



Entire Screen Builder

Data Transfer

Version 4.3.1



This document applies to Entire Screen Builder Version 4.3.1 and to all subsequent releases.

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

© January 2002, Software AG
All rights reserved

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

Table of Contents

Data Transfer	1
Data Transfer	1
General Data Transfer Information	2
General Data Transfer Information	2
Natural Data Transfer	2
What can be Transferred?	3
Download	3
Upload	3
Field Formats	3
Supported File Formats	4
Supported Field Formats	5
Aborting Data Transfer	6
Specifying a File Name	7
Specifying a File Name	7
Downloading Data Using a Dialog Box	7
Uploading Data Using a Dialog Box	10
Specifying a File Name Using the SET Command	11
Creating a File Name Dynamically	11
Uploading and Downloading Data	12
Uploading and Downloading Data	12
Downloading Fixed-Length Data Records	12
Downloading Variable Data Records	12
Downloading Reports	13
Downloading Binary Data	13
Downloading Data to dBase with Labels	13
Downloading Data to HTML	13
Downloading Data to XML	14
Special Characters	15
Unnamed Fields	15
Natural Groups	16
Unidimensional Arrays	17
Two- and Three-Dimensional Arrays	18
Using Style Sheets	20
Downloading Numeric Data to ASCII Format	22
General Upload Information	23
Uploading Binary Data	23
Uploading Data with Labels from dBase	23
Uploading HTML Files	23
Uploading XML Files	24
Uploading ASCII Data	24

Data Transfer

The following sections describe how to transfer data between Natural applications on host systems and the PC using the different types of viewers that are provided with Entire Screen Builder:

- General Data Transfer Information
- Specifying a File Name
- Uploading and Downloading Data

General Data Transfer Information

This section provides the following information:

- Natural Data Transfer
 - What can be Transferred?
 - Supported File Formats
 - Supported Field Formats
 - Aborting Data Transfer
-

Natural Data Transfer

You can transfer data between Natural applications on host systems and your PC using the different types of viewers. 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 `SYSDATA WORK`, the work files listed as PC are set for data transfer with Entire Screen Builder.

Before you can use any data transfer function, you must tell Natural that you are an Entire Screen Builder 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 Screen Builder user

- Set the parameter `PC=ON` when invoking Natural.
Or enter the following Natural terminal command in any unprotected input field:
%+

Note:

The above Natural terminal command can also be sent to the host using the `TypeString` method in a script file. See the Script Files documentation for further information.

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

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

See also: Supported File Formats.

Field Formats

You can upload and download fields in the following formats:

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

See also: Supported Field Formats.

Supported File Formats

You can download data directly to and upload directly from files in any of the data formats supported by Entire Screen Builder. The format is determined by the extension of the file to be downloaded or uploaded and the format of the record in the Natural download statement.

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 data transfer properties for the required host session. See Host Sessions in the Administration documentation for further information.
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 files. However, you cannot upload column labels.
Data Interchange Format	*.dif
HTML	*.htm and *.html. This is a specific HTML format that can also be opened with Microsoft Excel 2000.
IBM-compatible COBOL	*.ncc
XML	*.xml

Supported Field Formats

The following formats are supported:

Format	Description
Ax	Alphanumeric format where x is a digit between 1 and 253.
Bx	Binary format where x is a digit between 1 and 126.
F4	Floating point format with 4 bytes.
F8	Floating point format with 8 bytes.
Ix	Integer format where x is 1, 2 or 4.
L	Logical format (true or false).
Nx.0	Numeric format where x is a digit between 1 and 29.
Nx.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.
Px.0	Packed format where x is a digit between 1 and 29.
Px.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.
Rx	Repeat the next format x times where x is a digit between 2 and 255. Example: R03A04 stands for A04A04A04.

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

The method for aborting data transfer depends on the type of viewer you are currently using.

▶ To abort data transfer with one of the GUI Viewers

- Click the **Cancel** button that is shown in the viewer window during upload and download.

For example:



▶ To abort data transfer with the Terminal Viewer

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

Or click the following toolbar button:



Specifying a File Name

This section provides the following information:

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

The format of a file is determined by its extension.

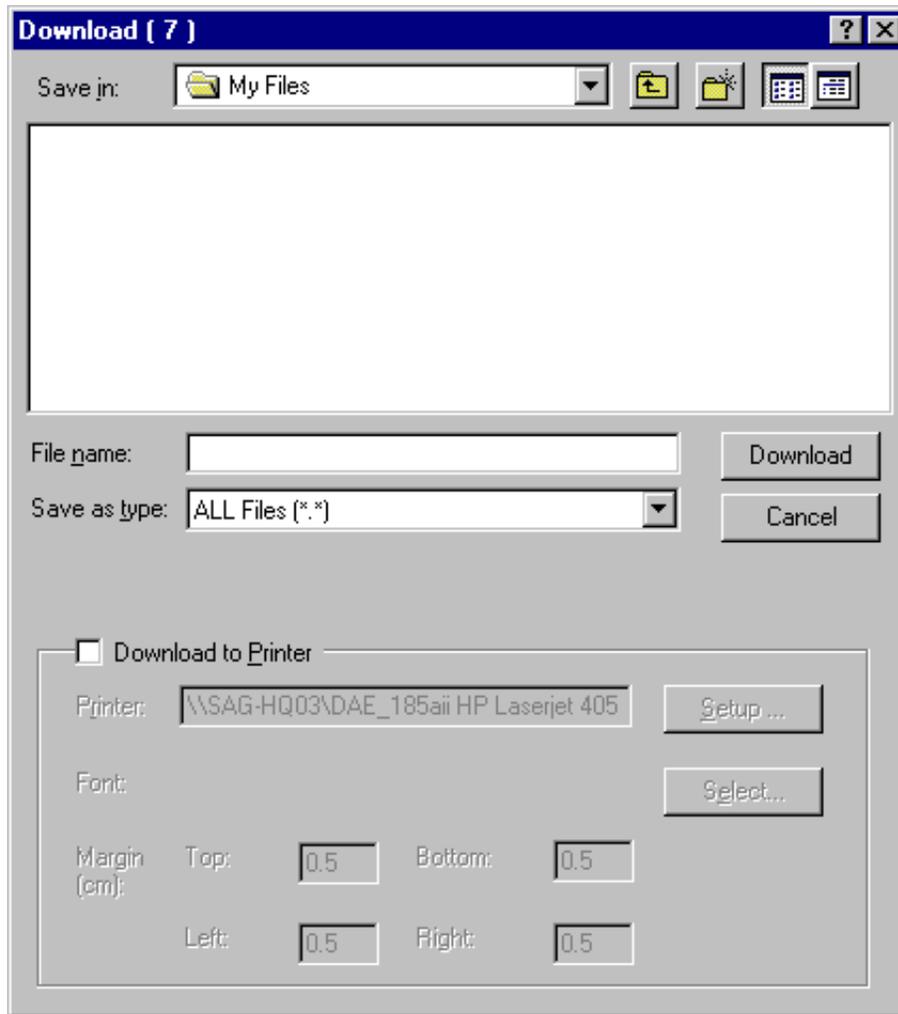
Downloading Data Using a Dialog Box

You can download data to a file or printer. To do so, start an application that uses a download command (for example, the sample Natural source program in folder "..\Software AG\Entire Screen Builder 4\samples\samplenattransfer\NATEX01.NSP").

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.

Note for the Java Viewer:

The dialog box which prompts you for a file name looks different for the Java Viewer. The dialog boxes shown in this documentation have been taken from the Web Viewer. For the Java Viewer, a "Save As" dialog box is shown. The option "Download to Printer" is not available for the Java Viewer.

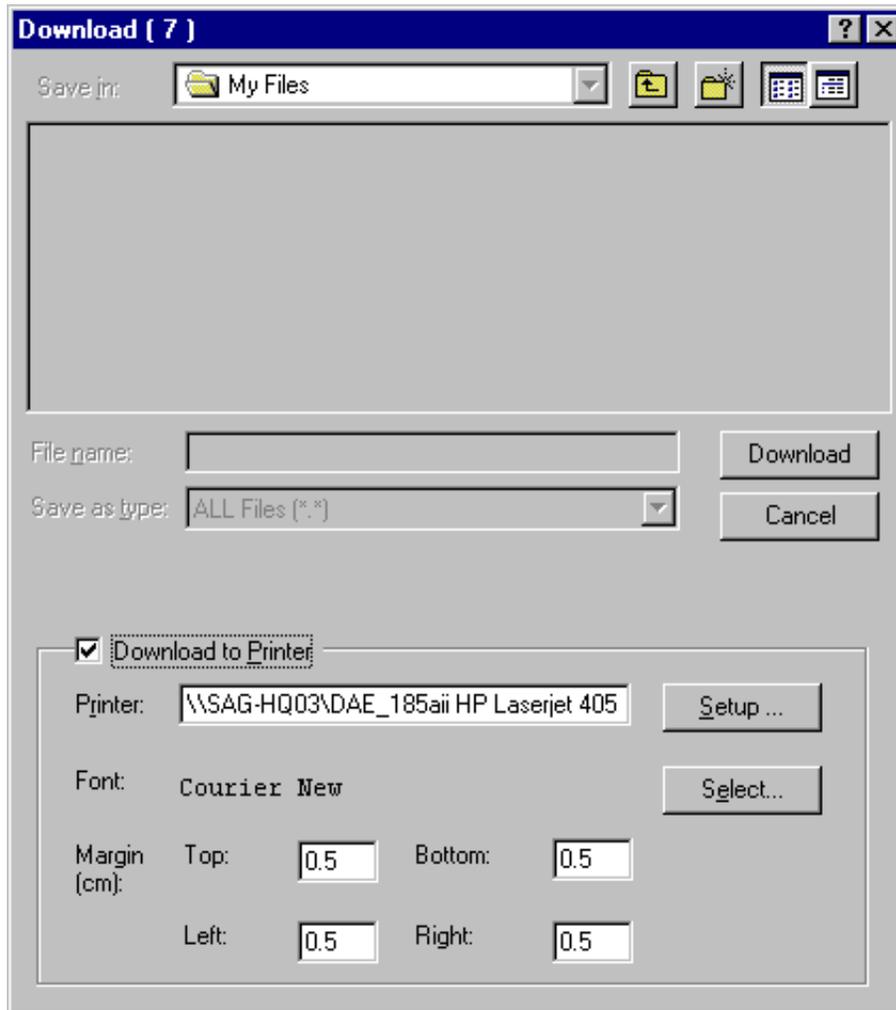


▶ To download data to a file

1. 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.
2. Choose the desired file type from the "Save as type" drop-down list box.
3. 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 "Download to Printer" check box.

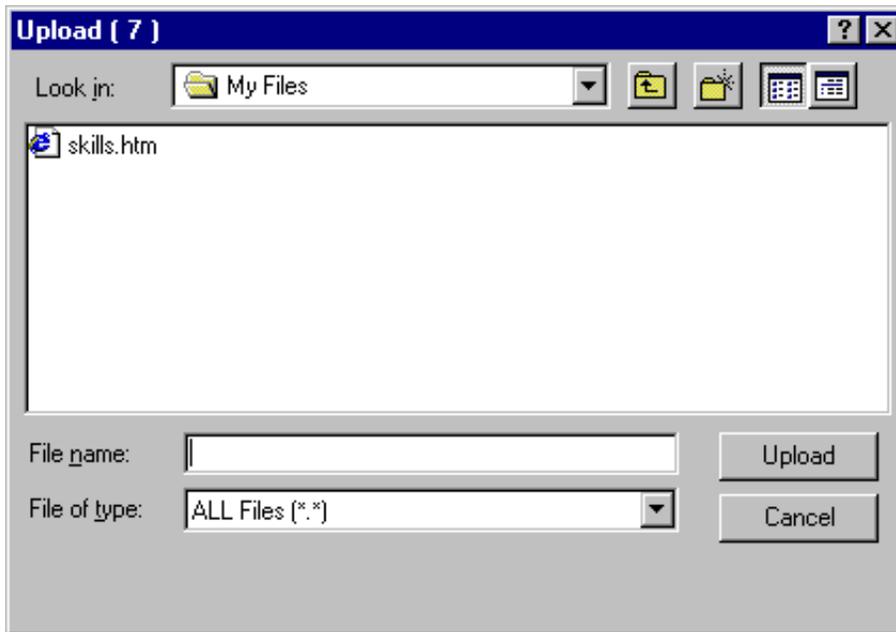


3. Optionally: choose the **Setup** button to select another printer or to define another orientation (e.g. landscape).
4. Optionally: choose the **Select** button to select another font.
5. Optionally: define the margins (top, bottom, left and right) by specifying the required values in centimeters in the corresponding text boxes.
6. Choose the **Download** button.
The data are downloaded to the specified printer.

Uploading Data Using a Dialog Box

You can upload data from a file. To do so, start an application that uses an upload command (for example, the sample Natural source program in folder "..\Software AG\Entire Screen Builder 4\samples\samplenattransfer\NATEX07.NSP").

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" dialog box, select the file to be uploaded.
2. 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.

You can issue the SET command in two different ways:

- in a Natural program with the DOWNLOAD COMMAND statement, or
- in an Entire Screen Builder script file with the `SetDataTransfer` method.

▶ 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.ext
```

```
SET PCFILE x UP DATA filename.ext
```

```
SET PCFILE y DOWN REPORT filename.ext
```

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

Do not use PRN, LPT1, LPT2 or LPT3 as the file name.

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

Note:

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

▶ To reset the number for a 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.

Creating a File Name Dynamically

You can instruct Entire Screen Builder 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 3 DOWN DATA ~~RANDOM.ext
```

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

Uploading and Downloading Data

This section provides the following information:

- Downloading Fixed-Length Data Records
- Downloading Variable Data Records
- Downloading Reports
- Downloading Binary Data
- Downloading Data to dBase with Labels
- Downloading Data to HTML
- Downloading Data to XML
- Downloading Numeric Data into ASCII Format
- General Upload Information
- Uploading Binary Data
- Uploading Data with Labels from dBase
- Uploading HTML Files
- Uploading XML Files
- Uploading ASCII Data

Sample Natural programs are copied to your hard disk during installation. They can be found in the following folder:

```
\Entire Screen Builder 4\samples\samplenattransfer
```

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 Screen Builder.

Sample Natural programs: 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. Variable data can only be downloaded into ASCII format. Leading zeros are always substituted with blanks.

Sample Natural program: 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.

Sample Natural program: 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 Screen Builder 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).

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. If the amount of data in last block equals the chosen block size, you must include an additional block containing X'FF' to mark the end of binary data.

With Natural Version 2.1.4 and above, 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 (OX09) are not recognized.

Sample Natural programs: NATEX05.NSP and NATEX06.NSP.

Downloading Data to dBase with Labels

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

Downloading Data to HTML

When you download data to an HTML file (extension "htm" or "html"), Entire Screen Builder creates an HTML format which can be opened with a browser (e.g. Microsoft Internet Explorer) and with Microsoft Excel 2000.

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 following information is provided below:

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

For further information on XML, see <http://www.w3.org/XML/> or <http://www.softwareag.com/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>
  .
  .
  .
</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 <http://www.w3.org/Style/>) and then insert the style sheet definition in the header of your downloaded XML file.

Note:

With the script file method `SetXMLStyleSheet`, a style sheet definition can be added automatically to all downloaded XML files.

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 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.

Sample Natural programs: 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 Screen Builder 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 2.1.4 and above, 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 (OX09) are not recognized.

Sample Natural programs: NATEX09.NSP, NATEX06.NSP and NATEX10.NSP.

Uploading Data with Labels from dBase

Column labels are not uploaded.

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 Microsoft Excel.

Supported encodings for upload are ISO-8859 and Windows Encoding. Make sure that you have defined one of these encodings in Excel. To check/set the encoding under Excel 2000: from the **Tools** menu, choose **Options**. In the resulting dialog box, click the "General" tab and then choose the **Web Options** button. In the resulting dialog box, click the "Encoding" tab. You can now select the required encoding from a drop-down list box.

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>
```

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).