time UA mode was not active. If you still want to process this task, you must define a new processing time.
C	Completed. The task has been processed according to schedule. From now on, it will no longer be executed.

The **Unattended Workstation** dialog box contains the following buttons (in addition to the standard buttons):

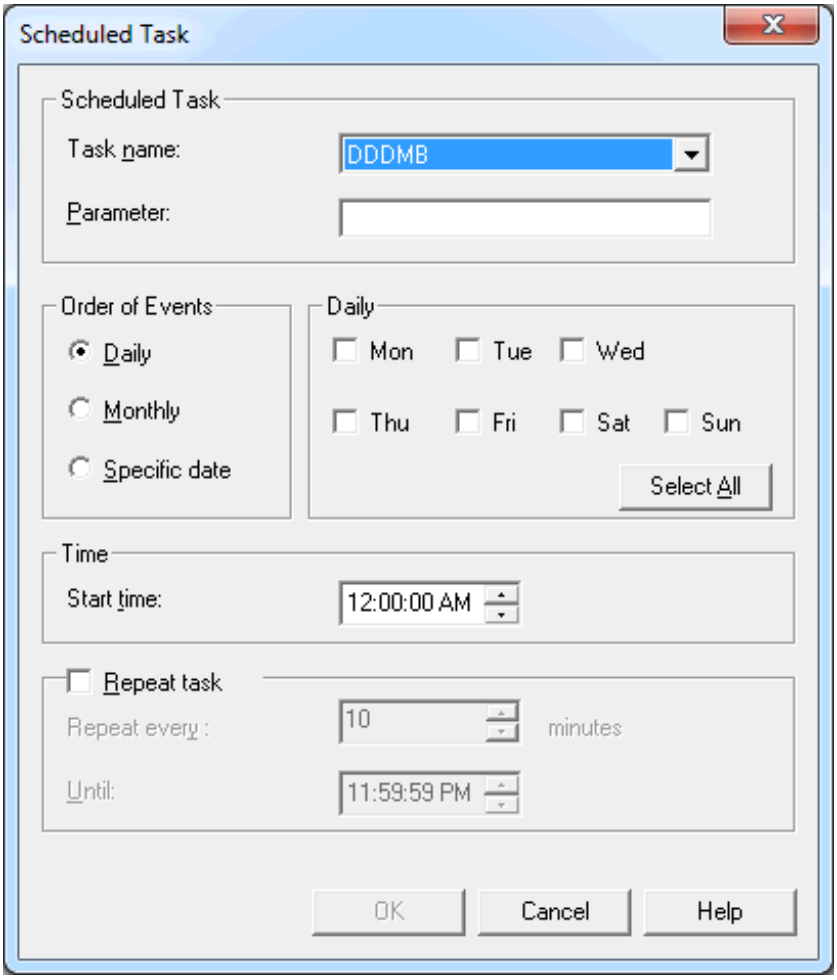
Add	Schedule a new task.
Delete	Delete the selected task.
Modify	Modify the selected task.
Suspend	Deactivate the selected task. The button name then changes to **Resume**.
Resume	Reactivate the selected task so that it can be executed according to its original schedule. The button name then changes to **Suspend**.
Start UA	Start **UA mode**. The button name then changes to **Stop UA**.
Stop UA	Stop UA mode. The button name then changes to **Start UA**.

When the check box **Leave terminal window in background when running scheduled tasks** is selected, message boxes that result from task execution are not shown.

To schedule a task

1 Choose the **Add** button.

The **Scheduled Task** dialog box appears. The date and time formats used in this dialog box are determined by your Windows settings.

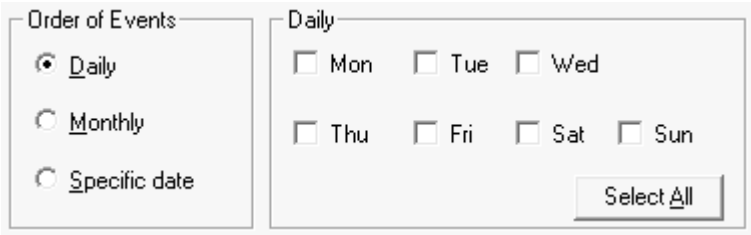


2 From the **Task name** drop-down list box, select the desired task.

3 If the task requires parameters, enter them in the **Parameter** text box.

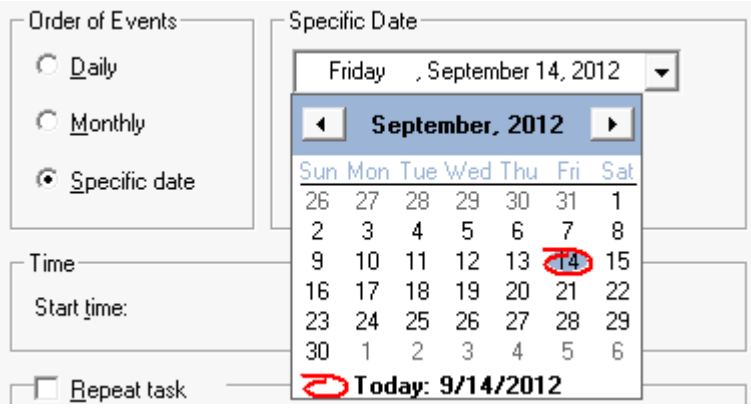
4 In the **Order of Events** group box, select the desired option button. The group box to the right changes according to the selected option button.

- When you select **Daily**, you must mark the check box for the desired day of the week. You can also choose the **Select All** button to select all days of the week at once.



- When you select **Monthly**, you must specify the number of the desired day in the spin box. When the number is greater than the last day of the month, the task is executed on the last day of the month.

- When you select **Specific date**, you must specify the desired date. To do so, select a component of the date (week day, day, month or year) and use the arrow keys to set this component to the desired value. Or open the drop-down list box to select the date from a calendar.



5 Optionally: specify the time at which the task is to be processed in the **Start time** spin box.

By default, midnight is shown as the start time.

6 If the task is to be repeated, mark the **Repeat task** check box.

7 In the **Repeat every** spin box, specify the number of minutes (1 to 1439) after which processing of the task is to be repeated.

8 Optionally: in the **Until** spin box, specify the time until when the task is to be repeated.

By default, 11:59:59 PM (or 23:59:59) is shown indicating that the task will be repeated for the rest of the day.

9 Choose the **OK** button.

The new task is now shown in the **Unattended Workstation** dialog box. Its status is A (active).

Tip: If you want to schedule a task that is to be repeated every *n* minutes without end, select **Daily** and choose the **Select All** button. Leave start time and end time with the default values, and specify the number of minutes after which processing of the task is to be repeated.

Example for a repeated task:

The task has been scheduled as follows:

- Start time: 12:00:00 AM
- End time: 10:00:00 PM
- Repeat task: every 10 minutes

UA mode is started at 09:00 AM.

In this case, where the start time has already passed, the task is started for the first time at 09:10 AM. It is repeated every 10 minutes until 10:00 PM.

Starting and Stopping UA Mode

Scheduled tasks are only processed if UA mode is active. UA mode cannot be started, if a host session is still open. Moreover, UA mode cannot be started, if all tasks in the **Unattended Workstation** dialog box have the status S (suspended), C (completed) or N (never).

It is possible that a task does not start at exactly the time it was scheduled because a previously scheduled task must be completed first.

To start UA mode

- In the **Unattended Workstation** dialog box, choose the **Start UA** button.

  UA mode is now active. The name of the command button changes to **Stop UA**.

  You cannot use the terminal application until you stop UA mode.

To stop UA mode

1. In the **Unattended Workstation** dialog box, choose the **Stop UA** button.

   The **Entire Connection Logon** dialog box appears. It shows the user name with which you are currently logged in.
2. Enter your password (if required).
3. Choose the **OK** button.

II Data Transfer

This part describes how to transfer data between the host and PC using Entire Connection.

- General Data Transfer Information
- Specifying a File Name
- Uploading and Downloading Data
- Transferring Natural Objects
- Sending Commands to the PC

6 General Data Transfer Information

- Natural Data Transfer 66
- What can be Transferred? 66
- Format Files 68
- Supported File Formats 69
- Supported Field Formats 70
- Aborting Data Transfer 71
- Auditing Data Transfer 71

Natural Data Transfer

You can transfer data between Natural applications on host systems and your PC. Data is compressed before being transferred to or from the host. Fewer buffers are thus transferred and the transfer is faster.

Your Natural administrator has to define the data transfer in the Natural parameter module (in the section where the work files and the printers are defined). If you display the work file definition using the Natural command `SYSFILE WORK`, the work files listed as PC are set for data transfer with Entire Connection.

Before you can use any data transfer function, you must tell Natural that you are an Entire Connection user. This is required only once per session. However, if your system administrator has set the parameter `PC=ON` in the Natural parameter module on the host, no further steps are required.

To tell Natural that you are an Entire Connection user

- Set the parameter `PC=ON` when invoking Natural (this parameter can also be sent to the host using a procedure file).

  Or:

  Enter the following Natural terminal command in any unprotected input field:

  ```
  %+
  ```

What can be Transferred?

Download

You can download the following from a host:

- Data into a PC data format
- Data in report form retaining the report format
- Data directly to a printer
- Data into encrypted form
- Natural data definition modules (DDMs)
- Natural source code
- Natural object code

See also: ***Supported File Formats***.

Upload

You can upload the following to a host:

- Data from a PC data format
- Binary data (for example, executable programs)
- Text data with printable characters
- Data in encrypted form
- Natural data definition modules (DDMs)
- Natural source code
- Natural object code

See also: ***Supported File Formats***.

Field Formats

You can upload and download fields in the following formats:

- Alphanumeric
- Unicode
- Binary
- Decimal
- Floating point
- Integer
- Logical
- Packed

See also: ***Supported Field Formats***.

Format Files

When the **Create format file** check box is marked in the user properties, a format file (extension *ncf*) is automatically created when you download data. Format files can also be created using the Format Converter.

A format file contains the following information:

Information	Description
File name	Name of the target file.
Status	Transfer/conversion status.
Start	Transfer/conversion start date and time.
End	Transfer/conversion end date and time.
Protocol	Version of the internal protocol.
Records	Number of records transferred/converted.
User ID	User ID of the host user.
Library	Name of the Natural library from which the download program was started.
Program	Name of the Natural download program.
Code Page	Encoding of the data. This can be UTF-16, UTF-8 (if the NCF file was created by a download with Natural for Linux and Cloud) or a Windows code page.
Translate	Information on the translation code pages is written as a support hint into the NCF file. This allows to recognize whether the Natural code page on the mainframe conforms with the code page that is the base for the Entire Connection EBCDIC/ASCII translation table. For example, "1140/1140" is written into the NCF file if the code page are conform. If the code pages are not conform, a warning is issued, for example, "1140/273 Translation code pages mismatch!". If the NCF file was created by a download with Natural for Linux and Cloud, the value here is "Not applicable".
Buffers	Number of transfer buffers used.
Globals	Decimal character, delimiter and repeat character. Entire Connection uses these characters internally. The values are defined by the Globals setting in Natural. The value for the decimal character is also used for the decimal character in the format file.
Format	The format string consists of the elements supported as of Natural Version 4.1 for Mainframes and Natural Version 6.1 for UNIX. See ***Supported Field Formats*** below. The elements are separated by a delimiter (see the table below). The following characters are used in the format string: **Character** — **Description** . — Decimal character. , — Delimiter.

Information	Description	
	/	Repeat character. The repeat character is followed by the number of repetitions. Example: N10.7/100.
Names	Natural field names.	

Sample Format File

```
File name: C:\temp\my_file.ncd
Status   : Terminated Successfully
Start    : 12/05/2007 11:58:07
End      : 12/05/2007 11:58:07
Protocol : 2
Records  : 3
UserID   : XYZ
Library  : XYZ
Program  : PCWRITE
Code Page: UTF-16
Translate: 1140/273 Translation code pages mismatch!
Buffers  : 2
Globals  : .,=
Format   : N7.3/4,A100,I2,F4,L,N7.3,
Names    :
           N
           N
           N
           N
           A
           I
           F
           L
           N
```

Supported File Formats

You can download data directly to and upload directly from files in any of the data formats supported by Entire Connection. The format is identified by the extension of the file to be downloaded or uploaded.

The following formats are supported:

Format	Extension
ASCII	*.ncd* or any other extension not included in this table (for example, *.txt*). When records are downloaded to ASCII format, trailing blanks at the end of downloaded records are removed. If you want to keep trailing blanks, you must specify this in the session properties.
Basic	*.prn*
Binary	Any binary file with any extension. The transfer format for Natural is a record with a single binary field.
dBase III	*.dbf* You can **download** field names as column labels to dBase and Lotus files. However, you cannot **upload** column labels.
Data Interchange Format	*.dif*
Encryption	*.enc*
Excel	*.xls* or *.xlsx* (depending on the Excel version)
HTML	*.htm* or *.html*. This is a specific HTML format that can also be opened with Excel.
IBM-compatible COBOL	*.ncc*
Lotus without labels	*.wks* or *.wk1*
Lotus with labels	*.wkl*
XML	*.xml*

Supported Field Formats

The following formats are supported:

Format	Description
A*x*	Alphanumeric format where *x* is a digit between 1 and 253.
A*y*	Alphanumeric format where *y* is a digit greater than 253. Can only be used with file formats HTML, XML, ASCII and COBOL. The maximum value which can be specified depends on your Natural version. See the Natural documentation for further information.
B*x*	Binary format where *x* is a digit between 1 and the maximum value. The maximum value which can be specified depends on your Natural version. See the Natural documentation for further information.
F4	Floating point format with 4 bytes.
F8	Floating point format with 8 bytes.
I*x*	Integer format where *x* is 1, 2 or 4.
L	Logical format (true or false).
N*x*.0	Numeric format where *x* is a digit between 1 and 29.

Format	Description
N*x*.*y*	Decimal format where *x* is a digit between 1 and 29. *y* must be less than or equal to 7. The sum of *x* and *y* must not be greater than 29.
P*x*.0	Packed format where *x* is a digit between 1 and 29.
P*x*.*y*	Packed decimal format where *x* is a digit between 1 and 29. *y* must be less than or equal to 7. The sum of *x* and *y* must not be greater than 29.
U*x*	Unicode format where *x* is a digit between 1 and 253.
U*y*	Unicode format where *y* is a digit greater than 253. Can only be used with file formats HTML, XML, ASCII and COBOL. The maximum value which can be specified depends on your Natural version. See the Natural documentation for further information.

For further information on field formats, see the Natural documentation.

Aborting Data Transfer

You can abort the current upload or download of data. This is useful, for example, if you notice that you are downloading the wrong file.

To abort data transfer

- From the **Utilities** menu, choose **Cancel Transfer**.

  Or:

  Choose the following toolbar button:

Auditing Data Transfer

If the **Audit data transfer** check box is marked in the user properties, Entire Connection writes data transfer information to the log file `<username>`*.log*.

The log file contains the following information:

- Transfer start date and time
- Transfer end date and time
- Name of the target file
- Transfer status

- Number of transferred records
- Hexadecimal format
- Number of used transfer buffers
- User ID of the host user.
- Name of the Natural library from which the download program was started
- Name of Natural program that was used to download data

Example

```
05/10/2007 09:45:37 Started download of file D:\tmp\Employees.ncd
05/10/2007 09:45:37 Ended   download of file D:\tmp\Employees.ncd; Statistics follow:
05/10/2007 09:45:37          File name: D:\tmp\Employees.ncd
05/10/2007 09:45:37          Status   : Terminated Successfully
05/10/2007 09:45:37          Records  : 220
05/10/2007 09:45:37          Format   : R02A14
05/10/2007 09:45:37          Buffers  : 3
05/10/2007 09:45:37          UserID   : USER1
05/10/2007 09:45:37          Library  : USER1
05/10/2007 09:45:37          Program  : WPC3
05/10/2007 09:45:37       End of statistics for download of file D:\tmp\Employees.ncd
```

7 Specifying a File Name

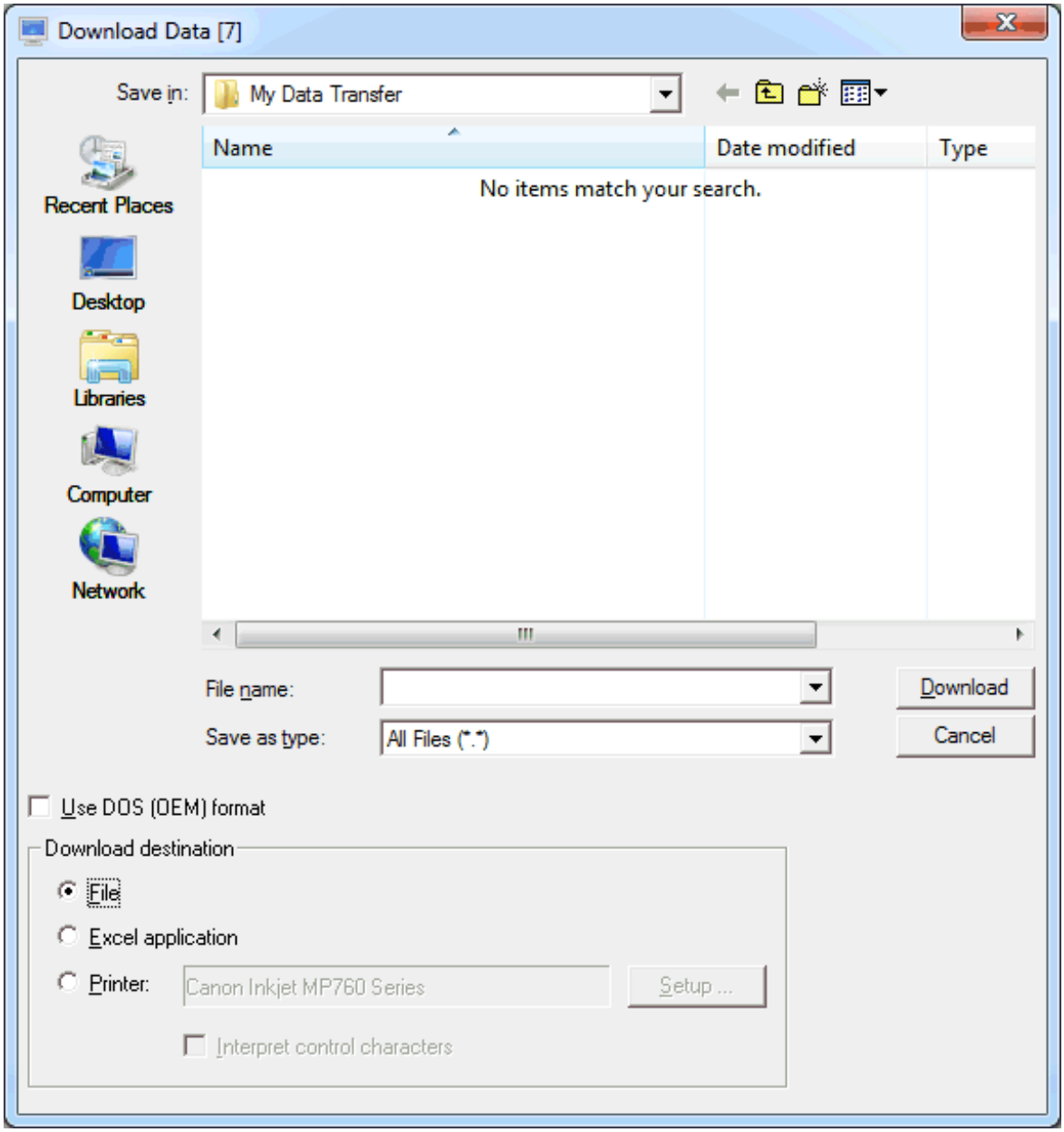
- Downloading Data Using a Dialog Box 74
- Uploading Data Using a Dialog Box 79
- Specifying a File Name Using the SET Command 80
- Creating a File Name Dynamically 82

The format of a file is determined by its extension.

Downloading Data Using a Dialog Box

You can download data to a file, a printer or directly to Excel.

If you download data to a file and no file name has been associated with the work file number used in a Natural program, a dialog box prompts you for a file name at execution time. The Natural work file number is shown in brackets in the title bar.



To download data to a file

1 Make sure that the download destination **File** is selected.

2 In the **File name** text box, enter the name of the file in which the data are to be stored on the PC.

   If you do not specify a path, the file is stored in the current directory.

3 Choose the desired file type from the **Save as type** drop-down list box.

4 Specify the format that is to be used.

   If the **Use DOS (OEM) download format** check box is not marked, the ANSI character set is used. This character set is used by all Windows applications.

   If this check box is marked, the DOS character set is used. This character set is used by most MS-DOS applications.

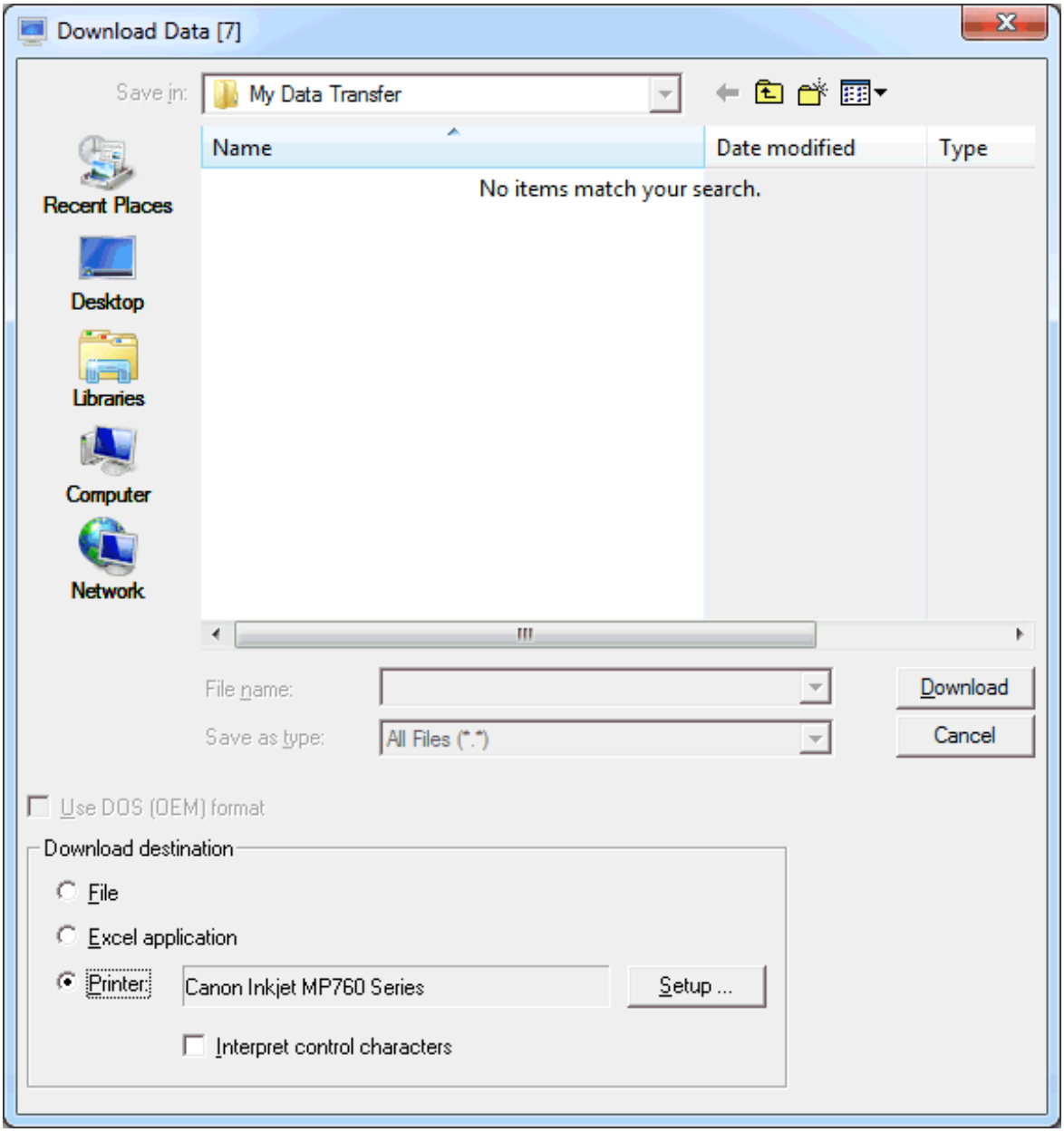
5 Choose the **Download** button.

   The data are downloaded to the specified file.

To download data to a printer

1 Leave the **File name** text box empty.

2 Select the **Printer** option button.



3 Optionally: choose the **Setup** button to select another printer or define another orientation (e.g. landscape).

4 Mark the **Interpret control characters** check box, if you want to download a Natural report containing control characters. Valid control characters are:

Control Character	Function
1	New page
blank	Single blank line
0	Double blank line
-	Triple blank line

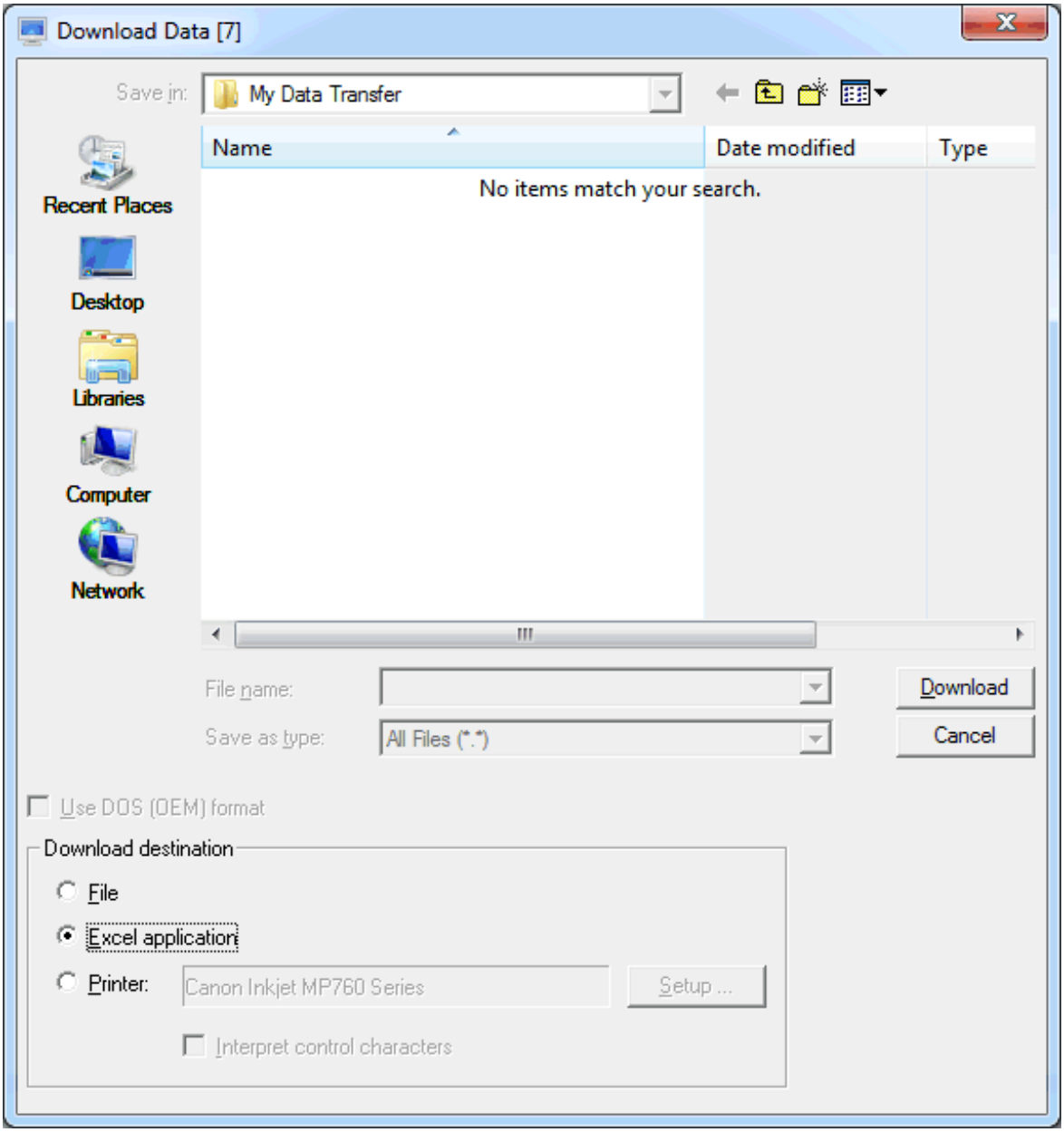
5 Choose the **Download** button.

The data are downloaded to the specified printer.

To download data directly to Excel

1 Leave the **File name** text box empty.

2 Select the **Excel application** option button.



3 Choose the **Download** button.

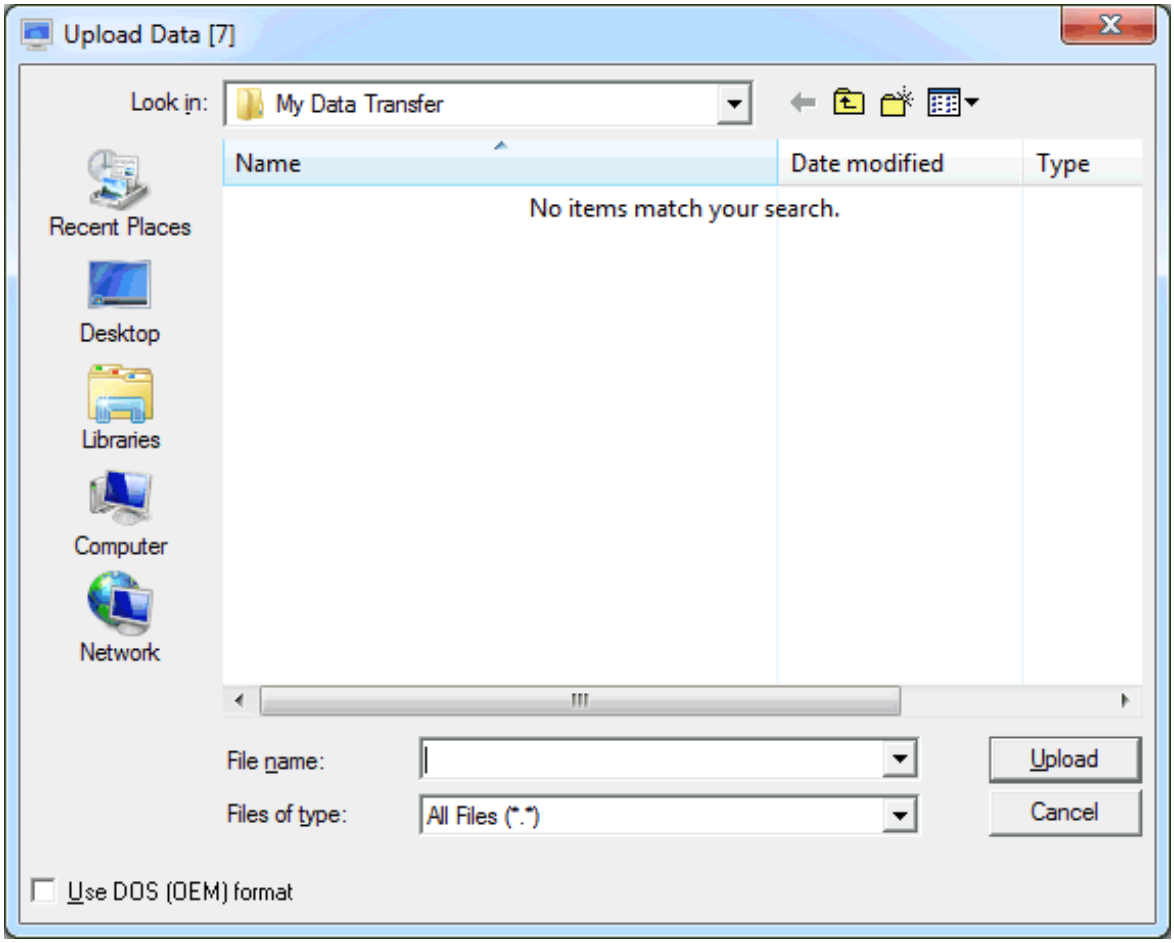
When download is complete, Excel is automatically started and the data are shown in an untitled Excel workbook.

4 Save the downloaded data in Excel.

Note: A format file (extension *ncf*) is not created in this case.

Uploading Data Using a Dialog Box

If no file name has been associated with the work file number used in a Natural program, a dialog box prompts you for a file name at execution time. The Natural work file number is shown in brackets in the title bar.



To upload data to a file

1 In the **Upload Data** dialog box, select the file to be uploaded.

2 Specify the upload format that is to be used.

If the **Use DOS (OEM) download format** check box is not marked, the ANSI character set is used. This character set is used by all Windows applications.

If this check box is marked, the DOS character set is used. This character set is used by most MS-DOS applications.

3 Choose the **Upload** button.

Specifying a File Name Using the SET Command

Natural programs download or upload data by referring to the number of a work file or printer file. This number must be an integer between 1 and 32. You can define up to 32 work files and up to 31 printer files.

A file name can be associated with several numbers for work files and printer files. The number of a work file or printer file, however, can only be associated with one file name.

To associate a file name with the number of a work file or printer file

- Issue the `SET` command using the following syntax:

```
SET PCFILE x DOWN DATA filename
```

```
SET PCFILE x UP DATA filename
```

```
SET PCFILE y DOWN REPORT filename
```

where *x* is a valid work file number, *y* is a valid printer file number, and *filename* is a valid file name. There is no warning if a file with that name already exists.

filename can be specified either with or without the path to the file. For example:

```
SET PCFILE 7 DOWN DATA C:\test\mydata.ncd
```

filename can also contain the name of an existing environment variable. The environment variable must be enclosed in percent signs (%). The environment variable may contain only a file name, only a path without a file name, both path and file name, or any part of *filename*. For example:

```
SET PCFILE 7 DOWN DATA %TEMP%\mydata.ncd
```

When the current value of the environment variable `TEMP` in the above example is "C:\Users\Jack\AppData\Local\Temp", the path and file name for the download is set to "C:\Users\Jack\AppData\Local\Temp\mydata.ncd". When the environment variable contains only a path (as in the above example), make sure to specify a backslash (\) before the file name.

If an environment variable with the given name is not found, the name including the percent signs remains as specified. For example, `SET PCFILE 7 DOWN DATA %UNKNOWN%\mydata.ncd` would set the path and file name for this download to "%UNKNOWN%\mydata.ncd".

When you download data directly to your local printer, you must specify PRN, LPT1, LPT2 or LPT3 as the file name. If the first character in the downloaded data is to be interpreted as a printer control character, you must specify *PRN, *LPT1, *LPT2 or *LPT3 as the file name.

The file name remains associated with the specified work file number until you leave Entire Connection or until the `SET` command is issued to reset the work file.

Note: To associate a file name with a printer file number, use `DOWN REPORT`.

To reset the work file or printer file

- Issue the `SET` command without specifying a file name.

  The following commands reset the file numbers defined above:

```
SET PCFILE x DOWN DATA
```

```
SET PCFILE x UP DATA
```

```
SET PCFILE y DOWN REPORT
```

  where *x* is a valid work file number, and *y* is a valid printer file number.

To specify an XSL style sheet type and name for download to XML

- Issue the `SET` command using the following syntax:

```
SET PCFILE x DOWN CONVERT stylesheet-type stylesheet-name
```

  where *x* is a valid work file number, *stylesheet-type* is the type of the **style sheet** (for example, "text/xsl") and *stylesheet-name* is a valid file name. There is no warning if a file with that name does not exist.

  Example:

```
SET PCFILE 7 DOWN CONVERT text/xsl employ.xsl
```

  This inserts the following the style sheet definition in all downloaded XML files:

```
<?xml-stylesheet type="text/xsl" href="employ.xsl"?>
```

The style sheet type and name remain associated with the specified work file number until you leave Entire Connection or until the `SET` command is issued to reset the work file.

To reset the XSL style sheet type and name for download to XML

- Issue the `SET` command without specifying a style sheet type and name:

```
SET PCFILE x DOWN CONVERT
```

where *x* is a valid work file number.

Creating a File Name Dynamically

You can instruct Entire Connection to dynamically create a file name at download time. In this case, the file name consists of the date and time the file was created. The format is DDHHMMSS (DD = day, HH = hour, MM = minute, SS = second).

To create a file name dynamically

- Issue the following command:

```
SET PCFILE x DOWN DATA ~~RANDOM.ext
```

You can specify an extension (optional). You must not specify a drive or directory.

8 Uploading and Downloading Data

- Uploading and Downloading Encrypted Files 84
- Downloading Fixed-Length Data Records 84
- Downloading Variable Data Records 84
- Downloading Reports 85
- Downloading Binary Data 85
- Downloading Data to dBase or Lotus with Labels 86
- Downloading Data to HTML 86
- Downloading Data to XML 86
- Downloading Data to Excel 95
- Downloading Numeric Data to ASCII Format 95
- General Upload Information 95
- Uploading Binary Data 96
- Uploading Data with Labels from dBase or Lotus 96
- Uploading HTML Files 96
- Uploading XML Files 97
- Uploading Excel Files 98
- Uploading ASCII Data 98
- Restrictions for Unicode Data 98

Notes:

1. The Natural program examples mentioned in this chapter are located in Entire Connection's *proc* folder. See also *Program Folders* in *Installing Entire Connection*.
2. With Entire Connection, it is possible to upload NCD files which have been created in Unicode UTF-8 format by Natural for Linux and Cloud.

Uploading and Downloading Encrypted Files

Using Entire Connection, you can upload encrypted files and download data to an encrypted format. To do so, specify *enc* as the file extension.

Enhanced encryption provides additional security for data transfer. In the Configuration Manager, the user can specify whether enhanced encryption is to be used.

Downloading Fixed-Length Data Records

The most common form of data transfer is downloading data that consist of multiple records. These records have a fixed number of fields and can be converted into one of the PC file formats supported by Entire Connection.

Natural program examples: *Natex01.nsp* and *Natex02.nsp*.

Downloading Variable Data Records

Data that are not in the form of a record and have neither a fixed length nor a fixed format are referred to as variable data. Variable data cannot be uploaded or converted. The Natural statement for downloading variable data is `DOWNLOAD PC FILE n VARIABLE field1 .. fieldn` or `WRITE PC FILE n VARIABLE field1 .. fieldn`. Variable data can only be downloaded into ASCII format. Binary data are converted to readable ASCII format, and leading zeros are always substituted with blanks.

Natural program example: *Natex03.nsp*.

Downloading Reports

To download a report, you must associate a printer file number with the file name. The data are downloaded to the PC in a printable format (for example, with headings, page numbers, etc.). Reports cannot be uploaded or converted.

Natural program example: *Natex04.nsp*.

Downloading Binary Data

Binary data are typically object code or executable code which does not contain displayable or printable characters. To avoid the standard character translations that occur during data transfer, Entire Connection uses special methods for transferring binary data.

To download binary data, you must define a binary variable (B1 to B126). The best performance is achieved when you choose a block size of 126 bytes (i.e. a binary variable B126).

With Natural Version 4.1 for Mainframes and Natural Version 6.1 for UNIX, you can use binary variables greater than 126. The maximum value which can be specified depends on your Natural version. See the Natural documentation for further information.

You must mark the end of your binary data with X'FF' (i.e. the hexadecimal value of FF). If the last block of downloaded data contains less data than the chosen block size, you must insert X'FF' at the position that marks the end of binary data and fill the rest of the block with binary zeros. If the amount of data in last block equals the chosen block size, you must include an additional block containing X'FF' and fill the rest of the block with binary zeros to mark the end of binary data.

It is possible to combine binary data and all other types of data (alphanumeric, numeric, integer etc.). Files containing binary data and ASCII data, or multiple binary fields, can be uploaded or downloaded. During data transfer, such a file is treated like an ASCII file. The records within the file must have a fixed length and must end with CR and LF. Spaces at the end of the record are not truncated. Tabulator characters (0X09) are not recognized.

Natural program examples: *Natex05.nsp* and *Natex06.nsp*.

Downloading Data to dBase or Lotus with Labels

When you download data with the extension *dbf* to dBase, dBase column labels are automatically generated for Natural field names.

When you download data with the extension *wkl* to Lotus, the first record is reserved for column labels. If field names are available from the host, the names of the variables are automatically entered as column labels in the first row of the file. If field names are not available, the first row is left blank.

Downloading Data to HTML

When you download data to an HTML file (extension *htm* or *html*), Entire Connection creates an HTML format which can be opened with a browser (e.g. Internet Explorer) and with Excel.

The character set to be used is defined in the session properties. It is also possible to download data to a file with the extension *htm* or *html* in NCD (ASCII) format. See the description of the **Data Transfer** property page for further information.

Downloading Data to XML

When you download data to an XML file (extension *xml*), the resulting file has the following structure:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NATURAL_DOWNLOAD LIBRARY="NATLIB" program="MYPROG" user="THE_USER">
   <RECORD>
      <. . .>
   </RECORD>
     . . .
   <RECORD>
      < . . .>
   </RECORD>
</NATURAL_DOWNLOAD>
```

Each `<RECORD>` element contains the fields of a downloaded record. The names of the variables are downloaded as element names, and the values of the variables are downloaded as element contents.

Example for a simple Natural download to XML format:

```
0010 DEFINE DATA LOCAL
0020 1 NAME (A40)
0030 1 CITY (A40)
0040 END-DEFINE
0050 NAME := 'SMITH'
0060 CITY := 'NEW YORK'
0070 WRITE WORK 7 NAME CITY
0080 NAME := 'BUSH'
0090 CITY := 'LONDON'
0100 WRITE WORK 7 NAME CITY
0110 END
```

This results in the following XML file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NATURAL_DOWNLOAD library="NTWTEST" program="PROG1" user="NTWTEST1">
   <RECORD>
      <NAME>SMITH</NAME>
      <CITY>NEW YORK</CITY>
   </RECORD>
   <RECORD>
      <NAME>BUSH</NAME>
      <CITY>LONDON</CITY>
   </RECORD>
</NATURAL_DOWNLOAD>
```

The character set to be used is defined in the session properties. It is also possible to download data to a file with the extension *xml* in NCD (ASCII) format. See the description of the **Data Transfer** property page for further information.

The following topics are covered below:

- Special Characters
- Unnamed Fields
- Natural Groups
- Unidimensional Arrays
- Two- and Three-Dimensional Arrays
- Using Style Sheets

For further information on XML, see ***https://www.w3.org/XML/***.

Special Characters

The following special characters, which are allowed in Natural variable names, are not allowed in XML element names:

#
@
/
$
§
&
+

During download, each of these characters in a Natural variable name is converted to an underscore (_) character.

Unnamed Fields

Fields without variable names are downloaded to an `<UNNAMED_FIELD_n>` element, where *n* represents the index of the downloaded field in the record.

Example for a Natural download without field names:

```
0010 WRITE WORK 7 2 3 4
0020 END
```

This results in the following XML file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NATURAL_DOWNLOAD library="NTWTEST" program="DN_ARR4" user="NTWTEST1">
   <RECORD>
      <UNNAMED_FIELD_0>2</UNNAMED_FIELD_0>
      <UNNAMED_FIELD_1>3</UNNAMED_FIELD_1>
      <UNNAMED_FIELD_2>4</UNNAMED_FIELD_2>
   </RECORD>
</NATURAL_DOWNLOAD>
```

Natural Groups

Natural groups are downloaded as parent elements for the fields.

Example for downloading Natural groups:

```
0010 DEFINE DATA LOCAL
0020 1 NAME
0030 2 FIRST_NAME (A40)
0040 2 LAST_NAME (A40)
0050 1 CITY (A40)
0060 END-DEFINE
0070 FIRST_NAME := 'ALAN'
0080 LAST_NAME := 'SMITH'
0090 CITY := 'NEW YORK'
0100 WRITE WORK 7 NAME CITY
0110 FIRST_NAME := 'TOM'
0120 LAST_NAME := 'BUSH'
0130 CITY := 'LONDON'
0140 WRITE WORK 7 NAME CITY
0150 END
```

This results in the following XML file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NATURAL_DOWNLOAD library="NTWTEST" program="PROG2" user="NTWTEST1">
   <RECORD>
      <NAME>
         <FIRST_NAME>ALAN</FIRST_NAME>
         <LAST_NAME>SMITH</LAST_NAME>
      </NAME>
      <CITY>NEW YORK</CITY>
   </RECORD>
   <RECORD>
      <NAME>
         <FIRST_NAME>TOM</FIRST_NAME>
         <LAST_NAME>BUSH</LAST_NAME>
      ></NAME>
      <CITY>LONDON</CITY>
   </RECORD>
</NATURAL_DOWNLOAD>
```

Unidimensional Arrays

During download, `index` attributes are created for the elements of a unidimensional array.

Example for downloading unidimensional arrays:

```
DEFINE DATA LOCAL
1 NAME
2 FIRST_NAMES (1:3)
3 FIRST_NAME (A40)
2 LAST_NAME (A40)
1 CITY (A40)
END-DEFINE
*
FIRST_NAME(1):='ALAN'
```

```
FIRST_NAME(2):='PAUL'
FIRST_NAME(3):='OLIVER'
LAST_NAME := 'SMITH'
CITY := 'NEW YORK'
WRITE WORK 7 NAME CITY
*
RESET FIRST_NAMES(1:3)
FIRST_NAME(1) := 'ERIKA'
FIRST_NAME(2) := 'CLARA'
LAST_NAME := 'BUSH'
CITY := 'LONDON'
WRITE WORK 7 NAME CITY
END
```

This results in the following XML file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NATURAL_DOWNLOAD library="NTWTEST" program="ADDRESS" user="NTWTEST1">
   <RECORD>
      <NAME>
         <FIRST_NAME index="1">ALAN</FIRST_NAME>
         <FIRST_NAME index="2">PAUL</FIRST_NAME>
         <FIRST_NAME index="3">OLIVER</FIRST_NAME>
         <LAST_NAME>SMITH</LAST_NAME>
      </NAME>
      <CITY>NEW YORK</CITY>
   </RECORD>
   <RECORD>
      <NAME>
         <FIRST_NAME index="1">ERIKA</FIRST_NAME>
         <FIRST_NAME index="2">CLARA</FIRST_NAME>
         <FIRST_NAME index="3"></FIRST_NAME>
         <LAST_NAME>BUSH</LAST_NAME>
      </NAME>
      <CITY>LONDON</CITY>
   </RECORD>
</NATURAL_DOWNLOAD>
```

Two- and Three-Dimensional Arrays

Two- and three-dimensional arrays are downloaded as child elements.

In a two-dimensional array, the first child element is `<COLUMN>`. Its `pos` attribute indicates the position of the column in the array. `<ROW>` is a child element of `<COLUMN>`. Its `pos` attribute indicates the position of the row in the array. The downloaded two-dimensional array has the following structure:

```
<DIM2ARRAY>
   <COLUMN pos="1">
      <ROW pos="1">data1</ROW>
      <ROW pos="2">data2</ROW>
         . . .
   </COLUMN>
   <COLUMN>
      . . .
   </COLUMN>
</DIM2ARRAY>
```

A three-dimensional array contains the additional element `<PLANE>`. Its `pos` attribute indicates the position of the plane in the three-dimensional array. The downloaded three-dimensional array has the following structure:

```
<DIM3ARRAY>
   <PLANE>
      <COLUMN pos="1">
         <ROW pos="1">data1</ROW>
         <ROW pos="2">data2</ROW>
            . . .
      </COLUMN>
      <COLUMN>
         . . .
      </COLUMN>
   </PLANE>
   <PLANE>
      .  .  .
   </PLANE>
</DIM3ARRAY>
```

Example for downloading a three-dimensional array:

```
DEFINE DATA LOCAL
1 ARRAY
  2 MYROW (2:3)
   3 MYCOLUMN (1:2)
     4 MYPLANE (1:2)
       5 FIELD (P3) INIT (2:3,1:2,1:2) <99>
END-DEFINE
WRITE WORK 7 FIELD (*,*,*)
END
```

This results in the following XML file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NATURAL_DOWNLOAD library="NTWTEST" program="DN_ARR4" user="NTWTEST1">
   <RECORD>
      <ARRAY>
         <FIELD>
            <PLANE pos="1">
               <COLUMN pos="1">
                  <ROW pos="2">99</ROW>
                  <ROW pos="3">99</ROW>
               </COLUMN>
               <COLUMN pos="2">
                  <ROW pos="2">99</ROW>
                  <ROW pos="3">99</ROW>
               </COLUMN>
            </PLANE>
            <PLANE pos="2">
               <COLUMN pos="1">
                  <ROW pos="2">99</ROW>
                  <ROW pos="3">99</ROW>
               </COLUMN>
               <COLUMN pos="2">
                  <ROW pos="2">99</ROW>
                  <ROW pos="3">99</ROW>
               </COLUMN>
           </PLANE>
         </FIELD>
      </ARRAY>
   </RECORD>
</NATURAL_DOWNLOAD>
```

Using Style Sheets

You can use XSL style sheets with your downloaded XML files. This is helpful, for example, if you want to display the contents of an XML file in a browser. To do so, you must first create an appropriate style sheet (see *https://www.w3.org/Style/*) and then insert the style sheet definition in the header of your downloaded XML file.

Note: With the `SET` command, a style sheet definition can be added automatically to all downloaded XML files. See *Specifying a File Name Using the SET Command*.

Example for downloading data and displaying it as a simple table using a style sheet:

```
0010 DEFINE DATA LOCAL
0020 1 EMPLOY-VIEW VIEW OF EMPLOYEES
0030 2 FULL-NAME
0040 3 FIRST-NAME
0050 3 MIDDLE-NAME
0060 3 NAME
0070 2 PERSONNEL-ID
0080 1 VEHIC-VIEW VIEW OF VEHICLES
0090 2 MAKE
0100 2 REG-NUM
0110 END-DEFINE
0120 FIND EMPLOY-VIEW WITH CITY = 'DARMSTADT' SORTED BY NAME
0130 FIND VEHIC-VIEW WITH PERSONNEL-ID = PERSONNEL-ID
0140 WRITE WORK 7 FULL-NAME MAKE
0150 END-FIND
0160 END-FIND
0170 END
```

This results in the following XML file (the style sheet definition that is to be entered manually is indicated in boldface):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="employ2.xsl"?>
<NATURAL_DOWNLOAD library="NTWTEST" program="EMPLOYE2" user="NTWTEST1">
  <RECORD>
    <EMPLOY-VIEW>
      <FIRST-NAME>KRISTINA</FIRST-NAME>
      <MIDDLE-NAME>MARIA</MIDDLE-NAME>
      <NAME>FALTER</NAME>
    </EMPLOY-VIEW>
    <VEHIC-VIEW>
      <MAKE>FORD</MAKE>
    </VEHIC-VIEW>
  </RECORD>
  <RECORD>
    <EMPLOY-VIEW>
      <FIRST-NAME>DIETER</FIRST-NAME>
      <MIDDLE-NAME>PETER</MIDDLE-NAME>
      <NAME>MUELLER</NAME>
    </EMPLOY-VIEW>
    <VEHIC-VIEW>
      <MAKE>FORD</MAKE>
    </VEHIC-VIEW>
  </RECORD>
  <RECORD>
    <EMPLOY-VIEW>
      <FIRST-NAME>CHRISTIAN</FIRST-NAME>
      <MIDDLE-NAME></MIDDLE-NAME>
      <NAME>SCHIRM</NAME>
    </EMPLOY-VIEW>
    <VEHIC-VIEW>
```

```
      <MAKE>BMW</MAKE>
    </VEHIC-VIEW>
  </RECORD>
  <RECORD>
    <EMPLOY-VIEW>
      <FIRST-NAME>ROLAND</FIRST-NAME>
      <MIDDLE-NAME>KARL</MIDDLE-NAME>
      <NAME>VOGEL</NAME>
    </EMPLOY-VIEW>
    <VEHIC-VIEW>
      <MAKE>VOLVO</MAKE>
    </VEHIC-VIEW>
  </RECORD>
</NATURAL_DOWNLOAD>
```

The name of the following style sheet (*employ2.xsl*) has been manually inserted in the above XML file:

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <BODY>
        <table border="5">
          <tr>
          <th colspan="3" bgcolor="#99CCCC">Name</th>
          <th rowspan="2" bgcolor="#99CCCC">Vehicle</th>
          </tr>
          <tr>
          <th bgcolor="#0099CC">First</th>
          <th bgcolor="#0099CC">Middle</th>
          <th bgcolor="#0099CC">Last</th>
          </tr>
          <xsl:for-each select="NATURAL_DOWNLOAD/RECORD">
            <tr>
            <xsl:for-each select="EMPLOY-VIEW">
                <td><xsl:value-of select="FIRST-NAME"/></td>
                <td><xsl:value-of select="MIDDLE-NAME"/></td>
                <td><xsl:value-of select="NAME"/></td>
            </xsl:for-each>
            <xsl:for-each select="VEHIC-VIEW">
              <td><xsl:value-of select="MAKE"/></td>
            </xsl:for-each>
            </tr>
          </xsl:for-each>
         </table>
      </BODY>
    </HTML>
  </xsl:template>
</xsl:stylesheet>
```

When you display your XML file in a browser, it looks as follows:

Name			Vehicle
First	Middle	Last	
KRISTINA	MARIA	FALTER	FORD
DIETER	PETER	MUELLER	FORD
CHRISTIAN		SCHIRM	BMW
ROLAND	KARL	VOGEL	VOLVO

Downloading Data to Excel

When you download data to Excel (*xls* or *xlsx*, depending on the installed Excel version), Entire Connection does not create the file itself. Instead, it starts the automation server for the corresponding Excel version. Depending on the Excel version installed on your PC, data are thus downloaded to the corresponding Excel format. If Excel 2007 or above is the installed version, the file that is created has the new Excel file format and the corresponding new extension *xlsx*.

Downloading Numeric Data to ASCII Format

When you download numeric data to ASCII format, the position preceding a number is reserved to indicate negative or positive numbers. For positive numbers, this position contains a blank character. For negative numbers, it contains a minus (-) character.

General Upload Information

You can upload data from any of the supported file formats. Exception: variable data and reports cannot be uploaded. Once the data are uploaded, you can use them in a Natural application.

Natural program examples: *Natex07.nsp* and *Natex08.nsp*.

Uploading Binary Data

Binary data are typically object code or executable code which does not contain displayable or printable characters. To avoid the standard character translations that occur during data transfer, Entire Connection uses special methods for transferring binary data.

To upload binary data, you must define a binary variable to which the data are uploaded.

With Natural Version 4.1 for Mainframes and Natural Version 6.1 for UNIX, you can use binary variables greater than 126. The maximum value which can be specified depends on your Natural version. See the Natural documentation for further information.

It is possible to combine binary data and all other types of data (alphanumeric, numeric, integer etc.). Files containing binary data and ASCII data, or multiple binary fields, can be uploaded or downloaded. During data transfer, such a file is treated like an ASCII file. The records within the file must have a fixed length and must end with CR and LF. Spaces at the end of the record are not truncated. Tabulator characters (0X09) are not recognized.

Entire Connection marks the end of the binary data with X'FF' (i.e. the hexadecimal value of FF) and fills the rest of the upload buffer with binary zeros.

Natural program examples: *Natex09.nsp* and *Natex06.nsp*.

Uploading Data with Labels from dBase or Lotus

Column labels are not uploaded.

When you upload a Lotus file with the extension *wkl*, Entire Connection assumes that the first record contains column labels and thus does not upload it.

When you upload a dBase file with the extension *dbf*, all records are uploaded.

Uploading HTML Files

In addition to regular HTML files, you can also upload HTML files that have been edited using Excel (up to Excel 2007). Because of the new file format of HTML files that are created with Excel 2007 or above, these HTML files cannot be uploaded.

Supported encodings for upload are ISO-8859 and Windows Encoding. Make sure that you have defined one of these encodings in Excel. Starting with Excel 2000, you can check/set the encoding as follows: from the **Tools** menu, choose **Options**. In the resulting dialog box, select the **General**

tab and then choose the **Web Options** button. In the resulting dialog box, select the **Encoding** tab. You can now select the required encoding from a drop-down list box.

It is also possible to upload HTML files so that they are written in NCD (ASCII) format. See the description of the **Data Transfer** property page for further information.

Uploading XML Files

You can upload your **downloaded XML files** or create XML files for upload yourself. The upload procedure checks for data in the `<RECORD>` elements. The names of the child elements are not considered. Thus, the following two sample uploads produce identical results:

```
<NATURAL_DOWNLOAD>
  <RECORD>
    <NAME>
      <FIRSTNAME>
         <FIRST>KLAUS</FIRST>
         <SECOND>OTTO</SECOND>
      </FIRSTNAME>
    </NAME>
    <AGE>88</AGE>
  </RECORD>
</NATURAL_DOWNLOAD>
```

```
<NATURAL_DOWNLOAD>
 <RECORD>
  <DATA>KLAUS</DATA>
  <DATA>OTTO</DATA>
  <DATA>88</DATA>
 </RECORD>
</NATURAL_DOWNLOAD>
```

The character set to be used is defined in the session properties. It is also possible to upload XML files so that they are written in NCD (ASCII) format. See the description of the **Data Transfer** property page for further information.

Uploading Excel Files

When you upload data from an Excel file (*xls* or *xlsx*, depending on the installed Excel version), make sure that the option in the session properties which defines whether the first row in Excel is used for the Natural field names corresponds to the contents of your Excel file. Otherwise an error message appears and the upload does not complete. An error message appears in the following cases:

- When there are no column labels (Natural field names) in the Excel sheet, but the above mentioned option has been switched on.
- When there are column labels (Natural field names) in the Excel sheet, but the above mentioned option has been switched off.

See the description of the **Data Transfer** property page for further information.

Uploading ASCII Data

When the format used for uploading is too small for the data in the file to be uploaded, the excess data for each record are truncated (i.e. they are not uploaded).

Restrictions for Unicode Data

The following operations are not supported for Unicode data:

- Writing Unicode data to an NCC file and reading Unicode data from an NCC file.
- Sending Unicode data directly to the printer.
- Uploading binary data from UTF-16 files.

9 Transferring Natural Objects

- Transfer Tasks 100
- Executing a Transfer Task 102

Using the Natural Object Handler, you can transfer data from and to different platforms.

Transfer Tasks

Transfer tasks can be used on all platforms. A prerequisite is that Natural Version 4.1 or above is installed.

The files created with the transfer tasks can be used directly with the Object Handler (and vice versa).

When Natural Security is active, the security settings for the Object Handler apply.

Natural work file 7 is always used.

The following Entire Connection tasks are available:

Task	Description
DSYSTRANS	Download Natural source code to a PC.
NATLOADD	Load a Natural DDM from a PC. Natural and the TP monitor must be in lower-case mode.
NATLOADO	Load a Natural object from a PC. Natural and the TP monitor must be in lower-case mode.
NATUNLDD	Unload a Natural DDM to a PC. Natural and the TP monitor must be in lower-case mode.
NATUNLDO	Unload a Natural object to a PC. Natural and the TP monitor must be in lower-case mode.
USYSTRANS	Upload Natural source code from a PC.

The above sample transfer tasks show how the Object Handler can be used. These tasks can be modified according to your requirements. See the Natural documentation for further information on the Object Handler.

The syntax for the transfer tasks is described below.

DSYSTRANS: Download Natural Source Code to a PC

```
DSYSTRANS parm1 parm2 parm3
```

parm1 is the Natural program name. You can also use the wildcard characters "*" and "?" to specify several program names.

parm2 is the name of the Natural library in which the source code is stored.

parm3 is the PC file name without the extension. The extension *.trn* is automatically added.

NATLOADD: Load a Natural DDM from a PC

```
NATLOADD parm1 parm2
```

parm1 is the DDM name. You can also use the wildcard characters "*" and "?" to specify several DDM names.

parm2 is the PC file name without the extension. The extension *.sag* is automatically added.

NATLOADO: Load a Natural Object from a PC

```
NATLOADO parm1 parm2 parm3 parm4
```

parm1 is the object name. You can also use the wildcard characters "*" and "?" to specify several object names.

parm2 is the name of the Natural library in which the object is stored.

parm3 is the name of the Natural library into which the object is loaded.

parm4 is the PC file name without the extension. The extension *.sag* is automatically added.

NATUNLDD: Unload a Natural DDM to a PC

```
NATUNLDD parm1 parm2
```

parm1 is the DDM name. You can also use the wildcard characters "*" and "?" to specify several DDM names.

parm2 is the PC file name without the extension. The extension *.sag* is automatically added.

NATUNLDO: Unload a Natural Object to a PC

```
NATUNLDO parm1 parm2 parm3 parm4
```

parm1 is the object name. You can also use the wildcard characters "*" and "?" to specify several object names.

parm2 is the name of the Natural library in which the object is stored.

parm3 is the name of the Natural library into which the object is unloaded.

parm4 is the PC file name without the extension. The extension *.sag* is automatically added.

USYSTRANS: Upload Natural Source Code from a PC

```
USYSTRANS parm1 parm2 parm3
```

parm1 is the Natural program name. You can also use the wildcard characters "*" and "?" to specify several program names.

parm2 is the name of the Natural library as used by DSYSTRANS.

parm3 is the PC file name without the extension. The extension *.trn* is automatically added.

Executing a Transfer Task

To execute a transfer task, you must first establish the host connection.

To establish the host connection and execute a transfer task

1. Log on to the host using Entire Connection.
2. Start a Natural session on the host.
3. If PC=ON was not specified when Natural was started, you must issue the **Natural terminal command %+**.
4. Go to the NEXT prompt.

   If the main menu is shown, enter MAINMENU OFF in the Natural command line to go to the NEXT prompt.
5. In Entire Connection, execute a transfer task.

   There are several possibilities to execute a transfer task:

   - Select the transfer task from the **task list**.
   - **Execute a procedure file** which invokes a transfer task. During installation, example procedure files are copied to your hard disk (if specified).
   - Enter the name of the transfer task and, if applicable, all required parameters in the **command line**.

10 Sending Commands to the PC

- Asynchronous Processing 104
- Synchronous Processing 104

The Natural statement `DOWNLOAD PC FILE` with the `COMMAND` clause can be used to send Entire Connection commands from Natural to the PC. See the Natural documentation for detailed information on this statement.

See the section *Commands* for descriptions of all Entire Connection commands.

Asynchronous Processing

Asynchronous processing is started when you issue the `DOWNLOAD PC FILE` statement as follows:

```
DOWNLOAD PC FILE work-file-number COMMAND command-name
```

where *`command-name`* can be any command which can also be entered in the **command line** of Entire Connection.

Processing on the host continues when the command has been recognized as being valid and when it has been started.

Synchronous Processing

This feature is available starting with Natural Version 4.1 for Mainframes and Natural Version 6.1 for UNIX.

Synchronous processing is started when you issue the `DOWNLOAD PC FILE` statement with the `SYNC` option:

```
DOWNLOAD PC FILE work-file-number COMMAND command-name SYNC
```

Processing on the host is not continued until the command has been completely executed on the PC.

When the `DOWNLOAD PC FILE` statement is used to execute a procedure file in synchronous mode, the following commands - when defined in the procedure file - cannot be executed since they require communication with the host:

```
CONNECT
DISCONNECT
EMULATE
LEARN
MONITOR
PAUSE
QA
QUIT
```

REC_BUFF
REC_SCR
REC_XFER
REVEAL
RSPMONITOR
SET (with the parameters listed below)
TOGGLE (with the parameters listed below)
TYPE
WAITFOR
WAITM

The following parameters of the commands SET and TOGGLE cannot be used in synchronous mode:

DISPLAY
EOSDELAY
EAB
INSTOGGLE
LOGOFF
LOGON
NTCON
NATFILE
NTCOFF
NUMLOCK
OEM
RESPONSE
STATUS
TCPADDR
TCPPORT

In addition, commands which use *SCREEN cannot be used in synchronous mode.

An error message occurs when one of the above commands or parameters is detected.

During the execution of a command or procedure file, the SYNC option can only be used in conjunction with the command SET PCFILE. See ***Specifying a File Name Using the SET Command***.

III Procedure Files

11 Procedure Files

- What is a Procedure File? 110
- Executing a Procedure File 110
- External Parameters 111
- Addressing a Host Session from a Procedure File 111
- Aborting or Terminating a Procedure File 112
- Creating a Procedure File Using Learn Mode 112
- Debugging a Procedure File 113
- Conventions 115
- Command Categories 117
- Variables in Procedure Files 121

This chapter describes how to work with procedure files.

What is a Procedure File?

Procedure files (extension *ncp*) initiate operations which are executed on the host, the PC, or both. You can create procedure files using a PC editor, a host editor or in **learn mode**.

For example, you can create a procedure file which automates the process of logging on to a host system. All required logon commands are contained in the procedure file and are executed when the procedure file is invoked.

Example - logging on to a VM mainframe system:

```
:LOGON
   TYPE CR
   IF *SCREEN NE 'CP READ' GOTO ERROR
   TYPE 'LOGON' *USERID1 CR
   TYPE *PASSWORD1 CR
   SET #CNT1 0
:VM
   IF *SCREEN EQ 'VM/SP RELEASE 4' GOTO NAT
   INCR #CNT1
   IF #CNT1 GT 10 GOTO ERROR
   PAUSE 200
   GOTO VM
:NAT
   TYPE 'NAT' CR
   GOTO FINI
:ERROR
   MSG 'CANNOT LOGON - TRY LATER'
:FINI
   EXIT
```

Sample procedure files are copied to your hard disk during installation (if specified).

Executing a Procedure File

You can execute a procedure file in the following ways:

- Select the procedure file from the **task list**.
- Enter the name of the procedure file and, if applicable, all required parameters in the **command line**.
- Execute a procedure file which invokes another procedure file, see below.

- Press the key or key combination that is defined for the procedure file.

To invoke a procedure file from another procedure file

1. Open the procedure file from which you want to invoke another procedure file.
2. Insert the `EXECUTE` command at the appropriate location:

```
EXECUTE filename
```

3. Save your modifications.
4. Execute the modified procedure file.

External Parameters

You can pass up to 9 external parameters to a procedure file. The external parameters can be checked within the procedure file, using the appropriate parameter variables.

Example:

The following command executes the procedure file `MYPROG` with the parameters `1234`, `abc` and `AbC`:

```
MYPROG 1234 abc 'AbC'
```

See also: *Task Parameters* in the *Overview of Object Properties*.

Addressing a Host Session from a Procedure File

A host session can be addressed from a procedure file in the following ways:

- **Explicitly connect to a specific host session**
  Use the `CONNECT` command:

```
CONNECT sessionname
```

- **Implicitly connect to the active host session**
  If you execute a procedure file while a host sessions is open.

- **Implicitly connect to the default host session**
  If you execute a procedure file while no host session is open, the default session is automatically opened when the first statement addressing a host session is encountered. The default session is specified in the user properties. Commands that require open sessions are, for example, `TYPE` and `IF *SCREEN`.

Aborting or Terminating a Procedure File

The execution of a procedure files is aborted or terminated in the following cases:

- When you choose the **Cancel** button in a dialog box that was invoked by the procedure file.
- When a non-recoverable error occurs during procedure file processing (for example, a syntax or logic error).
- When the maximum host response time is exceeded.
- When the `EXIT` or `CANCEL` command is issued from within the procedure file.
- When a normal exit from the procedure file occurs (i.e. when the last statement is executed).

You can also abort the currently active procedure file as described below.

To abort the currently active procedure file

- From the **Utilities** menu, choose **Cancel Procedure**.

  Or:

  Choose the following toolbar button:

Creating a Procedure File Using Learn Mode

In learn mode, all keyboard input entered in a host session is stored in a procedure file. The commands in the procedure file are automatically executed when you execute the procedure file.

For example, you can use learn mode to record all keyboard input required to log on to the host.

Caution: Though learn mode creates procedure files that can be executed successfully, they are not robust as long as you do not add error and exception handling.

To invoke learn mode

1 Open a host session.

2 Use the key combination CTRL+L to invoke learn mode.

CTRL+L is the default. You can also define another key combination.

The **Learn** dialog box appears.

3 Select the desired procedure file from the list box, or specify the name of the (new) procedure file with the extension *ncp* in the **File name** text box.

If you specify an existing procedure file, it will be overwritten by the new procedure.

4 Choose the **OK** button.

Learn mode is now active and all keyboard input is recorded.

To disable learn mode

- Use the key combination CTRL+L (default) to switch off learn mode.

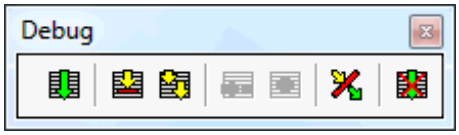
  The resulting procedure file contains all of your keyboard input. You can modify this procedure file using an ASCII editor (for example, to include check routines or logical queries).

Debugging a Procedure File

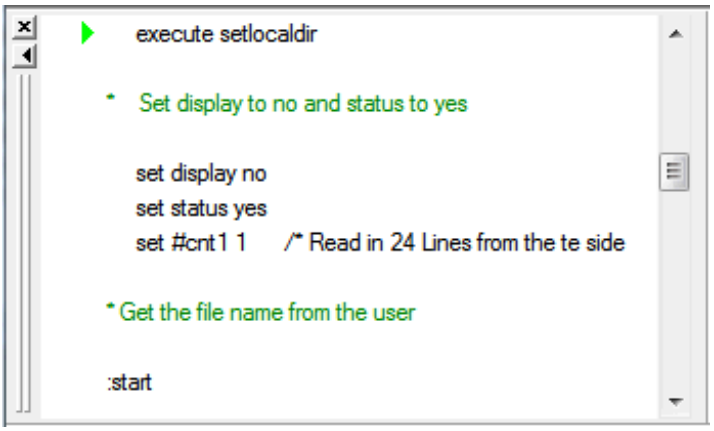
When debugging a procedure file, the following additional elements are shown in the terminal application window:

- The **Debug** menu. This menu is only available as long as you debug a procedure file. It provides the same commands as the debug toolbar.
- A debug toolbar. This toolbar is only available as long as you debug a procedure file.

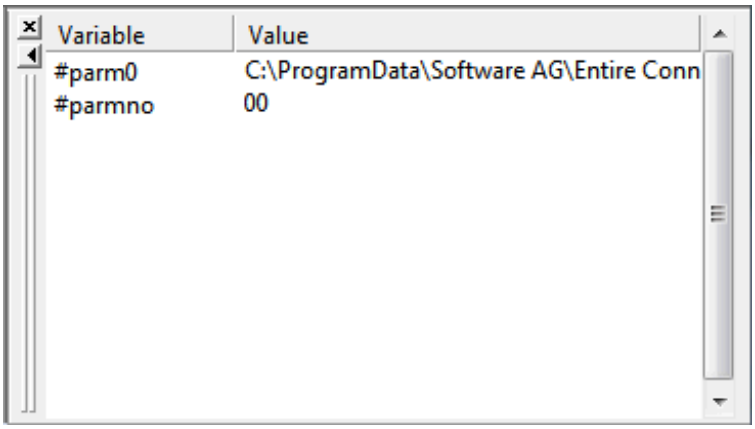
  You can move it to the application window (e.g. below the menu bar or next to the terminal emulation screen) so that it is no longer shown in a window. To prevent docking, press CTRL while moving the window.



- A procedure window on the left. The display starts with the first executable source text line which is indicated by an arrow.



- A variable window on the right, providing a list of all variables that are used in the procedure file.



You can modify the size of the individual windows by dragging a window border with the mouse. You can also display these windows as dockable windows.

Using the following commands or buttons in the debug toolbar, you debug the procedure file.

	CTRL+SHIFT+R	**Run Procedure**.
	CTRL+SHIFT+S	**Step Procedure**.
	CTRL+SHIFT+A	**Animate Procedure**.
	CTRL+SHIFT+P	**Toggle Breakpoint**.
	CTRL+SHIFT+B	**Break Procedure**.
	CTRL+SHIFT+E	**Set Echo Time**. The **Set Echo Time** dialog box appears, in which you can specify the desired delay in milliseconds before execution of the next statement. See the system variable `ECHO`.
	CTRL+SHIFT+T	**Terminate Debug**.

To debug a procedure file

1. From the **Utilities** menu, choose **Task List**.

   The **Select Task or Procedure** dialog box appears.
2. Select the **Procedures** option button.
3. Select the procedure to be debugged.
4. Choose the **Debug** button.
5. Use the **Debug** menu or debug toolbar to debug the procedure file.
6. When debugging has been completed, terminate the debug session using the corresponding command.

Conventions

The following rules apply:

- Each line in a procedure file can consist of up to 127 characters.
- A line can contain only one command and related operands.

  Exception: the commands `IF`, `IFNOT` und `WAITFOR`.
- Statements can be indented for readability.

  Exception: tags and comment lines must start in the first column.
- Elements such as tags and commands are not case-sensitive. Character strings can be case-sensitive.

- The limit for the source code is 64 KBytes, including comments. If this size is exceeded, an error message appears and the procedure aborts.

A line can consist of the elements described below.

- Command String
- Tag
- Comment
- Blank Line

Command String

A character string must be enclosed with quotation marks if case-sensitivity is required or if there are embedded blanks. You can either use apostrophes or double quotation marks, as long as they are paired consistently. For example:

```
'This string is a valid single quoted string'
'This string is "also" valid'
"This string is a valid double quoted string"
"This string is 'also' valid"
"This string is invalid because the double quotation marks are not paired'
```

Entire Connection expects character strings to be in the Windows character set (ANSI).

Tag

A tag is a branch point within a procedure file. The name of a tag must always begin with a colon and can be up to 127 characters in length. A tag that is not unique results in an error message.

Examples for tags:

```
:START
:LOGON-ERROR
:QUIT
```

`$TIMEOUT` and `$ESC` are tags with specific meanings (see below).

$TIMEOUT

When the `$TIMEOUT` tag is used in a procedure file and a timeout condition occurs (for example, when the host does not respond within the defined period of time), control is automatically passed to the statement that follows the tag.

The system variable `RESPONSE` defines how long Entire Connection is to wait for a response from the host before processing is to continue with the next statement after the tag.

`$ESC`

When the `$ESC` tag is used in a procedure file and the user chooses **Cancel Procedure** from the **Utilities** menu (or the corresponding toolbar button), control is automatically passed to the statement that follows the tag.

If the `$ESC` tag is not used in a procedure file and the user chooses the **Cancel Procedure** command, procedure file processing is aborted.

If a procedure file is executing a task that can be aborted, the following applies:

- When the task is aborted, control is passed back to the procedure file.
- When the user then chooses the **Cancel Procedure** command, control is passed to the `$ESC` tag.

Comment

A comment line starts with an asterisk (*) in the first column. For example:

```
* This is a sample comment line
*
* The line before this one is also a comment line
```

A comment can also be inserted after a statement. In this case, it starts with a slash followed by an asterisk (/*). For example:

```
SET LOGON NO /* this is a comment after a statement
:ERROR       /* this is a comment after a tag
```

Blank Line

You can use blank lines anywhere within a procedure file. Entire Connection ignores them.

Command Categories

The commands that can be used in a procedure file can be divided into the categories described below.

- Host Communication
- Processing
- Input/Output
- Task Management
- Operating System Functions

- Environment Control

Host Communication

Using the following commands, you can maintain host sessions from within a procedure file:

Command	Description
CONNECT	Open a host session.
DISCONNECT	Close a host session.
EMULATE	Switch to terminal emulation mode.
QA	Record sessions, terminal emulation screens and user input in Entire Test Client format to disk.
REC_BUFF	Record data untranslated from the terminal emulation buffer to disk.
REC_SCR	Record terminal emulation screens to disk.
REC_XFER	Record data transfer buffers to disk.
REVEAL	Display the field attributes of the 3270 emulation and the value of the ASCII character at the current cursor position.
RSPMONITOR	Switch the response time monitor for terminal emulation on and off.
TYPE	Send simulated keyboard input to the host or PC.

Important: The REC_BUFF and REC_XFER commands are used for Entire Connection problem resolution and should only be used with the assistance and direction of your technical support.

Processing

Using the following commands, you can determine the processing logic for a procedure file. For example, depending on the data transmitted from the host, you can execute different commands.

Command	Description
DECR	Subtract 1 from the global and local counter variables or screen position variables.
ELAPSETIME	Calculate the difference, in seconds, between two date and time stamps.
EXIT	Leave a procedure file and return to the previous procedure file or to Entire Connection.
GOTO	Branch to another location in the procedure file.
IF/IFNOT	Check a condition.
INCR	Add 1 to the global and local variable counters or screen position variables.
INPUT	Prompt the user for input.
LEARN	Create a procedure file in learn mode.
PAUSE	Suspend processing for a specific period of time.

Command	Description
PERFORM	Branch to another location in the procedure file and execute the statements defined at this location.
RESET	Reset a local or global variable to zero or blank.
RETURN	Continue processing with the statement that occurs directly after the PERFORM command.
SET	Assign a value to a local, global, or system variable.
SHIFT	Shift the contents of the global or local parameter variables PARM2 through PARM9 down into PARM1 through PARM8 in order to set PARM9 to a null value.
SLEEP	Suspend procedure file processing for a specific period of time.
TOGGLE	Toggle between two possible states of a system variable.
WAIT	Suspend processing of a procedure file until the user presses a key.
WAITFOR	Check a condition on the next terminal emulation screen that will be sent by the host.
WAITM	Suspend processing of a procedure file for a specific period of time.
WAITUNTIL	Suspend processing of a procedure file until a specific date and time.

Input/Output

Using the following commands, you can read files and write data to files.

Command	Description
CLOSE	Close a file.
OPEN-I	Open a file from which data are read (input file).
OPEN-O	Open a file into which data are written (output file).
READ	Read data from an input file.
WRITE	Write data to an output file.

Task Management

Using the following commands, you can schedule or execute other procedure files or tasks. You can break down complex operations into more manageable, separate procedure files, or execute a number of operations based upon external variables.

Command	Description
EXECTASK	Execute an Entire Connection task.
EXECUTE	Execute a procedure file.
SCHEDTOP	Schedule further tasks or procedure files on a first-in-first-out (FIFO) basis.
SCHEDULE	Schedule further tasks or procedure files on a last-in-first-out (LIFO) basis.

Operating System Functions

Using the following commands, you can invoke operating system functions from within a procedure file.

Example: using a procedure file, you can download data from the host in regular intervals. In order to write this data to a PC file, the procedure file must rename or erase the PC file that was created during the previous download. Otherwise, manual modifications are required prior to each execution of the procedure file.

Command	Description
CD	Change the current directory on the current drive.
CHDRIVE	Change the current drive.
CHMOD	Change the file attributes.
DOS	Execute a DOS command.
DOSDIR	Display directory information.
ERASE	Erase a file.
MD	Make a directory.
OS	Same as the DOS command.
POPDIR	Return to the position in the directory hierarchy that was saved using the PUSHDIR command.
PUSHDIR	Save the current position in the directory hierarchy.
RD	Remove a directory.
RENAME	Rename or move a file.

Environment Control

The following commands are available from within a procedure file:

Command	Description
BEEP	Sound the PC alarm.
CANCEL	Abort processing of a procedure file.
LOG	Write a message to the log file.
MSG	Display a message.
QUIT	Close an Entire Connection terminal.

Variables in Procedure Files

Procedure files can contain **variables**, where the values of the variables are determined either by the user or by the system.

For complex operations that involve nested procedure files, global variables that are accessed from multiple procedure files can be defined. A global variable can, for example, pass a return code from one procedure file to another. This prevents processing of subsequent program steps if a preceding procedure file has not been completed successfully.

IV Variables

This part describes all variables available for inclusion in procedure files. A variable is replaced by a parameter value at execution time, thus allowing flexible use of procedure files.

There are 4 types of variables. All variables, except system variables, start with a special character.

Variable Type	Special Character
Dynamic Variable	*
Global Variable	+
Local Variable	#
System Variable	

The values for global, local and system variables are defined using the `SET` and `RESET` commands. The values for dynamic variables are determined by the system settings of the operating system and Entire Connection.

Examples

The following examples demonstrate the usage of the local variables `#FILEDRIVE` and `#FILEPATH` in a procedure file. The following expression contains a mandatory operand with optional parameters:

```
{[drive:]\path\...}
```

Example 1

Specification in program:	`#FILEDRIVE ':' #FILEPATH`
Value of `#FILEDRIVE`:	`C`
Value of `#FILEPATH`:	`\SAG`
Result:	`C:\SAG`

Example 2

Specification in program:	`\SAG #FILEPATH`
Value of `#FILEPATH`:	`\PCC`
Result:	`\SAG\PCC`

12 Dynamic Variables

- List of Dynamic Variables 126

Dynamic variables start with an asterisk (*).

```
*API        *HOUR         *RSPTOTAL
*COMMTYPE   *MINUTE       *SCREEN
*CURCOL     *MONTH        *SECOND
*CURPOS     *PASSWORD     *SYSTEM
*CURROW     *PASSWORD1-4  *TE
*DATE       *RSPAVG       *TIME
*DAY        *RSPCOUNT     *UA
*DAYOFWEEK  *RSPMAX       *USERID
*DISKSPACE  *RSPMIN       *USERID1-4
*DRIVE      *RSPTIME      *YEAR
```

Dynamic variables cannot be modified. The values are determined by the system settings of the operating system and Entire Connection. Dynamic variables can be used with any command if the command syntax allows it.

List of Dynamic Variables

*API

Sample value: YES

This variable indicates whether a procedure file is being executed, using the Entire Connection application programming interface.

*COMMTYPE

Sample value: MyHost

This variable contains the name of the session which is addressed by the procedure file. When no session is currently active, this variable contains the name of the user's default session.

*CURCOL

Sample value: 80

This variable contains a numeric value between 1 and the maximum line size. The value indicates the column on the terminal screen in which the cursor is located.

*CURPOS

Sample value: 1919

This variable contains a numeric value between 0 and the maximum screen size. The value indicates the current position of the cursor on the terminal screen.

*CURROW

Sample value: 24

This variable contains a numeric value between 1 and the maximum number of rows. The value indicates the row on the terminal screen in which the cursor is located.

*DATE

Sample value: 1998/12/21

This variable contains the current date.

*DAY

Sample value: 21

This variable contains the number of the current day.

*DAYOFWEEK

Sample value: MONDAY

This variable contains the name of the current day of the week.

*DISKSPACE

Sample value: 105

This variable contains the available disk space on the current drive in bytes.

*DRIVE

Sample value: C

This variable contains the current drive.

*HOUR

Sample value: 13

This variable contains the current hour.

*MINUTE

Sample value: 03

This variable contains the current minute.

*MONTH

Sample value: 03

This variable contains the current month.

*PASSWORD

Sample value: ABCDEF

This variable contains a password which may be 1 to 16 characters long. The password is required to access Entire Connection.

*PASSWORD1 - 4

Sample value: SYSTEM

Each of the variables `*PASSWORD1` to `*PASSWORD4` may contain a password which may be 1 to 16 characters long. These passwords may be required when logging on to different host systems or applications. The values are defined on the **Procedure** property page of the **User Properties** dialog box.

Note: These variables are actually system variables which can be modified during runtime, using the `SET` command.

*RSPAVG

Sample value: 0.44

This variable contains the average host response time in seconds.

*RSPCOUNT

Sample value: 128

This variable contains the number of processed transactions.

*RSPMAX

Sample value: 5.32

This variable contains the maximum host response time in seconds.

*RSPMIN

Sample value: 0.87

This variable contains the minimum host response time in seconds.

*RSPTIME

Sample value: 1.55

This variable contains the current host response time in seconds.

*RSPTOTAL

Sample value: 123.75

This variable contains the total host response time in seconds.

*SCREEN

This variable contains the current terminal emulation screen. You can only access `*SCREEN` or a specific position within `*SCREEN` using the commands `SET` and `IF`.

*SECOND

Sample value: 15

This variable contains the current second.

*SYSTEM

Sample value: MS/WIN

This variable contains the value MS/WIN. It guarantees compatibility between Entire Connection versions on different platforms.

*TE

Sample value: YES

This variable indicates whether the procedure file is being executed in terminal emulation mode.

*TIME

Sample value: 13:03:15

This variable contains the current time.

*UA

Sample value: YES

This variable indicates whether the procedure file is being executed by the Unattended Workstation utility.

*USERID

Sample value: SABCD

This variable contains a user ID which may be 1 to 16 characters long. The user ID is required to access Entire Connection.

*USERID1 - 4

Sample value: SBCDE

Each of the variables `*USERID1` to `*USERID4` may contain a user ID which may be 1 to 16 characters long. These user IDs may be required when logging on to different host systems or applications. The values are defined on the **Procedure** property page of the **User Properties** dialog box.

Note: These variables are actually system variables which can be modified during runtime, using the `SET` command.

*YEAR

Sample value: 1998

This variable contains the current year.

13 Global Variables

- Global Variables with Value Type Integer 134
- Global Variables with Value Type String 137

Global variables start with a plus sign (+). There are two types of global variables:

Global variables with value type Integer:

```
+CNTO-9      +FILEHOUR    +FILETYPE
+COL         +FILEMINUTE  +FILEYEAR
+ELAPSETIME  +FILEMONTH   +LENGTH
+FILEDATE    +FILESECOND  +PARMNO
+FILEDAY     +FILESIZE    +ROW
```

Global variables with value type String:

```
+ENVIRONMENT  +FILEMODE  +FILETIME
+FILEDRIVE    +FILENAME  +PARMO-9
+FILEEXT      +FILEPATH  +RC
+FILEINFO     +FILESPEC
```

Mnemonic names have been chosen for the variables. For example, it is good policy to use the integer variable `+COL` to store a column number. However, it can also be used for totally different purposes.

You can access global variables from within a procedure file. All global variables, except `+RC`, are blank. The values are defined by the programmer in the procedure file or by the user in the command line, using the commands `SET` and `RESET`. Global variables, unlike **local variables**, are not automatically reset when the procedure file terminates. The value you set for a global variable will also be available to subsequent procedure files.

For each local variable there is a global variable with the same name.

Global Variables with Value Type Integer

Variables with value type Integer may contain a positive number between 0 and 32767.

+CNT0 - 9

You can use the variables `+CNT1` through `+CNT9` as counters.

+COL

You can use this variable to store the column number of the terminal emulation screen.

+ELAPSETIME

You can use this variable to store the difference between two date and time stamps.

+FILEDATE

You can use this variable to store the date on which a file was created or updated.

+FILEDAY

You can use this variable to store the day of the month on which a file was created or updated.

+FILEHOUR

You can use this variable to store the hour at which a file was created or updated.

+FILEMINUTE

You can use this variable to store the minute at which a file was created or updated.

+FILEMONTH

You can use this variable to store the month in which a file was created or updated.

+FILESECOND

You can use this variable to store the second at which a file was created or updated.

+FILESIZE

You can use this variable to store the size, in bytes, of the current file.

+FILETYPE

You can use this variable to store the file attributes. Valid values are:

01	read-only
02	hidden
03	read-only and hidden
04	system
05	read-only and system
06	hidden and system
07	read-only, hidden and system
10	directory
20	archive

Using combinations of the above list, you can also define attributes 21 through 27. For example, 25 contains the following values: archive, system and read-only (20+4+1).

+FILEYEAR

You can use this variable to store the year in which a file was created or updated.

+LENGTH

You can use this variable to store the length of data for which a search is to be performed or was performed.

+PARMNO

You can use this variable to store the number of `+PARM0` through `+PARM9` variables that were passed to a procedure file during processing.

+ROW

You can use this variable to store the row number of the terminal emulation screen.

Global Variables with Value Type String

Variables with value type String may contain any character or a group of characters. The maximum length is 254 characters.

+ENVIRONMENT

You can use this variable to store the value of an environment parameter. This variable is blank unless the `SET` command is used. The syntax of the `SET` command must include the name of the environment parameter from which the value is to be obtained. Example:

```
SET +ENVIRONMENT PATH
```

+FILEDRIVE

You can use this variable to store the drive on which a file is located.

+FILEEXT

You can use this variable to store the extension of a file name.

+FILEINFO

You can use this variable to store the following file information: name, extension, size, attributes, and the day and time the file was created or updated.

+FILEMODE

You can use this variable to store the file attributes. Valid values are:

A	archive
D	directory
H	hidden
R	read-only
S	system

+FILENAME

You can use this variable to store the name of a file.

+FILEPATH

You can use this variable to store the directory or directories in which a file is located.

+FILESPEC

You can use this variable to store the path for a file.

+FILETIME

You can use this variable to store the time at which a file was created or updated.

+PARM0 - 9

You can use the variables `+PARM0` through `+PARM9` to read and write data, and to pass data to a procedure file (or to a group of nested procedure files).

+RC

This variable is filled by the `EXIT` command. It contains a programmer-defined return value which, for example, indicates the result of a procedure file.

14 Local Variables

- Local Variables with Value Type Integer 140
- Local Variables with Value Type String 143

Local variables start with a hash sign (#). There are two types of local variables:

Local variables with value type Integer:

```
#CNT0-9      #FILEHOUR    #FILETYPE
#COL         #FILEMINUTE  #FILEYEAR
#ELAPSETIME  #FILEMONTH   #LENGTH
#FILEDATE    #FILESECOND  #PARMNO
#FILEDAY     #FILESIZE    #ROW
```

Local variables with value type String:

```
#ENVIRONMENT  #FILEMODE  #FILETIME
#FILEDRIVE    #FILENAME  #PARM0
#FILEEXT      #FILEPATH  #PARM1-9
#FILEINFO     #FILESPEC  #RC
```

The value of a local variable is reset when the procedure file terminates. Using the `SET` command, you can define values for local variables. Many Entire Connection commands save their result(s) in local variables.

Local Variables with Value Type Integer

Variables with value type Integer may contain a positive number between 0 and 32767.

#CNT0 - 9

You can use the variables `#CNT1` through `#CNT9` as counters.

#COL

This variable is set by the commands `IF` and `WAITFOR`. It then contains a number between 1 and the maximum line length. This number indicates the column number of the terminal emulation screen.

#ELAPSETIME

This variable is set by the `ELAPSETIME` command. It then contains the difference between two date and time stamps.

#FILEDATE

This variable is set by the `DOSDIR` command. It then contains the date on which a file was created or updated.

#FILEDAY

This variable is set by the `DOSDIR` command. It then contains the day of the month on which a file was created or updated.

#FILEHOUR

This variable is set by the `DOSDIR` command. It then contains the hour at which a file was created or updated.

#FILEMINUTE

This variable is set by the `DOSDIR` command. It then contains the minute at which a file was created or updated.

#FILEMONTH

This variable is set by the `DOSDIR` command. It then contains the number of the month in which a file was created or updated.

#FILESECOND

This variable is set by the `DOSDIR` command. It then contains the second at which a file was created or updated.

#FILESIZE

This variable is set by the `DOSDIR` command. It then contains the size, in bytes, of the current file.

#FILETYPE

This variable is set by the `DOSDIR` command. It then contains the file attributes. Valid values are:

01	read-only
02	hidden
03	read-only and hidden
04	system
05	read-only and system
06	hidden and system
07	read-only, hidden and system
10	directory
20	archive

Using combinations of the above list, you can also define attributes 21 through 27. For example, 25 contains the following values: archive, system and read-only (20+4+1).

#FILEYEAR

This variable is set by the `DOSDIR` command. It then contains the year in which a file was created or updated.

#LENGTH

You can use this variable to store the length of data for which a search is to be performed or was performed.

#PARMNO

This variable contains the number of `#PARM0` through `#PARM9` variables that were passed to a procedure file during processing.

#ROW

This variable is set by the commands `IF` and `WAITFOR`. It then contains a number between 1 and the maximum number of rows. This number indicates the row number of the terminal emulation screen.

Local Variables with Value Type String

Variables with value type String may contain any character or a group of characters. The maximum length is 254 characters.

#ENVIRONMENT

This variable is used to obtain the value of an environment parameter. The variable is blank unless the `SET` command is used. The syntax of the `SET` command must include the name of the environment parameter from which the value is to be obtained. Example:

```
SET #ENVIRONMENT PATH
```

#FILEDRIVE

This variable is set by the `DOSDIR` command. It then contains the drive on which a file is located.

#FILEEXT

This variable is set by the `DOSDIR` command. It then contains the extension of a file name.

#FILEINFO

This variable is set by the `DOSDIR` command. It then contains the following file information: name, extension, size, attributes, and the day and time the file was created or updated.

#FILEMODE

This variable is set by the `DOSDIR` command. It then contains the file attributes:

A	archive
D	directory
H	hidden
R	read-only
S	system

#FILENAME

This variable is set by the `DOSDIR` command. It then contains the name of a file.

#FILEPATH

This variable is set by the `DOSDIR` command. It then contains the directory or directories in which a file is located.

#FILESPEC

This variable is set by the `DOSDIR` command. It then contains the path for a file.

#FILETIME

This variable is set by the `DOSDIR` command. It then contains the time at which a file was created or updated.

#PARM0

This variable is set by the `EXECUTE` command. It then contains the name of the procedure file that is currently being executed.

#PARM1 - 9

You can use the variables `#PARM1` through `#PARM9` to read and write data, and to pass data to a procedure file (or to a group of nested procedure files).

#RC

The variable `#RC` is set by the following commands:

```
CD
CHDRIVE
CHMOD
DOSDIR
ERASE
MD
OPEN-I
OPEN-O
RD
READ
WRITE
```

The `#RC` variable contains a return value which indicates whether a particular command has been executed successfully.

15 System Variables

- Types of System Variables 148
- Value Types for System Variables 149
- List of System Variables 149

System variables affect the behavior of Entire Connection during runtime.

AUDIT	INSTOGGLE	PCFILE
AUDITXFER	LOGOFF	PRINTOPTION
AUTOLOCK	LOGON	RESPONSE
BEEP	LOOPCNT	STARTTASK
CURSOR	NATFILE	STATUS
DATE	NCF	STOPTASK
DISPLAY	NTCOFF	TCPADDR
EAB	NTCON	TCPPORT
ECHO	NUMLOCK	TIME
EOSDELAY	OEM	

See also: *PASSWORD1 - 4 and *USERID1 - 4.

System variables can be modified during runtime, using the SET command. Using the TOGGLE command, you can also modify system variables of type Binary.

Types of System Variables

There are two types of system variables:

- **System variables that affect the whole Entire Connection system**
  These variables are set at startup time with values from the share file. You can modify the values during runtime, either from the command line or from a procedure file. The modifications are valid only as long as Entire Connection is active. In the corresponding object properties, variable values can be changed permanently.
- **System variables that only affect specific sessions**
  When a session is opened, these variables are filled with values from the session properties. You can modify the values while the session is open, using a procedure file. The modifications are valid only as long as the session is open. In the session properties, variable values can be changed permanently.

Value Types for System Variables

Value Type	Value
Binary	TRUE/FALSE You can also use the combinations YES/NO and ON/OFF which are synonymous with TRUE/FALSE. When you display the value of a variable, TRUE or FALSE is always shown.
Date	Date format (month, day, year and delimiter character) as defined in the user properties.
Path	{[[*drive:*]*directory*\...\] *filename*[.*extension*]}
Integer	Whole positive number.
String	Any ASCII character or a group of ASCII characters. The maximum length is 254 characters.
Task Name	Name of a valid task.
Time	Time format (hour, minute, second and delimiter character) as defined in the user properties. If the delimiter character is A, the time is displayed in the AM/PM format (for example, 01:30 PM).

List of System Variables

The default value of a system variable is indicated by underlining.

AUDIT

Affects	System
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	User properties / Procedure / Audit procedure files

AUDITXFER

Affects	System
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	User properties / Procedure / Audit file transfer

AUTOLOCK

Affects	System
Value Type	Integer
Valid Values	0 - 1440 (minutes)
Object / Page / Option	User properties / Parameters 1 / Autolock time

BEEP

Affects	System
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	User properties / Parameters 1 / PC speaker beep on errors and warnings

CURSOR

Affects	System
Value Type	Integer
Valid Values	0 - 4 1 (small)
Object / Page / Option	User properties / Parameters 1 / Cursor size

DATE

Affects	System
Value Type	Date
Valid Values	All data formats available in the user properties YMD.
Object / Page / Option	User properties / Parameters 2 / Date style

DISPLAY

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / Procedure / Display all host screens while executing a procedure file

EAB

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / Display / Ignore extended attribute colors

ECHO

Affects	System
Value Type	Integer
Valid Values	0 - 100 0 (no echo)
Object / Page / Option	Used to **debug procedure files**

EOSDELAY

Affects	Session
Value Type	Integer
Valid Values	0 - 1000 (milliseconds) 6
Object / Page / Option	Session properties / Telnet communication / End-of-screen delay

INSTOGGLE

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / Terminal / Enable insert mode toggling

LOGOFF

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / Procedure / Enable automatic logoff

LOGON

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / Procedure / Enable automatic logon

LOOPCNT

Affects	System
Value Type	Integer
Valid Values	0 - 32767 256
Object / Page / Option	User properties / Procedure / Limit loops in procedure files

NATFILE

Affects	Session
Value Type	Integer
Valid Values	1-32, if work file 1-31, if printer file 7
Object / Page / Option	Session properties / File transfer / Natural work file

NCF

Affects	System
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	User properties / Parameters 2 / Create format file

NTCOFF

Affects	Session
Value Type	Path
Valid Values	See description of **value types**
Object / Page / Option	Session properties / Procedure / Logoff procedure

NTCON

Affects	Session
Value Type	Path
Valid Values	See description of **value types**
Object / Page / Option	Session properties / Procedure / Logon procedure

NUMLOCK

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / Terminal / Allow alphanumeric input in numeric fields

OEM

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / File transfer / Use DOS character set for upload/download in NCD format

PCFILE

Affects	Session
Value Type	Integer
Valid Values	1-32, if work file 1-31, if printer file
Object / Page / Option	See the description of the `SET` command

PRINTOPTION

Affects	Session
Value Type	String
Valid Values	Size=*nn* Orient={P|L} Duplex={S|V|H}
Object / Page / Option	See the description below.

With the `PRINTOPTION` variable, you can change the print settings for download to printer and for printing from a procedure file. You can change the size of the font, the page orientation and you can define whether to print on one or both sides of the paper.

Note: The font used for printing is defined in the user properties.

The value is a string which may be composed of three elements:

Element	Description
Size=*nn*	Font size. *nn* is a number from 1 to 99.
Orient=*option*	Page orientation. Valid options are P (portrait) and L (landscape).
Duplex=*option*	Paper usage. Valid options are S (simplex), V (vertical) and H (horizontal).

The elements must be separated by one blank. There must be at least one element in the value string. The elements may be in any order.

Example - use font size 16, landscape, and vertical printing on both sides of the paper:

```
SET PRINTOPTION "Size=16 Orient=L Duplex=V"
```

To use the defaults (font size as selected in the user properties, portrait and printing on one side of the paper), you have to clear the current settings as follows:

```
SET PRINTOPTION ""
```

RESPONSE

Affects	Session
Value Type	Integer
Valid Values	1 - 1000 (1 - 1000 seconds) 1001 - 2440 (1 - 1440 minutes) 15 (seconds) **Note:** These special internal representations for seconds and minutes are always used when a procedure file is executed (for example, with the command `SET` or `WAIT`). Any

	values for seconds or minutes that you specify using the Configuration Manager are automatically transformed to the above special internal representation.
Object / Page / Option	Session properties / Procedure / Maximum host response time

STARTTASK

Affects	System
Value Type	Task Name
Valid Values	Valid Entire Connection task name
Object / Page / Option	User properties / Parameters 2 / Startup task

STATUS

Affects	Session
Value Type	Binary
Valid Values	OFF/ON
Object / Page / Option	Session properties / Procedure / Display status lines while executing a procedure file

STOPTASK

Affects	System
Value Type	Task Name
Valid Values	Valid Entire Connection task name
Object / Page / Option	User properties / Parameters 2 / Shutdown task

TCPADDR

Affects	Session
Value Type	String
Valid Values	TCP/IP address (no default) A TCP/IP address can either be a string which is resolved in the host's table (up to 128 bytes), or it can be formed as follows: *nnn.nnn.nnn.nnn* where *nnn* is a number between 000 and 255.
Object / Page / Option	Session properties / TN3270 communication or Telnet VTxxx communication / Host name / IP address

TCPPORT

Affects	Session
Value Type	Integer
Valid Values	TCP/IP port ID 23
Object / Page / Option	Session properties / TN3270 communication or Telnet VTxxx communication / Default TCP port or Custom TCP port

TIME

Affects	System
Value Type	Time
Valid Values	All time formats available in the user properties HMS:
Object / Page / Option	User properties / Parameters 2 / Time style