

Content Server / Spark

Version: 6.2

Internationalization Settings Guide

Document Revision Date: May 12, 2005

FatWire[®]
S O F T W A R E

FATWIRE CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. In no event shall FatWire be liable for any loss of profits, loss of business, loss of use of data, interruption of business, or for indirect, special, incidental, or consequential damages of any kind, even if FatWire has been advised of the possibility of such damages arising from this publication. FatWire may revise this publication from time to time without notice. Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Copyright © 2005 FatWire Corporation. All rights reserved.

This product may be covered under one or more of the following U.S. patents: 4477698, 4540855, 4720853, 4742538, 4742539, 4782510, 4797911, 4894857, 5070525, RE36416, 5309505, 5511112, 5581602, 5594791, 5675637, 5708780, 5715314, 5724424, 5812776, 5828731, 5909492, 5924090, 5963635, 6012071, 6049785, 6055522, 6118763, 6195649, 6199051, 6205437, 6212634, 6279112 and 6314089. Additional patents pending.

FatWire, Content Server, Content Server Bridge Enterprise, Content Server Bridge XML, Content Server COM Interfaces, Content Server Desktop, Content Server Direct, Content Server Direct Advantage, Content Server DocLink, Content Server Engage, Content Server InSite Editor, Content Server Satellite, and Transact are trademarks or registered trademarks of FatWire, Inc. in the United States and other countries.

iPlanet, Java, J2EE, Solaris, Sun, and other Sun products referenced herein are trademarks or registered trademarks of Sun Microsystems, Inc. *AIX, IBM, WebSphere*, and other IBM products referenced herein are trademarks or registered trademarks of IBM Corporation. *WebLogic* is a registered trademark of BEA Systems, Inc. *Microsoft, Windows* and other Microsoft products referenced herein are trademarks or registered trademarks of Microsoft Corporation. *UNIX* is a registered trademark of The Open Group. Any other trademarks and product names used herein may be the trademarks of their respective owners.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>) and software developed by Sun Microsystems, Inc. This product contains encryption technology from Phaos Technology Corporation.

You may not download or otherwise export or reexport this Program, its Documentation, or any underlying information or technology except in full compliance with all United States and other applicable laws and regulations, including without limitations the United States Export Administration Act, the Trading with the Enemy Act, the International Emergency Economic Powers Act and any regulations thereunder. Any transfer of technical data outside the United States by any means, including the Internet, is an export control requirement under U.S. law. In particular, but without limitation, none of the Program, its Documentation, or underlying information of technology may be downloaded or otherwise exported or reexported (i) into (or to a national or resident, wherever located, of) Cuba, Libya, North Korea, Iran, Iraq, Sudan, Syria, or any other country to which the U.S. prohibits exports of goods or technical data; or (ii) to anyone on the U.S. Treasury Department's Specially Designated Nationals List or the Table of Denial Orders issued by the Department of Commerce. By downloading or using the Program or its Documentation, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list or table. In addition, if the Program or Documentation is identified as Domestic Only or Not-for-Export (for example, on the box, media, in the installation process, during the download process, or in the Documentation), then except for export to Canada for use in Canada by Canadian citizens, the Program, Documentation, and any underlying information or technology may not be exported outside the United States or to any foreign entity or "foreign person" as defined by U.S. Government regulations, including without limitation, anyone who is not a citizen, national, or lawful permanent resident of the United States. By using this Program and Documentation, you are agreeing to the foregoing and you are representing and warranting that you are not a "foreign person" or under the control of a "foreign person."

Content Server and Spark Internationalization Settings Guide

Document Revision Date: May 12, 2005

Product Version: 6.2

FatWire Technical Support

www.fatwire.com/Support

FatWire Headquarters

FatWire Corporation
330 Old Country Road
Suite 207
Mineola, NY 11501
www.fatwire.com

Table of Contents

About This Guide	5
1 Database Settings	6
1.1 General Settings	6
1.2 Oracle	6
1.3 SQL Server	7
1.4 DB2 (<i>for CS Only</i>)	7
2 Application Server Settings	8
2.1 WebLogic 81	8
2.2 WebLogic 81 (<i>for CS Only</i>)	8
2.2.1 Control via System Encoding	8
2.2.2 Control via JVM file.encoding	9
2.3 Websphere 4.0.4	9
2.4 Sun ONE 7.0	9
2.5 Resin (<i>for CS Only</i>)	9
2.5.1 Control via System Encoding	9
2.5.2 Control via JVM file.encoding	10
3 File System (Windows/UNIX) Settings (optional)	11
4 CS-Specific Settings	12
4.1 cs.contentType Property	12
4.2 cs.contentType Variable (Set in SiteCatalog resargs1)	12
4.3 _charset_ hidden Variable in Forms	12
4.4 Preferred Encoding	12

4.5	Encoding Specified in XML or JSP Elements	13
4.6	Using SetVar Tag	13
4.7	Using HTTP META tag	14
5	xcelerate.charset	15
6	Areas to Watch Out For	16
6.1	Files Stored on the File System	16
6.1.1	XML and JSP Files	16
6.1.2	HTML Files	16
6.1.3	SystemSQL Queries	16
6.1.4	Page Cache Files Referenced from the SystemPageCache table	16
6.1.5	Attribute Editor	16
6.1.6	Article Bodies, Flex Assets, User-Defined Assets	16
6.2	eWebEditPro	17
6.3	XML Post	17
6.4	Catalog Mover	17
6.5	Upgrades and Applications (<i>for CS Only</i>)	17
6.5.1	Upgrades from 4.0 to 5	17
6.5.2	Content Server Explorer	17
6.5.3	CS-DocLink (Windows Explorer Integration)	18
6.5.4	CS COM	18
6.5.5	InSite Editor	18
6.5.6	Content Server's Tree	18
7	Major Encodings	19
8	Known Problems and Limitations	20
8.1	Sun JDK and Content Apps Tree (<i>for CS Only</i>)	20
8.2	Functional Restrictions (<i>for CS Only</i>)	20
9	Browser Configurations	21
9.1	For All Browsers	21
9.2	Netscape 4.7x	21
9.3	Internet Explorer	21
	Appendix A. Configuring SQL Server 2000 for Japanese	22

About This Guide

This guide provides instructions for configuring two FatWire products—Content Server and Spark—to support multi-language operations. For example, it shows you how to set the character set of the database and application server, set properties that determine character encoding, and configure browsers to support content in different languages.

Note that Spark is a portal application, whereas Content Server supports both a web and a portal environment. This guide applies to all the environments. When information is unique to a particular product, it is identified in section titles by one of the following labels: *for CS only*, or *for Spark only*.

Who Should Use This Guide

This reference is written for installation engineers, developers, and administrators of Content Server systems. Users of this guide must be familiar with databases, application servers, web servers, and browsers; with file system encodings and product-specific property files and tags. For CS-based operations, users of this guide must also be familiar with Content Server's interfaces and clients, such as CS-Desktop and CS-DocLink.

How This Guide Is Organized

This guide presents information that is common to both the Content Server and Spark products. If a section applies to only one of the products, it is tagged with one of the following labels: *for CS only* or *for Spark only*.

Related Publications

The FatWire library includes publications written for Content Server and Spark administrators, developers, installation engineers, and content providers. The publications are supplied as product manuals with your Content Server installation. They are also posted on the Web at the following url: <http://e-docs.fatwire.com/CS>. Check the site regularly for updates.

Other publications, such as case studies and white papers, provide information about Content Server's feature set and its business applications. To obtain these publications, contact sales@fatwire.com.

1. Database Settings

This section specifies settings for the databases that are supported by Content Server and Spark.

1.1 General Settings

The character set of the database should be preferably set to UTF-8. If it is set to some particular character set (for example, Shift JIS) then it will work only for that particular character set and its subsets.

SQL92 introduced the concept of the National Character and SQL99 expanded upon it. The details can be subtle, but essentially the database character set is a byte oriented encoding such as 8859-1, Shift_JIS, UTF-8, and the National Character set (for practical purposes) is a 16-bit encoding such as UTF-16, JIS208, JIS212. Characters in the database character set are stored in CHAR and related fields. Characters from the National character set are stored in NCHAR and related fields.

SQL-2000 does not support UTF-8 as a database character set. It has UTF-16 as the national character set. Oracle and DB2 have supported UTF-8 as a database character set for some time.

Note

Content Server has a property that causes all character data to be stored in NCHAR fields instead of CHAR fields.

1.2 Oracle

If you plan to use languages other than US English in your database, it is a good idea to include the latest `nls_charset.zip` with `classes12.zip` in your JDBC driver. (You can get the latest drivers `classes12.zip` and `nls_charset.zip` from www.oracle.com.)

Some example character sets are shown below:

Database server encoding ("CHARACTER SET")	For Environments
UTF8 or WE8ISO8859P1	Latin-1
UTF8 or JA16SJIS	Japanese (Shift_JIS)
ZHS16GBK	Simplified Chinese (GB2312)

1.3 SQL Server

Configuring SQL Server for Japanese is described in detail in Appendix A, “Configuring SQL Server 2000 for Japanese.”

SQLServer 2000 uses the fixed character set for storing data in its database; UCS-2. UCS-2 is a 2-byte fixed-width encoding of Unicode, and has the same effect as using UTF-8. Therefore SQLServer 2000 is ready for multilingual site by default.

For SQLServer2000 on Windows2000 AND UTF8 for page encoding, the following properties must be set through Property Editor during the Content Server or Spark installation:

```
cc.char=NCHAR
cc.varchar=NVARCHAR
cc.maxvarchar=4000
cc.bigtext=NTEXT
```

Note

During installation, the properties listed above must be set as shown above.

1.4 DB2 (*for CS Only*)

When configuring a database in DB2, you need to be aware of the following I18n settings:

- Country (Language)
- Territory
- Code Set

The first two properties are used for controlling the messages and date/time formats used by clients, and are not used by Content Server. Hence they will not be discussed further in this document.

The code set should be set to the character set that can hold all the different types of characters your database will store now and in future. For maximum flexibility, UTF-8 is recommended.

2. Application Server Settings

All the settings described below for the application server are based on the assumption that you want Content Server or Spark to be able to handle all the major languages of the world. Towards that end, the character set refers to UTF-8. If you want to handle only one particular type of language, then you can set the encoding to the particular type. For example, if you want the system to work for Japanese (and its subsets), you could set the `LANG` and `LC_ALL` variables to `ja_JP.PCK` and/or the JVM file encoding to `Shift_JIS`.

2.1 WebLogic 81

In your JVM startup option, add `-Dfile.encoding=UTF-8`.

2.2 WebLogic 81 (for CS Only)

2.2.1 Control via System Encoding

2.2.1.1 Solaris

Set the `LANG` and `LC_ALL` environment variables to a UTF-8 encoding. The language and country codes do not matter (they are for date/time display only). For example, any of the following is good for a UTF-8 encoding: `en_US.UTF-8`, `fr_FR.UTF-8`, `ja_JP.UTF-8`. Once your system encoding is set, you need not provide `-Dfile.encoding` in your startup scripts.

Note

How to change LANG and LC_ALL on UNIX

In this example, we assume that you want UTF-8 encoding, but the date, time and number formats to be in US English. If you want to show your date/time/number formats in French, for example, use `fr_FR.UTF-8`. The best way to do this is to issue a `locale -a` command. This will list all the available encoding on your system.

- t-shell:
 - `setenv LANG en_US.UTF-8`
 - `setenv LC_ALL en_US.UTF-8`
- k shell:
 - `LANG=en_US.UTF-8`
 - `export LANG`
 - `LC_ALL=en_US.UTF-8`
 - `export LC_ALL`

Make sure that the following language packs are installed on your system so that you can work with the encodings above:

To install the Solaris language packs you will need the Solaris OS CDs. Use the following command for Solaris CD 1:

```
# pkgadd -d /cdrom/sol_9_1203_sparc/s0/Solaris_9/Product SUNWarrf
SUNWeurf SUNWi2rf SUNWi4rf SUNWi5rf SUNWi7rf SUNWi8rf SUNWi9rf
SUNWi13rf SUNWi15rf SUNWtxfnt SUNWinttf "SUNW5xmft" SUNWcxmft
```

```
SUNWjxmft SUNWkxmft SUNWeu8df SUNWeu8os SUNWeu8ox SUNWeudba
SUNWeudda SUNWeudhr SUNWeudis SUNWeudiv SUNWeudlg SUNWeudmg
SUNWeuezt SUNWeuhed SUNWeuluf SUNWeulux SUNWeuodf SUNWeuxwe
SUNWuiu8 SUNWuiu8x SUNWulcf SUNWulcfx SUNWulocf SUNWuxlcf
SUNWuxlcf
```

Use the following command for Solaris CD 2:

```
# pkgadd -d /cdrom/sol_9_1203_sparc/s0/Solaris_9/Product SUNWeudbd
SUNWeudhs SUNWeusru SUNWuium NSCPeu8cm
```

More information on installing Solaris locale packs can be found at the following website:
http://developers.sun.com/dev/gadc/techtips/sol9_locale_pkgs/americas.html

2.2.1.2 Windows

On Windows systems, set the Language and Locale Settings in the Regional Options. For instructions, see “How to change Windows code page settings” on page 11.

2.2.2 Control via JVM file.encoding

If for some reason you cannot set the LANG and LC_ALL environment variables, then you will need to provide `file.encoding` in your WebLogic start scripts. In your JVM startup option, just add `-Dfile.encoding=UTF-8` and you should be all set as far as configuring your WebLogic is concerned.

2.3 Websphere 4.0.4

Specify the `file.encoding` parameter in the application server's JVM options. For example: `-Dfile.encoding=UTF-8`

2.4 Sun ONE 7.0

Specify the `file.encoding` parameter in the application server's JVM options. For example: `-Dfile.encoding=UTF-8`

2.5 Resin (*for CS Only*)

2.5.1 Control via System Encoding

2.5.1.1 Solaris

Set the LANG and LC_ALL environment variables to a UTF-8 encoding. The language and country codes do not matter (they are for date/time display only). For example, any of the following is good for a UTF-8 encoding: `en_US.UTF-8`, `fr_FR.UTF-8`, `ja_JP.UTF-8`. Once your system encoding is set, you need not provide `-Dfile.encoding` in your startup scripts.

2.5.1.2 Windows

On Windows systems, set the Language and Locale Settings in the Regional Options. For instructions, see “How to change Windows code page settings” on page 11.

2.5.2 Control via JVM `file.encoding`

If for some reason you cannot set the `LANG` and `LC_ALL` environment variables, then you will need to provide `file.encoding` in your resin start scripts. In your JVM startup option, just add `-Dfile.encoding=UTF-8` and you should be all set as far as configuring your resin is concerned.

3. File System (Windows/UNIX) Settings (optional)

If you have the correct file system encoding set (e.g., via Windows codepage or UNIX LANG and LC_ALL environment variables), you can avoid setting the `file.encoding` for the application server.

How to change Windows code page settings

In this example, we assume you are changing the code page to Japanese.

1. Go to **Control Panel > Regional Options > General tab**.
2. Set BOTH the "Language settings for the system" and "Locale" to Japanese, in that order.
3. Set Japanese as the default language. (You may be prompted for the original Win2K CD for some files.)
4. Go to **Control Panel > Regional Options > Input Locales tab**.
5. Add "Japanese" and make it the default.
6. Re-start the machine.
7. After re-starting, select **Control Panel > Regional Options** to check the regional options and verify that the changes have been implemented (often, you will find that English is still the locale). If the verification passes, continue with the next step. Otherwise, repeat this procedure from step 1.
8. As a final test, do the following:
 - a. Open the command prompt in Windows by selecting **Start menu > run > cmd**.
 - b. Type **chcp** at the command prompt. It should show 932.

4. CS-Specific Settings

Note

If none of the properties and variables described below are set, the `cs.contentType` property defaults to `text/html`. The character set of the output now defaults to default system encoding.

4.1 cs.contentType Property

`cs.contentType` is a system-wide (global) property that defines the outgoing character encoding. By default, the property is set to `text/html; charset=UTF-8`. If you need a specific encoding, change the value. For example, if you want the outgoing encoding to be Shift JIS, set this property to `text/html; charset=Shift_JIS`. Content Server Explorer depends on this setting to display data correctly.

4.2 cs.contentType Variable (Set in SiteCatalog resargs1)

The `cs.contentType` variable enables you to control the outgoing encoding on a page-by-page basis. The value should be set in the same way as the `cs.contentType` property, as shown in section 4.1, “cs.contentType Property .” This variable overrides the value defined in `cs.contentType` property.

Note that pages under WebServices are set to `cs.contentType=application/xml; charset=UTF-8`.

4.3 _charset_ hidden Variable in Forms

If you are using forms to input international data, make sure that you set this input type variable at the very beginning, after the form declaration. For example, if you expect the user to use non-ASCII data, make sure that the form looks similar to the following:

```
<form action='ContentServer' method='get'>
<input type='hidden' name='_charset_'/>
  <input type='hidden' name='pagename'
value='<%=ics.GetVar("pagename")%>' />
  <input type='text' name='name'
value='<%=ics.GetVar("name")%>' />
  <input type='submit' />
</form>
```

One point to note is that the `_charset_` hidden variable without a value works only on the IE browser. To work on Netscape, you must pass the outgoing encoding from the page served with the form. In CS-Direct they use the `xcelerate.charset` property here. There is no harm in specifying a value with IE, because IE will override the value, if for some reason it is not correct.

4.4 Preferred Encoding

When Content Server needs to consume HTTP requests with certain encodings (Cp943C for example) that are closely related to a more widely used encoding (Shift_JIS), it is not sufficient to merely rely on the `_charset_` hidden variable alone. Internet Explorer,

when it encounters a `_charset_` value set to Cp943C, changes it to Shift_JIS. This forces Content Server to read all data in Shift_JIS. To overcome this, a special names property syntax is used:

```
cs.contenttype.<character set>=<preferred encoding for that
character set>
```

For example, in relation to the scenario described above, this property would be specified as follows to indicate to Content Server to use Cp943C:

```
cs.contenttype.Shift_JIS=Cp943C
```

Note that this property structure is necessary only in special circumstances such as the one described above, where the behavior of Internet Explorer conflicts with and changes the value of `_charset_` value.

4.5 Encoding Specified in XML or JSP Elements

The encoding in the `<?xml` line of an XML element specifies the encoding of the `.xml` file on disk. The same is true of JSP. The encoding specified in the page directive specifies two things. The first is the encoding of the `.jsp` file on disk. The second is the outgoing encoding of the evaluated JSP element. This gets converted to the encoding of the enclosing JSP, or in the XML case, the outgoing encoding of the page (content-type). So `cs.contenttype` can be used to indicate that the outgoing page will have a specific encoding, like Shift-JIS, but a JSP can output UTF-8, and the UTF-8 will get converted to Shift-JIS into the output stream of the page response. An example on how to specify the encoding is:

- XML: `<?xml version="1.0" encoding="UTF-8"?>`
- JSP: `<%@ page contentType="text/html; charset=UTF-8" %>`

4.6 Using SetVar Tag

You can also control the outgoing page encoding by using the `SetVar` tag in JSP and XMLs. The catch in using the tag is that it must be set before anything is streamed out.

In JSPs we can do:

```
<cs:ftcs>
  <ics:setvar name="cs.contenttype" value="text/html;
charset=UTF-8" />
  ...
</cs:ftcs>
```

In XML, you have the following options:

```
<ftcs>
  <setvar name="cs.contenttype" value="text/html;
charset=utf-8"/>
  ...
</ftcs>
```

The second option is to use the `ics.streamheader` XML tag, but again this must be done before anything is streamed, and only in XML.

```
<ftcs>
  <ics.streamheader name="Content-Type" value="text/html;
charset=utf-8"/>
  ...
</ftcs>
```

4.7 Using HTTP META tag

If the encoding is specified by any of the methods above, then the META tag has no effect. Otherwise, the browser tries to display the data in the encoding specified by the META tag.

5. xcelerate.charset

`xcelerate.charset` is set to UTF-8 by default. We recommend not changing this setting. This property is used to specify encoding for the outgoing content type for all pages that are part of the CS-Direct application. This is different from the published site.

The `xcelerate.charset` property is used to set the charset portion of the `cs.contenttype` variable on every CS-Direct page, with the rest of the content type set to `text/html`. The `xcelerate.charset` property is also used to set the special hidden variable `_charset_` on all forms that are part of the CS-Direct application. Because Netscape browsers do not automatically set the value for this hidden variable, it is set by FatWire so that the posted-to page recognizes the encoding of the form post.

6. Areas to Watch Out For

The following section lists several features in Content Server and Spark with specific internationalization requirements.

6.1 Files Stored on the File System

Many indirect files are stored on the file system because of the url column references. The files are identified in this section.

6.1.1 XML and JSP Files

See section 4.5, “Encoding Specified in XML or JSP Elements.”

6.1.2 HTML Files

The HTML files are read using java's `file.encoding` parameter value. The data in the file also depends on the way it was stored initially.

6.1.3 SystemSQL Queries

SystemSQL uses a url column to point to a file on the file system that holds a SQL query. When this file is loaded, it is assumed that the encoding of the file is the Java default encoding (`System.getProperty("file.encoding")`). There are several possible ways to create the SystemSQL queries (CSE, text editor, CS-Direct application). It is probably best to always stick to ASCII7 when creating queries, since any data is typically merged using variable replacement at run time.

6.1.4 Page Cache Files Referenced from the SystemPageCache table

For page cache files, we manage the page cache file so that the encoding of the file is UTF-8. Since we write and read this file only through Content Server (or Spark), it can be managed this way.

6.1.5 Attribute Editor

You have two ways to specify the text for an attribute editor:

- Type in the text in the text area provided. The form post will determine the encoding.
- Use the **Browse** button to select a text file. The text file encoding should match the encoding specified in `xcelerate.charset` encoding.

6.1.6 Article Bodies, Flex Assets, User-Defined Assets

The article bodies are stored on the file system using java's `file.encoding` property value.

6.2 eWebEditPro

If you are running your site in UTF-8, you will need to modify eWebEditPro settings so that it displays the non-ASCII characters correctly. In `ewebeditpro3\config.xml` change the value for `charencode` from `charref` to `utf-8`.

For more detailed information, refer to http://www.ektron.com//support/ewebeditprokb.cfm?doc_id=1229

6.3 XML Post

When posting non ASCII files using XML Post, the java file encoding must match the encoding of the file. For example, if you are posting a Japanese file (stored as UTF-8) to a UTF-8 system, then one of the following should be set before the XML Post command is run:

- The system locale must be set to UTF-8.
- The option of `-Dfile.encoding=UTF-8` must be specified in the XML Post command.

Similarly, if the file is stored as Shift_JIS, then the corresponding system locale should be set or the `java file.encoding` option must be specified.

As of Content Server 5.0 we support the encoding in the `<?xml` line, as the first line in the posted XML file. This overrides everything else as far as the encoding in which the `.xml` file is read.

6.4 Catalog Mover

Edit the catalog mover `.bat` (or `.sh` on UNIX) file and modify the java command to include the `file.encoding` parameter with a value that reflects the correct encoding needed to display the characters stored in the catalogs. This step can be avoided if the default encoding of the file system matches that of the data stored in the catalogs.

6.5 Upgrades and Applications (*for CS Only*)

6.5.1 Upgrades from 4.0 to 5

Turn OFF magic switches before upgrading (for example, in the WebSphere application server, turn OFF `default.file.encoding` before the new 5.0 jars are executed). In Resin, remove all instances of character encoding.

6.5.2 Content Server Explorer

Content Server Explorer requires that `cs.contenttype` property in `futuretense.ini` be set to the correct value. If you are viewing simple ASCII characters, nothing needs to be done. However, for viewing complex characters, such as Japanese, you need to set `cs.contenttype` to one of the following:

- `text/html; charset=SJIS`
- `text/html; charset=UTF-8`

Also, users may need to load font support for Japanese and various other character sets in order to have them display correctly. To do this in Win2K for example, go to **Settings > Control Panel > Regional Options**. In the first tab, **General**, select the languages you

want to support from the list titled “Language settings for the system.” Click **Apply** then **OK**. At this point you will be required to put in the Windows installation CD.

6.5.3 CS-DocLink (Windows Explorer Integration)

6.5.4 CS COM

6.5.5 InSite Editor

6.5.6 Content Server’s Tree

The user's machine must be able to support the characters that are to be displayed in Content Server’s tree. For languages other than English, the user needs to make sure of the following:

- The appropriate fonts to display the characters are installed.
- For a Windows machine, the locale and language settings must support the characters coming through. For example, for the tree to be able to display Japanese chars, Windows must be configured as shown in “File System (Windows/UNIX) Settings (optional)” on page 11.
- For a UNIX machine, the locale (`LANG` and `LC_ALL` environment variables) must be appropriately set.
- The browser's encoding must be correctly set.
- There is an issue with Sun's JDK and the tree. See section 8, “Known Problems and Limitations.”

7. Major Encodings

- US ASCII7
- Latin-1 (ISO-8859-1)
- UTF-8
- Shift-JIS
- Korean
- Simplified Chinese

8. Known Problems and Limitations

8.1 Sun JDK and Content Apps Tree (*for CS Only*)

If your website user is using JRE 1.4.0, then the tree is unable to display non-Latin-1 characters. Both JRE 1.3.1 and 1.4.1 work.

8.2 Functional Restrictions (*for CS Only*)

Content Server has the following functional restrictions for international use:

- The Property Editor supports ASCII only.
- Decimal numbers must be entered in US format ("." decimal separator).

9. Browser Configurations

9.1 For All Browsers

- Make sure your system's locale is set correctly. See section 3, “File System (Windows/UNIX) Settings (optional)” for information on how to set your system’s locale.
- Read the I18n configurations instructions provided by the browser vendor.

9.2 Netscape 4.7x

Neither Content Server nor Spark supports internationalization in Netscape.

9.3 Internet Explorer

IE 6 and above by default have all the languages installed. If you are unable to see text in a particular language, you most probably need to enable it in your browser.

To view content in different languages:

1. Go to **Tools > Internet Options > General tab**.
2. Click on **Languages** on the bottom right of the page.
3. Click on **Add** to add more languages to the list of languages already displayed in the box.
4. Click **OK**.
5. Restart your browser. You should now see the content in your specified language if the web page so provides.

Appendix A

Configuring SQL Server 2000 for Japanese

Complete the following steps to configure SQL Server 2000 for Japanese.

Step I. Change the Windows Running Mode to Japanese

For instructions, see “How to change Windows code page settings” on page 11. Once the changes are implemented, go to “Step II. Install SQL Server 2000.”

Step II. Install SQL Server 2000

1. Un-install SQL Server if it is already installed.
2. Insert the Microsoft SQL Server™ 2000 compact disc in your CD-ROM drive. If the compact disc does not autorun, double-click `Autorun.exe` in the root directory of the compact disc.
3. Select **SQL Server 2000 Components**.
4. Select **Install Database Server** and setup will start the SQL Server Installation Wizard. At the **Welcome Screen**, click **Next**.
5. In the **Computer Name** dialog box, **Local Computer** is the default option and the local computer name appears in the edit box. Click **Next**.
6. In the **Installation Selection** dialog box, click **Create a new instance of SQL Server**, and then click **Next**.
7. Follow directions on the **User Information**, **Software License Agreement** and related screens.
8. In the “Installation Definition” dialog box, click **Server and Client Tools**, and then click **Next**.
9. In the “Instance Name” dialog box, if the **Default** checkbox is available, you can install either the default or a named instance. If the **Default** checkbox is not available, a default instance has already been installed, and you can install only a named instance.
10. To install the default instance, select the **Default** checkbox, and click **Next**.

11. In the “Setup Type” dialog box, click **Custom**, and then click **Next**.
12. In the “Select Components” list, select all the components and click **Next**.
13. In the “Service Accounts” dialog box, accept the default settings, enter your domain password, and then click **Next**.
14. In the “Authentication Mode” dialog box, select **Mixed** mode, provide and note the password and click **Next**.
15. In **Collation Settings**, set **Collation designator** as `Japanese_Unicode`, deselect all the sort orders, and click **Next**.
16. In “Network Libraries,” note the port number (1433, the default port on which SQL Server listens) and click **Next**.
17. When you finish specifying options, click **Next** in the “Start Copying Files” dialog box.
18. In the “Choose Licensing Mode” dialog box, make selections according to your license agreement, and click **Continue** to begin the installation.
19. In the “Setup Complete” dialog box, click **Yes, I want to restart my computer now**, and then click **Finish**.
20. Restart the computer if it does not restart automatically.

Step III. Configure SQL Server 2000

1. Expand **Console Root > Microsoft SQL Servers > SQL Server Group > Machine Name**.
2. Right-click on **Databases** and select **New Database**.
3. Provide a database name (e.g., `ftcs1`). Leave the “Collation Name” as **Server Default**.
4. In the **Data Files** tab, set the initial size to 50MB and click **OK**.
5. Expand the **Security** folder under the machine name and select **Logins**.
6. Right-click and select **New Login**.
7. Provide a name (e.g., `ftuser`), select the radio button for **SQL Server Authentication** and provide a password.
8. Select the database just created above and set the language as Japanese.
9. In the **Server Roles** tab, select **System Administrators** and **Security Administrators**.
10. In the **Database Access** tab, select the database you created above and select **db_owner** in **Permit in Database roles**. Click **OK**. SQL Server is now configured for Japanese.