StreamServe Persuasion SP5
Code pages and Unicode support

User Guide

Rev A
## Contents

About code pages and Unicode support .......................................................... 5  
  Code pages and Unicode support in StreamServe ....................................... 6  
    Specifying a default code page for the Design Center .............................. 8  
    Known limitations ................................................................................. 9  
  Code pages supported by the StreamServer ............................................. 9  

Specifying code pages for input data ........................................................... 13  
  Identifying the code page used to encode input data ............................... 14  
  Specifying code pages per input connector ............................................. 15  
  Specifying code pages per input type ..................................................... 17  
  Dynamically selecting code pages for the input to an Event .................. 19  

Specifying code pages for output data ....................................................... 21  

Specifying code pages for support files and logs ....................................... 23  
  Specifying code pages for table files .................................................... 23  
  Specifying code pages for function files ............................................... 24  
  Specifying code pages for the log .......................................................... 25  

Bidirectional text ....................................................................................... 27  
  Reordering visually ordered input .......................................................... 28  
    Reordering page formatted input ..................................................... 28  
    Reordering field and record based input ........................................... 28  
  Reordering output to visual order .......................................................... 29  
    Reordering page formatted output ................................................... 29  
    Reordering record based output ....................................................... 30
About code pages and Unicode support

The Unicode standard provides a code point for every character in modern use worldwide. It enables plain text data to be transported through different platforms, systems, and programs without corruption. Unicode standardizes three encoding forms and seven encoding schemes:

**Encoding forms**
- UTF-8
- UTF-16
- UTF-32

Mapping from a character set definition to the actual code units used to represent the data.

**Encoding schemes**
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16
- UTF-32
- UTF-32BE
- UTF-32LE

Encoding form plus byte serialization, and possible use of Byte Order Mark (BOM).

A code page is a coded character set, in which each character is assigned a unique code within the Unicode code space. Code pages usually cover only a small subset of the Unicode characters.

For more information about the Unicode standard, see [http://www.unicode.org](http://www.unicode.org).
Code pages and Unicode support in StreamServe

The StreamServer and the StreamServe tools support Unicode version 2.1, in the following encoding schemes:

- UTF-8 (with and without BOM)
- UTF-16
- UTF-16BE
- UTF-16LE

In Unicode version 2.1, the support is restricted to characters in the Basic Multilingual Plane (BMP) for all Unicode encoding schemes. UTF-16 in the BMP is referred to as UCS-2 in the StreamServe tools.

How the StreamServer handles character encoding

Internally, the StreamServer handles all data in the UCS-2 encoding form. This means that the input data must be converted to UCS-2 before the StreamServer starts processing the data. When the StreamServer has finished processing the data, the output is encoded using the appropriate code page.

In order to convert the input to UCS-2, without corrupting the input data, the StreamServer must know which code page is used to encode the input data. You must specify this when you configure your Project. You must also specify which code page to use to encode the output data.

Example 1 ISO 8859-15 encoded input and output.

In this example, input data is ISO 8859-15 encoded. The StreamServer converts the input data to UCS-2, processes the data, and uses ISO 8859-15 to encode the output data before sending it to the printer.

Preparing the workstation environment

Before you configure your Projects in the Design Center you may have to prepare your workstation environment with respect to the fonts and code pages to be used, and to the language version of your operating system.

Whenever possible you should use the appropriate language version of the operating system. For example, always try to configure a Greek Project on a Greek operating system.
You must make sure all required fonts are available, as well as complex script support (e.g. for Arabic or Hebrew) if needed. Consult the Microsoft® Windows® documentation or your System Administrator for information on how to do this.

**Specifying code pages for input and output data**

Code pages for the input can be specified as filters in filter chains that you add to the input connector. A code page can also be specified for the Event, either by using the scripting function `ConvCurrMsgToUC` in a retrieved script, or by using lookup tables or script aliases. Code pages for the output are specified in the output connector configuration (Platform).

Plain text data that contains Arabic or Hebrew text in logical order is treated the same way as data that contains unidirectional left-to-right text. Arabic/Hebrew text in visual order must be reordered to logical order before the StreamServer processes the text. Output from the StreamServer can also be reordered from logical to visual order if required (e.g. Arabic text in PDF output). See **Bidirectional text** on page 27.

**Export files from the Design Center**

All configuration files included in the export from the Design Center are UTF-8 encoded.

**Table files and function files**

All table files and function files must be UTF-8 encoded. See **Specifying code pages for table files** on page 23 and **Specifying code pages for function files** on page 24.
Log files
If you want the StreamServer log to display “non-Latin 1” characters you can use the startup arguments -logfilecp and -logcp. See Specifying code pages for the log on page 25.

ODBC scripting functions
The following ODBC scripting functions can be used to specify code pages when retrieving input from an ODBC data source:

- OdbcSetCodepage
- OdbcConnectW

See the Scripting reference for information.

Specifying a default code page for the Design Center
You can specify a default code page for the Design Center. This code page will be the default code page for all Projects you create. For each Project you can override the Design Center default code page, and specify a new default code page for the Project. When you create a code page filter for an input connector, or a code page for an output connector, the Project’s default code page is selected by default.

To specify a default code page for the Design Center
1. In the Design Center, select Tools > Design Center Settings. The Design Center Settings dialog box opens.
2. From the Default code page drop-down list, select the appropriate code page.

To specify a default code page for a Project
1. Open the Project in the Design Center.
2. In the Project browser, right-click the top node and select Settings. The Project Settings dialog box opens.
3. From the Default code page drop-down list, select the appropriate code page.
Known limitations

The StreamServe Unicode support has some limitations:

- Unicode encoded text in overlays, created in the StreamServe Overlay Editor, is not supported.
- The StreamServe MailOUT Process does not support Unicode. Instead, you must use an SMTP (MIME) output connector and the appropriate Process.
- Do not use characters outside the ASCII range in executable scripts or for variables.
- The Lotus Notes Fetch Wizard does not support Unicode, only ASCII.

Code pages supported by the StreamServer

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG5</td>
<td>Traditional Chinese, Taiwan</td>
</tr>
<tr>
<td>CNS 11643</td>
<td>Traditional Chinese, Taiwan</td>
</tr>
<tr>
<td>cp437_DOSLatinUS</td>
<td>Microsoft DOS US</td>
</tr>
<tr>
<td>cp737_DOSGreek</td>
<td>Microsoft DOS Greek</td>
</tr>
<tr>
<td>cp775_DOSBaltRim</td>
<td>Microsoft DOS Baltic Rim</td>
</tr>
<tr>
<td>cp850_DOSLatin1</td>
<td>Microsoft DOS Latin1 (Western Europe)</td>
</tr>
<tr>
<td>cp852_DOSLatin2</td>
<td>Microsoft DOS Latin2 (Eastern Europe)</td>
</tr>
<tr>
<td>cp855_DOSCyrillic</td>
<td>Microsoft DOS Cyrillic (Russia)</td>
</tr>
<tr>
<td>cp857_DOSTurkish</td>
<td>Microsoft DOS Turkish</td>
</tr>
<tr>
<td>cp860_DOSPortuguese</td>
<td>Microsoft DOS Portuguese</td>
</tr>
<tr>
<td>cp861_DOSIcelandic</td>
<td>Microsoft DOS Icelandic</td>
</tr>
<tr>
<td>cp862_DOSHebrew</td>
<td>Microsoft DOS Hebrew (Israel)</td>
</tr>
<tr>
<td>cp863_DOSCanadaF</td>
<td>Microsoft DOS CanadaF (Canada French)</td>
</tr>
<tr>
<td>cp864_DOSArabic</td>
<td>Microsoft DOS Arabic</td>
</tr>
<tr>
<td>cp865_DOSNordic</td>
<td>Microsoft DOS Nordic</td>
</tr>
<tr>
<td>cp866_DOSCyrillicRussian</td>
<td>Microsoft DOS Cyrillic (Russia)</td>
</tr>
<tr>
<td>cp874_DOSThai</td>
<td>Microsoft DOS Thai</td>
</tr>
<tr>
<td>cp932_ShiftJIS</td>
<td>Microsoft ShiftJIS (Japanese industrial standard)</td>
</tr>
</tbody>
</table>
### Code pages supported by the StreamServer

#### About code pages and Unicode support

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp936_GBK</td>
<td>Microsoft GBK (Simplified Chinese)</td>
</tr>
<tr>
<td>cp949_UnifiedHangul</td>
<td>Microsoft Unified Hangul (Korea)</td>
</tr>
<tr>
<td>cp950_Big5</td>
<td>Microsoft Big5 (Traditional Chinese, Taiwan)</td>
</tr>
<tr>
<td>cp950_Big5_HKCS-2001</td>
<td>Microsoft Big5 (Hong Kong Supplementary Character Set)</td>
</tr>
<tr>
<td>cp1250_WinLatin2</td>
<td>Microsoft Windows Latin2 (Eastern Europe)</td>
</tr>
<tr>
<td>cp1251_WinCyrillic</td>
<td>Microsoft Windows Cyrillic (Russia)</td>
</tr>
<tr>
<td>cp1252_WinLatin1</td>
<td>Microsoft Windows Latin1 (Western Europe &amp; USA)</td>
</tr>
<tr>
<td>cp1253_WinGreek</td>
<td>Microsoft Windows Greek</td>
</tr>
<tr>
<td>cp1254_WinTurkish</td>
<td>Microsoft Windows Turkish</td>
</tr>
<tr>
<td>cp1255_WinHebrew</td>
<td>Microsoft Windows Hebrew (Israel)</td>
</tr>
<tr>
<td>cp1256_WinArabic</td>
<td>Microsoft Windows Arabic</td>
</tr>
<tr>
<td>cp1257_WinBaltic</td>
<td>Microsoft Windows Baltic</td>
</tr>
<tr>
<td>cp1258_WinVietnamese</td>
<td>Microsoft Windows Vietnamese</td>
</tr>
<tr>
<td>GB2312-80</td>
<td>Simplified Chinese</td>
</tr>
<tr>
<td>IBM CP 37</td>
<td>Common Europe EBCDIC</td>
</tr>
<tr>
<td>IBM CP 256</td>
<td>IBM Netherlands EBCDIC</td>
</tr>
<tr>
<td>IBM CP 273</td>
<td>Austria/Germany EBCDIC</td>
</tr>
<tr>
<td>IBM CP 277</td>
<td>Denmark/Norway EBCDIC</td>
</tr>
<tr>
<td>IBM CP 278</td>
<td>Finland/Sweden EBCDIC</td>
</tr>
<tr>
<td>IBM CP 280</td>
<td>Italian EBCDIC</td>
</tr>
<tr>
<td>IBM CP 284</td>
<td>Spanish EBCDIC</td>
</tr>
<tr>
<td>IBM CP 285</td>
<td>UK EBCDIC</td>
</tr>
<tr>
<td>IBM CP 297</td>
<td>French EBCDIC</td>
</tr>
<tr>
<td>IBM CP 423</td>
<td>Greek EBCDIC</td>
</tr>
<tr>
<td>IBM CP 424</td>
<td>Hebrew EBCDIC</td>
</tr>
<tr>
<td>IBM CP 500</td>
<td>International EBCDIC</td>
</tr>
<tr>
<td>IBM CP 852</td>
<td>Latin 2 PC-DATA</td>
</tr>
<tr>
<td>IBM CP 870</td>
<td>Latin-2 EBCDIC</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>IBM CP 875</td>
<td>Greek EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1026</td>
<td>IBM Turkey Latin-5</td>
</tr>
<tr>
<td>IBM CP 1140</td>
<td>US-Canada-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1141</td>
<td>Germany-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1142</td>
<td>Denmark-Norway-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1143</td>
<td>Finland-Sweden-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1144</td>
<td>Italy-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1145</td>
<td>Spain-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1146</td>
<td>UK-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1147</td>
<td>France-Euro EBCDIC</td>
</tr>
<tr>
<td>IBM CP 1149</td>
<td>Icelandic-Euro EBCDIC</td>
</tr>
<tr>
<td>ISO 8859-1</td>
<td>Western Europe</td>
</tr>
<tr>
<td>ISO 8859-2</td>
<td>Eastern Europe</td>
</tr>
<tr>
<td>ISO 8859-3</td>
<td>Southern Europe, Malta</td>
</tr>
<tr>
<td>ISO 8859-4</td>
<td>Northern Europe, Baltic countries, Greenland</td>
</tr>
<tr>
<td>ISO 8859-5</td>
<td>Cyrillic (Russia)</td>
</tr>
<tr>
<td>ISO 8859-6</td>
<td>Arabic</td>
</tr>
<tr>
<td>ISO 8859-7</td>
<td>Greek</td>
</tr>
<tr>
<td>ISO 8859-8</td>
<td>Hebrew</td>
</tr>
<tr>
<td>ISO 8859-9</td>
<td>Turkish</td>
</tr>
<tr>
<td>ISO 8859-10</td>
<td>Nordic</td>
</tr>
<tr>
<td>ISO 8859-13</td>
<td>Baltic Rim</td>
</tr>
<tr>
<td>ISO 8859-14</td>
<td>Celtic languages</td>
</tr>
<tr>
<td>ISO 8859-15</td>
<td>ISO 8859-1 + Euro sign and extra characters for France/Finland</td>
</tr>
<tr>
<td>JIS 0212</td>
<td>Japanese Industrial standard</td>
</tr>
<tr>
<td>Roman-8</td>
<td>HP-Roman8</td>
</tr>
<tr>
<td>Shift-JIS</td>
<td>Japanese Industrial standard</td>
</tr>
<tr>
<td>Unified Hangeul KSC5601-87</td>
<td>Korean DBCS</td>
</tr>
</tbody>
</table>
### About code pages and Unicode support

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unified Hangeul KSC5601-92</td>
<td>Korean DBCS</td>
</tr>
<tr>
<td>Unified Hangeul KSX1001</td>
<td>Korean DBCS</td>
</tr>
<tr>
<td>Unicode (UCS-2)</td>
<td>Unicode encoding schemes UTF-16, UTF-16BE, UTF-16LE (restricted to the BMP)</td>
</tr>
<tr>
<td>UTF8</td>
<td>Unicode encoding scheme(s) UTF-8, with or without BOM (restricted to the BMP)</td>
</tr>
</tbody>
</table>
Specifying code pages for input data

You must specify which code page the source application uses to encode the input to the StreamServer. First, you identify the code page used for encoding the input (see Identifying the code page used to encode input data on page 14), then you select this code page when you configure the code page settings for the input in the Design Center. Where possible, use a Unicode encoding scheme for the input data.

**Code pages applied per input connector**

If the same code page is used for all input data received by an input connector, you can specify a code page for the input connector. See Specifying code pages per input connector on page 15.

**Code pages applied per input type**

If the input connector receives different types of input, and if the input types are encoded using different code pages, you can specify one code page per input type. See Specifying code pages per input type on page 17.

**Retrieved script or lookup/script alias on the Event**

You can use a retrieved script, or lookup/script aliases to dynamically select which code page to use for the input to an Event. See Dynamically selecting code pages for the input to an Event on page 19.
Identifying the code page used to encode input data

Information about the code page used to encode the input data is often available from the data source documentation or from the System Administrator.

You can also open a sample input data file in StreamServe UTF Edit to identify the code page used.

To use UTF Edit to identify the code page

1. In UTF Edit, select Edit > Set Font.
2. From the Fonts drop-down list, select a font that supports a wide range of code pages (e.g. Arial) and click OK.
3. Open the sample input data file.
4. Select Edit > Code Page and select a code page that displays all characters correctly in UTF Edit.

If you cannot find the accurate code page, repeat the procedure with another font. If you still cannot find the accurate code page, you must install a more suitable font.
Specifying code pages per input connector

If the same code page is used for all input data received via an input connector, you can specify the code page when you configure the input connector in the Platform.

To specify the code page

1. Create a filter chain, or use an existing filter chain.
2. Open the filter chain.
3. In the filter chain editor, right-click the flow bar and select Add Filter > Codepage Filter. A new code page filter is added to the flow bar.
4. From the Code page drop-down list, select the appropriate code page.
5. Save the filter chain and close the filter chain editor.
6. Activate the generic layer in the Platform view.
7. In the Platform view, double-click the input connector. The Input Connector Settings dialog box opens.
8. Click the Filter Chain icon and browse to and select the filter chain that contains the code page filter.

To specify the byte order for UCS-2 encoded data

For the UTF-16 encoding schemes, each character code unit is represented by two bytes. When you select UCS-2 as code page, you must also specify how the bytes are ordered for each code unit – most significant byte first or last.

From the Byte order drop-down list, select the appropriate option.

<table>
<thead>
<tr>
<th>Option</th>
<th>When to select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most significant byte first (Big Endian)</td>
<td>When the input is encoded using the encoding schemes UTF-16BE (big endian without byte order mark) or UTF-16 (big endian with or without byte order mark)</td>
</tr>
<tr>
<td>Most significant byte last (Little Endian)</td>
<td>When the input is encoded using the encoding schemes UTF-16LE (little endian without byte order mark) or UTF-16 (little endian with byte order mark).</td>
</tr>
</tbody>
</table>
Example 2  

*Code page filter connected to the input connector (Platform).*

In this example, input data received via the input connector is ISO 8859-15 encoded. A code page filter with the code page ISO 8859-15 is connected to the input connector.
Specifying code pages per input type

If an input connector receives different types of input (XML, page formatted, etc.), and if the input types are encoded using different code pages, you can specify one code page per input type. See the Design Center documentation for information about filter chains per input type.

To specify the code pages
1  Create a filter chain, or use an existing filter chain.
2  Open the filter chain.
3  In the filter chain editor, right-click the flow bar and select Add Filter > Codepage Filter. A new code page filter is added to the flow bar.
4  From the Code page drop-down list, select the appropriate code page.
5  Save the filter chain and close the filter chain editor.
6  Repeat steps 1 through 5 for each input type and code page.
7  In the Project browser, right-click the Project node and select Project Export Settings. The Project Export Settings dialog opens.
8  Select the InputAnalyzer tab.
9  In the Available connectors list, select the connector.
10 For each input type in the Input Analyzer settings list, click the Select a Filter Chain icon and browse to and select the filter chain that contains the code page filter for the input type.

Note: If you connect a code page filter to the connector in both the Platform and in the Project Export Settings dialog, the StreamServer will not start.

To specify the byte order for UCS-2 encoded data
For the UTF-16 encoding schemes, each character code unit is represented by two bytes. When you select UCS-2 as code page, you must also specify how the bytes are ordered for each code unit – most significant byte first or last.

From the Byte order drop-down list, select the appropriate option.

<table>
<thead>
<tr>
<th>Option</th>
<th>When to select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Most significant byte first (Big Endian)</td>
<td>When the input is encoded using the encoding schemes UTF-16BE (big endian without byte order mark) or UTF-16 (big endian with or without byte order mark)</td>
</tr>
<tr>
<td>Most significant byte last (Little Endian)</td>
<td>When the input is encoded using the encoding schemes UTF-16LE (little endian without byte order mark) or UTF-16 (little endian with byte order mark).</td>
</tr>
</tbody>
</table>
Specifying code pages per input type

Specifying code pages for input data

**Example 3**  Code page filters in the Runtime configuration.

In this example the input connector receives ISO 8859-15 encoded page formatted data, and ISO 8859-2 encoded XML formatted data. A code page filter with the code page ISO 8859-15 is connected to the PageIN branch, and a code page filter with the code page ISO 8859-2 is connected to the XMLIN branch.
Dynamically selecting code pages for the input to an Event

You can use lookup/script aliases or retrieved scripts to dynamically select the appropriate code page for the input to an Event.

**Prerequisites**
- The input data must be represented by single-byte characters.
- No code page filter is added to the input connector that receives the input data.

**To select code pages using a retrieved script**
Add a retrieved script that includes the `ConvCurrMsgToUC` scripting function to the Event. See the Scripting reference for information.

**To select code pages using lookup/script aliases**
1. In the Runtime configuration view, right-click the Event and select **Settings**. The Runtime Event Settings dialog box opens.
2. On the Code Page tab, select **Lookup** or **Variable** and specify the alias settings. See the Design Center documentation for information about alias selection methods.

**Lookup table syntax**

```
<key value>  <code page>
```

Comments:
- Use TAB to separate `<key value>` and `<code page>`.
- Use quotation marks if `<key value>` contains white spaces.
- `<code page>` must be a name listed in *Code pages supported by the StreamServer* on page 9.

**Example 4**

```
<table>
<thead>
<tr>
<th>Lookup table</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Western Europe&quot;  ISO 8859-1</td>
</tr>
<tr>
<td>&quot;Eastern Europe&quot;  ISO 8859-2</td>
</tr>
<tr>
<td>Turkish            ISO 8859-9</td>
</tr>
</tbody>
</table>
```
Example 5  Code page selection using a lookup table.

In this example the PageIN Event receives both ISO 8859-15 and ISO 8859-2 encoded input. The following lookup table is used to dynamically select the appropriate code page:

Western    ISO 8859-15
Eastern    ISO 8859-2
Specifying code pages for output data

The output can inherit the code page specified for the input, or you can specify a new code page for the output. The code page you specify must be supported by the output device (e.g. a printer).

To specify a code page for the output
1. Activate the generic layer in the Platform view.
2. In the Platform view, double-click the output connector. The Output Connector Settings dialog box opens.
3. Click the Code page icon and select Inherit code page or select the appropriate code page from the Code page drop-down list.

<table>
<thead>
<tr>
<th>Option</th>
<th>When to select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit code page</td>
<td>If you want to use the same code page for both input and output</td>
</tr>
<tr>
<td>Select a code page</td>
<td>If you want to select a different code page for the output.</td>
</tr>
<tr>
<td></td>
<td>This code page must cover at a minimum all the characters covered in the code page for the input.</td>
</tr>
</tbody>
</table>

Dynamically selecting code pages for the output
You cannot apply scripting or aliases to the output in order to dynamically select code pages. However, if you specify dynamic selection of code pages for the input, and select Inherit code page for the output, the code pages for the output will be dynamically selected as well.

To specify the byte order for UCS-2 encoded data
For the UTF-16 encoding schemes, each character code unit is represented by two bytes. When you select UCS-2 as code page, you must also specify how the bytes are ordered for each code unit – most significant byte first or last.

From the Byte order drop-down list, select Most significant byte last (Little Endian) or Most significant byte first (Big Endian). The byte order to select depends on the application that receives the output.

To add a byte order mark to UTF-8 and UCS-2 encoded data
Select Add byte order mark if you want to add a byte order mark at the beginning of a UCS-2 (UTF-16) or UTF-8 encoded output file.
Specifying code pages for output data

The application that receives the output can use this byte order mark to automatically determine the encoding (UTF-8 or UTF-16 encoding scheme) and the byte order used for the data in the UTF-16 encoding scheme.
Specifying code pages for support files and logs

This section describes how to specify code pages for support files and logs.

- Specifying code pages for table files on page 23
- Specifying code pages for function files on page 24
- Specifying code pages for the log on page 25

Specifying code pages for table files

All table files used in the Design Center must be UTF-8 encoded. In order to make the StreamServer interpret the table file as UTF-8 encoded, the file must begin with the following text string:

```//!codePage UTF8!```

Example 6 UTF-8 encoded table file.

```//!codePage UTF8!
ENG    Printer_1
SWE    Printer_2```

Table files created from within a resource set

If you create a new table file from within a resource set, the encoding is automatically set to UTF-8. When you open the table file in the resource editor, the code page information string `//!codePage UTF8!` is added by default.

Table files created using external text editors

You can create a table file using an external text editor, and then import the table file to a resource set. Before you import the file you must:

- enter `//!codePage UTF8!` as the first line of text
- save the file UTF-8 encoded.

Table files in upgraded Projects

If you are upgrading Projects prior to 4.1 you must modify the table files. Before you upgrade you must:

- enter `//!codePage UTF8!` as the first line of text
- save the file UTF-8 encoded.
Specifying code pages for function files

All function files used in the Design Center must be UTF-8 encoded. In order to make the StreamServer interpret the function file as UTF-8 encoded, the file must begin with the following text string:

```
CodePage UTF8
```

**Example 7** UTF-8 encoded function file.

```
CodePage UTF8
func timetotal()
... 
```

**Function files created from within a resource set**

If you create a new function file from within a resource set, the encoding is automatically set to UTF-8. When you open the function file in the resource editor, the code page information string `CodePage UTF8` is added by default.

**Function files created using external text editors**

You can create a function file using an external text editor, and then import the function file to a resource set. Before you import the file you must:

- enter `CodePage UTF8` as the first line of text
- save the file UTF-8 encoded.

**Function files in upgraded Projects**

If you are upgrading Projects prior to 4.1 you must modify the function files. Before you upgrade you must:

- enter `CodePage UTF8` as the first line of text
- save the file UTF-8 encoded.
Specifying code pages for the log

If the characters displayed in the StreamServer log conform to Latin 1, you do not have to specify a code page for the log. If you want the log to display other characters, you can use the startup arguments `-logfilecp` and `-logcp`. See the `Startup argument` reference.

**StreamServer run from the Control Center**

If you run the StreamServer from the Control Center, you can use the startup argument `-logfilecp` to specify a code page for the log file.

**StreamServer run from the command line**

If you run the StreamServer from command line, you can use the startup argument `-logcp` to specify a code page for the log.
Specifying code pages for the log
Specifying code pages for support files and logs
Bidirectional text

Bidirectional text consists of mainly right-to-left text with left-to-right nested segments, or vice versa. In languages involving bidirectional text (the StreamServer supports Arabic and Hebrew), the general text flow proceeds horizontally from right to left, but numbers are written from left to right. In addition, embedded addresses, acronyms, and quotations in a left-to-right language are also written from left to right.

Logical and visual order

Bidirectional text can be visually or logically ordered:

• Logical order – the characters are ordered in the same way that they have been keyed. In this case, it does not matter if the text is Arabic, Hebrew, or Latin – direction has no meaning.

• Visual order – the characters are ordered as they are displayed on a screen, printed page, or other medium.

Visually ordered input must be reordered to logical order before the StreamServer processes the text. See Reordering visually ordered input on page 28. Output from the StreamServer can be reordered to visual order if required. See Reordering output to visual order on page 29.
Reordering visually ordered input

Visually ordered input must be reordered to logical order before the StreamServer processes the text. You can reorder the input for the following Events:

- **PageIN** – reordering is applied per line. See Reordering page formatted input on page 28.

- **StreamIN** – reordering is applied per field. See Reordering field and record based input on page 28.

Input to all other Events must be logically ordered.

Reordering page formatted input

You can enable reordering of the input when you configure the code page settings for the input. See Specifying code pages for input data on page 13.

To reorder the input using a code page filter

From the **Input order** drop-down list, select **Visual order (Arabic and Hebrew only)**.

To reorder the input using aliases (Runtime Event settings)

From the **Input order** drop-down list, select **Visual order (Arabic and Hebrew only)**.

To reorder the input using a retrieved script

In **ConvCurrMsgToUC(codepage, visual_order)**, set **visual_order** to 1. For example:

```
ConvCurrMsgToUC("UTF8",1);
```

Reordering field and record based input

You can enable reordering of the input when you configure the input connector.

To reorder the input

Enter the **ReorderRTLField** keyword to the input connector. See Using custom commands and keywords in the Design Center documentation for information on how to add custom commands and keywords to a connector.

**Note:** You must not enable reordering using code page filters, aliases, or retrieved script for input received via this connector if you use this keyword.
Reordering output to visual order

The StreamServer processes text in logical order. If required, the output can be reordered to visual order. You can reorder the output from the following Processes:

- **PageOUT** – reordering is applied per paragraph. See Reordering page formatted output on page 29.
- **StreamOUT** – reordering is applied per field. See Reordering record based output on page 30.

Reordering page formatted output

Page formatted bidirectional text displayed on the screen or as printed output is normally displayed in visual order. Some output devices reorder the text automatically, which means the output from the StreamServer must be in logical order when it is sent to such a device. The table below shows how to order the output from the StreamServer when using different types of drivers.

<table>
<thead>
<tr>
<th>Driver</th>
<th>Output from the StreamServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCL drivers</td>
<td>Visual order.</td>
</tr>
<tr>
<td>Postscript drivers</td>
<td>Visual order.</td>
</tr>
<tr>
<td>GIF, BMP, etc.</td>
<td>Visual order.</td>
</tr>
<tr>
<td>RTF</td>
<td>Logical order.</td>
</tr>
<tr>
<td>PDF</td>
<td>Visual order.</td>
</tr>
<tr>
<td><strong>Windows driver</strong></td>
<td>Logical order.</td>
</tr>
</tbody>
</table>

Visually ordered page formatted Arabic text is in addition shaped – the glyphs for the letters are cursively joined, lam-alif ligatures are formed, and mirror characters (e.g. parentheses and brackets) are mirrored. Contextual shaping must therefore be performed on the text in order to create the correct sequences of glyphs. Shaping is automatically performed if you enable reordering of page formatted output. This functionality is restricted to be without vowel marks.

**Note:** Visually ordered Arabic PCL output must be UTF-8 encoded.

**To reorder the output per output connector**

1. Activate the generic layer in the Platform view.
2. In the Platform view, double-click the output connector. The Output Connector Settings dialog box opens.
3. Click the **Code page** icon and select **Reorder BiDi output in visual order**.
To reorder the output from mirrored PageOUT layouts
If you mirror PageOUT output, reordering is enabled automatically. See the PageOUT documentation for information on how to mirror page layouts.

Reordering record based output
Record based output should normally be in logical order. If required, the order of the output can be changed to visual order.

To reorder the output
1. Activate the generic layer in the Platform view.
2. In the Platform view, double-click the output connector. The Output Connector Settings dialog box opens.
3. Click the Code page icon and select Reorder BiDi output in visual order.