

Using Content Server in International Environments
10g Release 3 (10.1.3.3.1)

May 2007

Using Content Server in International Environments, 10g Release 3 (10.1.3.3.1)

Copyright © 2007, Oracle. All rights reserved.

Contributing Authors: Ron van de Crommert

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

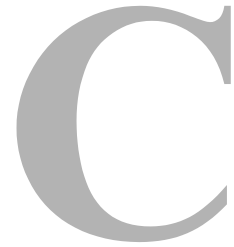
U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table of Contents



Chapter 1: Introduction

Overview	1-1
About This Guide	1-1
Content Server and International Environments	1-2

Chapter 2: Language Types

Overview	2-1
Single-Byte Languages	2-1
Double-Byte Languages	2-2
Bidirectional Languages	2-3

Chapter 3: Operating System Encoding

Overview	3-1
UNIX	3-1
Microsoft Windows	3-3

Chapter 4: Content Server File Encoding

Overview	4-1
About the Content Server File Encoding	4-1
File Encoding Setting	4-2
Changing the File Encoding	4-2
File Encoding Considerations	4-5

Chapter 5: Database Data Encoding

Overview	5-1
Text Field Encoding	5-2
Native Encoding	5-2
Unicode Encoding	5-2
Database Considerations	5-3

Chapter 6: Content Server Locales

Overview	6-1
System Locales	6-1
Setting the System Locale	6-2
Changing the System Locale Setting	6-2
Available System Locales	6-3
Editing an Existing System Locale	6-4
Adding a New System Locale	6-4
Locale Properties	6-6
System Locale Considerations	6-7
User Locales	6-8
Enabling User Locales	6-8
Available User Locales	6-9
User Locale Considerations	6-10

Chapter 7: Search Solution Settings

Overview	7-1
Database	7-2
SQL Server	7-2
Oracle	7-3
DB2	7-3
Sybase	7-4
PostgreSQL	7-4
Verity	7-4
Verity Locales	7-5
Supported Verity Locales	7-5
Changing the Verity Locale	7-7
Checking the Verity Version	7-8
Verity Considerations	7-8
FAST InStream	7-11

Chapter 8: Web Browser Settings

Overview	8-1
Adding Language Support	8-1
Browser Considerations	8-2

Chapter 9: Encoding Compatibility

Encoding Compatibility	9-1
------------------------------	-----

Chapter 10: International Scenarios

Overview	10-1
English-Only Environments	10-1
European Language Environments	10-2
Western European Languages	10-2
Other European Languages	10-3
Asian and Other Non-European Language Environments	10-4
Mixed Language Environments	10-10

Appendix A: Date and Time Format

Overview	A-1
Date and Time Format	A-1
Time Zone Format	A-5

Appendix B: Third Party Licenses

Overview	B-1
Apache Software License	B-1
W3C® Software Notice and License	B-2
Zlib License	B-3
General BSD License	B-4
General MIT License	B-5
Unicode License	B-5
Miscellaneous Attributions	B-7

Index

INTRODUCTION

OVERVIEW

This section covers the following topics:

- ❖ [About This Guide](#) (page 1-1)
- ❖ [Content Server and International Environments](#) (page 1-2)

ABOUT THIS GUIDE

This guide provides information about how to deploy Content Server in international, multilingual environments. It contains important considerations for installing the Content Server software on non-English operating systems and also explains how Content Server should be set up to handle multilingual content.






Important: The information in this document applies to 10gR3 releases of Content Server. Some of the information may not apply to earlier releases.

Symbols

The following symbols are used throughout this document:

Symbols	Description
	This is a note . It is used to bring special attention to information.

Symbols	Description
	This is a technical tip . It is used to identify information that can be used to make your tasks easier.
	This is an important notice . It is used to identify a required step or required information.
	This is a caution . It is used to identify information that might cause loss of data or serious system problems.

CONTENT SERVER AND INTERNATIONAL ENVIRONMENTS

Today's business world is becoming more and more international. Many organizations have international offices, and their operations span the globe. This means that their content management systems need to be deployed in increasingly international environments, with multilingual business content and users from different countries.

The Content Server software fully supports international environments:

- ❖ It can be installed on non-English operating systems.
- ❖ It supports non-English locales.
- ❖ It can be set up to offer localized end-user environments.
- ❖ It can handle and store content in a large number of languages.
- ❖ It can provide metadata and full-text searching of content in a large number of languages.

For everything to work correctly in international environments, the following factors are important:

- [single-byte, double-byte, and bidirectional languages \(see chapter 2\)](#)
- [the language encoding of the operating system \(see chapter 3\)](#)
- [the file encoding of the content server \(see chapter 4\)](#)
- [the data encoding of the content server database \(see chapter 5\)](#)
- [the system locale of the content server \(see chapter 6\)](#)
- [the user locales of the content server \(see chapter 6\)](#)

- ❑ the language settings of the search solution (see chapter 7)
- ❑ the language support of the web browser used to access the content server (see chapter 8)

It is crucial that the file encodings used in your overall content management system are all **compatible** (see chapter 9).

LANGUAGE TYPES

OVERVIEW

Content Server supports various language types:

- ❖ [Single-Byte Languages](#) (page 2-1)
- ❖ [Double-Byte Languages](#) (page 2-2)
- ❖ [Bidirectional Languages](#) (page 2-3)

SINGLE-BYTE LANGUAGES

Single-byte languages use SBCS (single-character character set) codepages. Content Server provides storage, indexing, and searching capabilities for the following single-byte languages:

Language	Single-Byte Encoding
Czech	Central Europe (i.e. Cp1250, ISO-8859-2)
Danish (Dansk)	Latin I (i.e. Cp1252, ISO-8859-1)
Dutch (Nederlands)	Latin I (i.e. Cp1252, ISO-8859-1)
English-UK	Latin I (i.e. Cp1252, ISO-8859-1)
English-US	Latin I (i.e. Cp1252, ISO-8859-1)

Language	Single-Byte Encoding
Finnish (Suomi)	Latin I (i.e. Cp1252, ISO-8859-1)
French (Français)	Latin I (i.e. Cp1252, ISO-8859-1)
German (Deutsch)	Latin I (i.e. Cp1252, ISO-8859-1)
Greek	Greek (i.e. Cp1253, ISO-8859-7)
Italian (Italiano)	Latin I (i.e. Cp1252, ISO-8859-1)
Hungarian (Magyar)	Central Europe (i.e. Cp1250, ISO-8859-2)
Norwegian (Bokmål)	Latin I (i.e. Cp1252, ISO-8859-1)
Norwegian (Nynorsk)	Latin I (i.e. Cp1252, ISO-8859-1)
Polish (Polski)	Central Europe (i.e. Cp1250, ISO-8859-2)
Portuguese (Português)	Latin I (i.e. Cp1252, ISO-8859-1)
Russian	Cyrillic (i.e. Cp1251, ISO-8859-5)
Spanish (Español)	Latin I (i.e. Cp1252, ISO-8859-1)
Swedish (Svenska)	Latin I (i.e. Cp1252, ISO-8859-1)
Turkish (Türkçe)	Turkish (i.e. Cp1254, ISO-8859-9)

DOUBLE-BYTE LANGUAGES

Double-byte languages use DBCS (double-character character set) codepages. Content Server provides storage, indexing, and searching capabilities for the following double-byte languages:

Language	Double-Byte Encoding
Japanese	Cp932, Shift-JIS, SJIS
Korean	Cp949, KSC5601, EUC-KR

Language	Double-Byte Encoding
Simplified Chinese	Cp936, GB2312
Traditional Chinese	Cp950, Big5

BIDIRECTIONAL LANGUAGES

Content Server provides storage, indexing, and searching capabilities for the following bidirectional languages:

Language	Single-Byte Encoding
Arabic	Arabic (i.e. Cp1256, ISO-8859-6)
Hebrew	Hebrew (i.e. Cp1255, ISO-8859-2)



Note: Both Arabic and Hebrew are single-byte languages.



Note: If you want to use any of the above bidirectional languages, it is recommended that you contact support.

OPERATING SYSTEM ENCODING

OVERVIEW

The language encoding of the operating system defines the character sets that the operating system can work with. The encoding characteristics depend on the operating system:

- ❖ [UNIX](#) (page 3-1)
- ❖ [Microsoft Windows](#) (page 3-3)

UNIX

The UNIX operating system does not have a specific language encoding. Its language support depends on the language packs that are installed for the operating system. You can see the available locales by using the “`locale -a`” command.

If Content Server is installed on a UNIX machine, an administrator needs to specify what system language should be used upon logon. You should make sure that the operating system is configured to use the same character encoding as the content server. If, for example, you want the content server to use the Korean locale, you need to make sure that you log on to UNIX using Korean as the operating system language. If you do not, there will be a file encoding mismatch, which will prevent the system from working correctly.

You need to make sure to set `LC_ALL` and `LANG` in the UNIX user login script. For example, if you are using *c shell* and you want to set the Korean locale, you need to set `LC_ALL` and `LANG` in the `.cshrc` file:

```
LC_ALL=ko_KR.EUC
export LC_ALL
LANG=ko_KR.EUC
export LANG
```



Important: You only need to be logged on with the correct language when starting the content server services and web server. After that, you may log off and log on again using a different language.

Determining Your Current Locale on Solaris

If you are using Sun Solaris, you can determine what locale the operating system is currently using by typing **locale** at the command prompt. This will display the current locale information, which will look something like this:

```
> locale
LANG=ja_JP.PCK
LC_CTYPE=en_US.ISO8859-1
LC_NUMERIC=en_US.ISO8859-1
LC_TIME=en_US.ISO8859-1
LC_COLLATE=en_US.ISO8859-1
LC_MONETARY=en_US.ISO8859-1
LC_MESSAGES="en_US.ISO8859-1"
LC_ALL=en_US.ISO8859-1
```

Setting Your Current Locale on Solaris

To set the locale, type the following at the command prompt:

```
>setenv LC_ALL [locale]
```

(where [*locale*] is the locale to be used).

Here are some commonly used locales:

Locale	Language	Encoding
en_US.ISO8859-1	English	ISO 8859-1
en_US.UTF-8	English	UTF-8
ja_JP.PCK	Japanese	SJIS
ja_JP.UTF-8	Japanese	UTF-8
ko_UTF-8	Korean	UTF-8 (Korean standard codeset on Solaris)

A full listing of encodings can be found in the “International Language Environment Guide” provided by Sun at <http://docs.sun.com>. Search for the title as there is one for each Solaris version.

MICROSOFT WINDOWS

Each Microsoft Windows installation has its own language encoding. This defines what character sets (code pages) Windows can work with (for example, display on screen, use in applications, handle in file names, etc.).

The initial language encoding is determined by the language version of Microsoft Windows that is installed. For example, if you installed the English version, the language encoding will be set to support all Western European languages. If you installed the Japanese Windows version, the default language encoding will be Japanese to enable the use and display of Japanese characters.

Adding Language Support

If you want Microsoft Windows to support multiple languages, you can install additional language packs. Each language pack provides support files for a particular language group, which enable Windows to display and handle the character sets specific to that group.

In **Windows 2000 Server**, you need to open Regional Options in Control Panel to add languages. On the General tab, under Language settings for the system, you can select the check box next to the language group(s) you want to install.

In **Windows Server 2003**, you can specify what language groups should be supported upon installation. You can also change the settings afterwards by opening Regional and Language Options in Control Panel. On the General tab, under Supplemental language support, you can specify whether you want to support right-to-left languages (such as Arabic and Hebrew) and/or East Asian languages (such as Japanese and Korean).



Note: To complete the installation, you must insert the Windows distribution media or have access to the system files on the network. Once the languages are installed, you need to restart the computer.

CONTENT SERVER FILE ENCODING

OVERVIEW

This section covers the following topics:

- ❖ [About the Content Server File Encoding](#) (page 4-1)
- ❖ [File Encoding Setting](#) (page 4-2)
- ❖ [Changing the File Encoding](#) (page 4-2)
- ❖ [File Encoding Considerations](#) (page 4-5)

ABOUT THE CONTENT SERVER FILE ENCODING

The file encoding of the content server specifies in what encoding the content server sends data to the database (and the external search engine, if required). It also defines what character sets can be used in the user interface, the metadata field names and values, the names of the files managed by Content Server, and the URLs used to interface with the web server.



Important: It is crucial that the file encodings of the content server, database, and search solution are all compatible. There is a lot of data traffic between these three system components, and incompatible file encodings will result in communication errors, display problems, etc. See [Chapter 9 \(Encoding Compatibility\)](#) for further details.

FILE ENCODING SETTING

During the installation of the Content Server software, you are prompted to specify what file encoding should be used for the content server instance: **UTF-8** or the **native** file encoding of the operating system.

Native file encoding supports English and Western European languages (i.e., German, French, Spanish, Italian, Portuguese, Dutch, Danish, Swedish, Norwegian, and Finnish). UTF-8 file encoding is generally recommended if the content server needs to handle content in languages of different file encodings (for example, Western European and Asian languages).



Caution: It is strongly recommended that you carefully consider which file encoding you need. If you anticipate content with different file encodings to be handled by the content server (for example, a combination of European and Asian languages), it is wise to choose UTF-8 as the Content Server file encoding.

If you initially chose to set up the content server with native encoding, this can be changed to UTF-8 later. However, this may be very cumbersome—especially if the database already contains data—and typically requires assistance from Oracle’s Consulting Services organization.

CHANGING THE FILE ENCODING

If you need to change the file encoding of the content server, it is strongly recommended that you do this without any data in the database. If the database already contains data, there are two scenarios:

- ❖ If your **database** is set up **with Unicode or UTF-8 support** and you change the file encoding of the content server, what you need to do depends on the search solution that is used with Content Server:
 - **Database searching**—Make sure that the database is capable of handling the additional character set(s). Other than that, you do not need to do anything.
 - **Verity or FAST**—Make sure that the search engine is capable of handling the additional character set(s). If required, modify the Verity locale. If you changed the Verity local, you need to rebuild the search index to ensure the data in the database can be accessed and retrieved correctly after the encoding change.



Caution: Please note that rebuilding the search index may be a very time-consuming process, depending on the number of content items managed by your Content Server instance. It is therefore recommended that you perform the index rebuild during off-peak hours of content server use (typically at night or on the weekend).

- ❖ If your [database](#) is set up *without Unicode or UTF-8 support* and you change the file encoding of the content server, you need to contact Oracle's Consulting Services organization to ensure the data in the database can be accessed and retrieved correctly after the encoding change.

Changing the File Encoding

To change the file encoding of the content server, complete the following steps:

1. Go to the file `[Install_Dir]/bin/intradoc.cfg` (where `[Install_Dir]` is the installation directory of the content server instance), and open it in a text editor.
2. Add the following entry (or modify it if it already exists):
`FileEncoding= [value]`
 in accordance with the table below, for example:
`FileEncoding=SJIS`
3. Save the modified configuration file and exit the text editor. (If a warning is displayed about saving the file to a text-only format, go ahead and confirm the action.)
4. Restart the content server instance.

Languages	Parameter
English (1)	No FileEncoding parameter or FileEncoding=ISO-8859-1 (no changes to default required)
English (1) + Western European languages (2)	No FileEncoding parameter or FileEncoding=ISO-8859-1 (no changes to default required)
Greek + English (1)	FileEncoding=ISO-8859-7
Russian + English (1)	FileEncoding=ISO-8859-5
Turkish + English (1)	FileEncoding=ISO-8859-9

Languages	Parameter
Czech + English (1)	FileEncoding=ISO-8859-6
Polish/Hungarian + English (1)	FileEncoding=ISO-8859-2
Arabic + English (1)	FileEncoding=Cp1256
Hebrew + English (1)	FileEncoding=Cp1255
Japanese + English (1)	FileEncoding=SJIS
Korean + English (1)	FileEncoding=KSC5601
Simplified Chinese + English (1)	FileEncoding=GB2312
Traditional Chinese + English (1)	FileEncoding=BIG5
Any mix of the above languages and language groups	FileEncoding=UTF8

Notes:

1. ‘English’ in the table essentially refers to all ASCII characters—i.e., a-z, A-Z, 0-9, and common punctuation marks (comma, colon, question mark, etc.), but not “international” letters such as é, Å, ö, ñ, ß, etc.
2. ‘Western European languages’ in the table refers to the following languages (in addition to English): German, French, Spanish, Italian, Portuguese, Dutch, Danish, Swedish, Norwegian, and Finnish (all sharing the ISO-8859-1 encoding).



Note: As shown in the table above, the ‘UTF8’ file encoding provides the most comprehensive character support. If you anticipate that the content server will need to handle multilingual content from various language groups, it is recommended that you use this Content Server file encoding.



Note: The file encoding setting for the content server does not affect any refinery connections that may be active. These refinery connections are providers to the content server, and their settings are separate and may be different from the content server’s settings.

FILE ENCODING CONSIDERATIONS

Please note the following considerations with regard to the content server file encoding:

- ❑ When using the Batch Loader to import content items into the content server, make sure that the file encoding of the batchload file matches the file encoding of the content server. Otherwise, extended characters (such as accented letters) are displayed and stored in the database incorrectly (typically as question marks or squares).
- ❑ If you change the file encoding of the content server after entering new users into the system, then any user name that has a non 7-bit character will have their password invalidated. A system administrator will have to re-enter the passwords. This is because the SHA1 hashing algorithm used with Content Server is based on a byte sequence produced using the current encoding of the content server.
- ❑ If you use the HTTP Basic protocol to log in and the native encoding of the client that launched the browser does not match that of the content server, then any non 7-bit ASCII characters cannot be used in user names. This is because the HTTP protocol does not have a mechanism for specifying the encoding used in user names and passwords (or in any other HTTP request header for that matter). If you use NTLM for authenticating Internet Explorer to the IIS web server, then the actual authentication is done in wide characters, but when you ask for the REMOTE_USER variable when inside IIS, the encoding will be in the default encoding of the computer that contains IIS. Again, this is a limitation in the original design of the web communication protocols.

One way to get around this is to use the CookieLoginPlugin plugin and do logins using cookies. Then there should be no encoding issues with performing logins.

- ❑ See the *Content Server Troubleshooting Guide* for more internationalization issues.

DATABASE DATA ENCODING

OVERVIEW

Content Server uses a database to store a variety of information, including:

- ❖ the metadata of all content item revisions (once or twice, depending on the search solution used),
- ❖ user profiles (with the logon and security information for each individual user),
- ❖ workflow definitions,
- ❖ subscriptions, and
- ❖ historical data.

In addition, the full text of the latest content item revisions may be stored in the database, depending on the search solution used.



Note: No actual content item files are stored in the database, just their file sizes, their metadata and, possibly, their extracted full text (depending on the search solution used). The content files, both native and web-viewable, are stored on the file system.

In the context of using the content server in international environments, the [text field encoding](#) of the database is important. Please note that there are some [important considerations](#) to take into account (see page 5-3).

TEXT FIELD ENCODING

The text fields in the content server database can be encoded in either of two ways:

- ❖ [Native Encoding](#)
- ❖ [Unicode Encoding](#)

You select which of these data encodings should be used for the content server database during the installation of the Content Server software.

Native Encoding

This is the encoding that was selected when the database software was installed (for example, English, German, EUC-KOR, and UTF-8). Using the native database encoding may restrict the character sets that the database can store. This, in turn, may limit the text that you can search for in Content Server.

If you choose native database encoding, the ‘varchar’ data type is used for the database columns created for the content server.

Unicode Encoding

This encoding always uses two bytes for each character rather than one. As a result, it can represent the writing systems of most of the world’s languages, including Asian languages such as Japanese and Korean. This means that this encoding provides the most comprehensive searching options. However, since two bytes are used to represent each character, using Unicode will double the database file size. It is therefore important that you size your database properly and that you allow for sufficient growth.

If you choose Unicode encoding, the ‘nvarchar’ data type is used for the database columns created for the content server.

DATABASE CONSIDERATIONS

Please note the following important considerations with regard to the database encoding:

- ❑ It is recommended that you carefully consider which data encoding you need for your content server instance. If you anticipate multilingual content of different encodings to be handled by the content server (for example, combinations of European and Asian languages), it is wise to choose Unicode as the data encoding for your content server database. If you initially chose to set up the content server database with native encoding, it can be converted to use Unicode encoding, but this is very cumbersome and typically requires assistance from Oracle's Consulting Services organization.
- ❑ Choosing Unicode encoding during the Content Server installation does not affect the language encoding, or charset, of the database. It only affects the data types of the database columns created for the content server ('nvarchar' vs. 'varchar').
- ❑ The data encoding for the Content Server database does not apply to the entire enterprise database, but only to Oracle's reserved tablespace within that database.
- ❑ Oracle may have trouble storing special characters such as em-dashes (—) and smart quotes if the database character set does not contain them. If you are using a Unicode Oracle solution, you should set the Oracle database encoding (charset) to Unicode (UTF-8).
- ❑ See the *Content Server Troubleshooting Guide* for more internationalization issues.

CONTENT SERVER LOCALES

OVERVIEW

This section covers the following topics:

- ❖ [System Locales](#) (page 6-1)
- ❖ [User Locales](#) (page 6-8)

SYSTEM LOCALES

Each content server instance has a particular system locale, which defines how that content server handles a number of language-specific issues such as the [date/time format](#) (see appendix A), default time zone, sort order, and default interface language.

This section covers the following topics:

- ❖ [Setting the System Locale](#) (page 6-2)
- ❖ [Changing the System Locale Setting](#) (page 6-2)
- ❖ [Available System Locales](#) (page 6-3)
- ❖ [Editing an Existing System Locale](#) (page 6-4)
- ❖ [Adding a New System Locale](#) (page 6-4)
- ❖ [Locale Properties](#) (page 6-6)
- ❖ [System Locale Considerations](#) (page 6-7)

Setting the System Locale

The system locale is initially set during the installation of the Content Server software. You are prompted to select the system locale to be used from a list. If you selected English-US (which is the default), the date format is set to *m/d/yy* (8/26/06).



Important: Each content server instance can only have *one* system locale. You can change the system locale of an instance, but you cannot add a system locale to give an instance multiple system locales.

Changing the System Locale Setting

The default system locale for Content Server is English-US. If you want, you can to change it using System Properties, which is one of the stand-alone administration utilities.



Note: Before changing the system locale, make sure that you close any applets that are running. This prevents UI elements from the original locale from being displayed after the system locale has been changed. In addition to restarting any applets, it is also necessary to restart the web browser.

To change the system locale, complete the following steps:

1. **UNIX:** Start the System Properties utility by running the *System_Properties* script, which is located in the *bin* subdirectory of the Content Server installation directory.

Windows: From the Windows Start menu, select Programs—Content Server Content Server—[Instance Name]—Utilities—System Properties.

The System Properties applet is started.
2. Open the Server tab.
3. From the System Locale dropdown list, choose the system locale you want to use for the current content server instance (see also [Available System Locales](#) on page 6-3).
4. Click **OK** to apply the setting and exit System Properties.
5. Stop and restart the content server:
 - a. Log into Content Server as an administrator.
 - b. Go to the Administration Applets page.
 - c. Choose Admin Server. (You may need to scroll down.)
 - d. Click the restart icon (curved arrow) next to the appropriate content server button.
6. Stop and restart the Admin Server:

UNIX:

- a) Go to the `[Install_Dir]/admin/etc` directory.
- b) Execute the `idcadmin_restart` script.

Windows:

- a) Go to Start—Settings—Control Panel—Administrative Tools—Services.
- b) Select the IDC Content Admin Service.
- c) Right-click and choose Restart from the popup menu.



Note: If you do not restart the Content Server and Admin Server, the changes will not take effect.

7. Depending on your database and search solution setup, you may need to perform an additional task:
 - ❖ If your [database](#) is set up to use Unicode or UTF-8 and your current [Verity locale](#) supports the character set of the new content server system locale, then you do not need to do anything. (This will generally be the case if you are using the ‘uni’ Verity locale.)
 - ❖ If your [database](#) is set up to use Unicode or UTF-8 and your current [Verity locale](#) does *not* support the character set of the new content server system locale, then you need to [change the Verity locale](#) to a locale that does [support the character set](#) (see chapter 7). After that, you need to rebuild the search index to ensure the data in the database can be accessed and retrieved correctly after the locale changes.
 - ❖ If your [database](#) is *not* set up to use Unicode or UTF-8 support, you need to contact Oracle’s Consulting Services organization to ensure the data in the database can be accessed and retrieved correctly after the system locale change.



Caution: Please note that rebuilding the search index may be a very time-consuming process, depending on the number of content items managed by your Content Server instance. It is therefore recommended that you perform the index rebuild during off-peak hours of content server use (typically at night or on the weekend).

Available System Locales

The System Locale dropdown list on the Server tab of System Properties contains all locales that can be selected. By default, only English-US and English-UK are available. If you want to set a different language as the system locale, you need to install a **language pack** that includes all [requisite files](#) for that language. If any language packs are available for your Content Server release, you can download them from Oracle’s support website. Installation instructions are provided with the language pack.

After you install a language pack for your Content Server version, all requisite directories and files are set up automatically and you can [change the system locale](#) (see page 6-2).



Important: Language packs will be made available for all major Content Server releases. There may be some lag time between the release of a new Content Server version and the availability of its associated language packs.

Editing an Existing System Locale

You can edit a number of properties of the existing system locales (for example, the time zone). To edit an existing system locale, complete the following steps:

1. **UNIX:** Start the System Properties utility by running the *System_Properties* script, which is located in the *bin* subdirectory of the Content Server installation directory.

Windows: From the Windows Start menu, select Programs—Content Server Content Server—[Instance Name]—Utilities—System Properties.

The System Properties applet is started.

2. Open the Localization tab.
3. In the list of locales, select the locale you wish to edit, and click **Edit**.

The Configure Locale screen is displayed.

4. Make the required changes (see [Locale Properties](#) on page 6-6), and click **OK** when you are done.
5. Click **OK** to apply the settings and exit System Properties.
6. Stop and restart the content server (otherwise the change will not take effect).



Note: If you edited the locale that is set as the current system locale, you also need to restart the Admin server.

Adding a New System Locale

The list of standard system locales on the Localization tab of System Properties includes definitions for most Western European and some common Asian languages. Please note that the languages in the list can only be set as the system locale if both of the following conditions are met:

- ❖ A correctly named [resource file directory](#) exists for the language.
- ❖ This directory contains the appropriate (localized) [resource files](#).



Note: If you install a language pack for your Content Server version, all requisite directories and files are set up automatically and you can [change the system locale](#) (see page 6-2), if desired.

Resource File Directory

The **resource file directory** for a locale is located in the following directory:

`[Install_Dir]/shared/config/resources/lang/[Lang_Code]`

where `[Install_Dir]` is the installation directory of the content server instance and `[Lang_Code]` is the letter code assigned to the language.



Note: The language code is also used to refer to the help system of the language (if available). The localized help system for a language is located in `[Install_Dir]/weblayout/help/[Lang_Code]`.

Resource Files

Each language-specific resource directory **must** contain the following four **resource files**, which hold (localized) user interface elements and output messages:

- ❖ `ap_strings.htm`
- ❖ `cs_strings.htm`
- ❖ `sy_strings.htm`
- ❖ `ww_strings.htm`



Caution: If any of these files is not present and the system locale is set to the corresponding language, the content server will not start.



Note: If you want to use the Classic (i.e., pre-7.0) layout, you also need to make sure there is a directory called `[Install_Dir]/weblayout/images/stellent/[Lang_Code]` with an image file called `corpinfo.gif`. Otherwise there will be a missing graphic on the portal page.



Tech Tip: If you want to use the English resource files for a system locale, make sure that you use the files at `[Install_Dir]/shared/config/resources/lang`. There is a `.../lang/en` subdirectory, but this contains only “dummy” resource files that are used for reference purposes.

Enabling a System Locale

Once all required files of the new system locale are where they should be, you can make the system locale available to Content Server:

1. **UNIX:** Start the System Properties utility by running the *System_Properties* script, which is located in the *bin* subdirectory of the Content Server installation directory.

Windows: From the Windows Start menu, select Programs— Content Server— *[Instance Name]*—Utilities—System Properties.

The System Properties applet is started.

2. Open the Localization tab.
3. Click **Add**.

The Configure Locale screen is displayed.

4. Enter or choose all parameters (see [Locale Properties](#) on page 6-6), and click **OK** when you are done.
5. Click **OK** to apply the settings and exit System Properties.
6. Stop and restart the content server (otherwise the change will not take effect).

Locale Properties

Each locale has the following properties:

- ❖ **Locale**—This is the locale identifier. If you are editing an existing locale, this field cannot be changed (it is shown for information purposes only). For new system locales, you can specify any name. The name you enter here will be displayed in the System Locale dropdown list on the Server tab of System Properties (providing the [criteria](#) on page 6-3 are met).
- ❖ **Language code**—This is the letter code assigned to the language. This code is used to refer to the [resource file directory](#) and help system directory (see page 6-5), which need to have the exact same name as this language code. This means that if you create a new locale with language code ‘hu’, its corresponding resource file directory, help system directory, and portal image directory (if required) must be called ‘hu’ as well:
 - *[Install_Dir]*/shared/config/resource/lang/hu
 - *[Install_Dir]*/weblayout/help/hu
 - *[Install_Dir]*/weblayout/images/stellent/hu (for the Classic layout; see page 6-5)

If the resource file directory does not contain the required [resource files](#) (see page 6-5), the content server will not start. If the help system directory does not exist or if it is empty, the default English help system is used.

Even though, in principle, you can use any code you want, it is recommended that you adhere to the two- and three-letter conventions laid down in the ISO 639-1 and 639-2 standards.

- ❖ **Verity locale** (only if Verity search engine is used)—This is the language locale used by Verity, which is one of the search engines that can be used with Content Server. It defines how special characters in text are indexed. For details refer to [Verity](#) (page 7-4).
- ❖ **Encoding**—This is the encoding scheme that is used in the communication with the content server. There is usually no need to change this parameter for existing system locales as it is already optimized for each language. If you are creating a new system locale, you need to select an encoding scheme that is compatible with the language.
- ❖ **Date/time format**—This parameter defines how the date and time are displayed on screen. By default, Content Server uses the system's short date (numeric-only) and time format everywhere in the Content Server user interface. You can override the system's display of the short date or time format by changing this parameter. There is usually no need to change this parameter for existing system locales as it is already optimized for each language. For details on the date and time format refer to [Appendix A \(Date and Time Format\)](#).
- ❖ **Time zone**—This is the time zone for this locale. This is something you may want to change, especially if the language area spans several time zones. For example, the default time zone for the English-US system locale is America/New_York (i.e., Eastern Standard Time). If the content server is located in Los Angeles, you may want to change this locale setting to America/Los_Angeles (i.e., Pacific Standard Time). This ensures that time stamps reflect the actual time at the content server location.



Note: The Enabled option is not related to the system locale, but to the corresponding user locale. System locales can be available (by meeting the [criteria](#) on page 6-3), without the corresponding user locale being enabled. For details on user locales refer to [User Locales](#) (page 6-8).

System Locale Considerations

Please note the following important consideration with regard to the Content Server system locale:

- ❑ If you want to use a single-byte system locale (for example, English) with double-byte [user locales](#) (for example, Japanese) or vice versa, you need to make sure the following requirements are met:
 - The content server [file encoding](#) is set to UTF-8 (see chapter 4).
 - Everyone accessing the content server has the required language packs installed on their operating systems.
 - Everyone accessing the content server has the correct languages set up in their [web browsers](#) (see chapter 8).
- ❑ See the *Content Server Troubleshooting Guide* for more internationalization issues.

USER LOCALES

System administrators can set up the Content Server software to offer different user environments based on user locales. These are sets of user preference information related to the user interface language, environment, and cultural conventions (for example, date and time formats). Users can specify their own user locale in their user profile, which is accessible from the content server's portal page. If localized versions of the user interface and help files of Content Server are available for the selected user locale, the Content Server environment will be presented in localized form (for example, German, French, or Spanish).

This section covers the following topics:

- ❖ [Enabling User Locales](#) (page 6-8)
- ❖ [Available User Locales](#) (page 6-9)
- ❖ [User Locale Considerations](#) (page 6-10)



Important: [System locales](#) (see page 6-1) and user locales are closely related. They use the same locale definitions as specified on the Localization tab of System Properties. This means you cannot have a system locale and user locale with the same name, but different definitions.

Enabling User Locales

User locales can only be selected if they are explicitly made available to the users. To make user locales available to users, complete the steps below.



Note: The user locale that corresponds to the active system locale will always be available, and does not need to be enabled using the procedure described below. For example, with the system locale set to ‘English-US’, the corresponding user locale does not need to be specifically enabled to be available to users. This means that if you do not enable any user locales, the “system” user locale is the only one available.

1. **UNIX:** Start the System Properties utility by running the *System_Properties* script, which is located in the *bin* subdirectory of the Content Server installation directory.

Windows: From the Windows Start menu, select Programs— Content Server—*[Instance Name]*—Utilities—System Properties.

The System Properties applet is started.

2. Open the Localization tab.
3. In the list of locales, select the user locale(s) you want to make available to users, and click **Enable**. You can select multiple locales by holding the Shift key (consecutive selection) or Ctrl key (non-consecutive selection) while clicking on the locale entries.
4. When you are done, click **OK** to apply the setting and exit System Properties.
5. Restart each instance of the content server.



Caution: If required, you can edit the properties of a locale by selecting the locale entry from the list and pressing **Edit** (see [Editing an Existing System Locale](#) on page 6-4 for details). Please be aware that any changes apply not only to the user locales, but also the system locale. In other words, the properties of the system locale and the corresponding user locale will always be identical.

Available User Locales

Essentially, all user locales in the list can be enabled. If localized versions of the user interface and help files are available for an enabled user locale, the Content Server environment will be presented in localized form. Otherwise, the user interface will be English, but other elements will follow the cultural conventions of the locale, such as date/time format, sort order, and the like.

Localized user interfaces for Content Server 10gR3 will be made available in the following languages:

- ❖ German (Deutsch)
- ❖ French (Français)
- ❖ Spanish (Español)

- ❖ Brazilian Portuguese (Português) *[user interface only]*
- ❖ Dutch (Nederlands) *[user interface only]*
- ❖ Japanese
- ❖ Korean *[user interface only]*

Language Packs

Optional language packs are used to provide localized user interfaces for Content Server (and also localized user help files for some languages). Language packs can be downloaded from Oracle's support website. Installation instructions are provided with the language pack.



Important: Language packs will be made available for all major Content Server releases. There may be some lag time between the release of a new Content Server version and the availability of its associated language packs.

User Locale Considerations

Please note the following important considerations with regard to user locales:

- You can make multiple user locales available to users, who then choose one to use. If you want to use a single-byte system locale (for example, English) with double-byte user locales (for example, Japanese) or vice versa, you need to make sure the following requirements are met:
 - The content server [file encoding](#) is set to UTF-8 (see chapter 4).
 - Everyone accessing the content server has the required language packs installed on their operating systems.
 - Everyone accessing the content server has the correct languages set up in their [web browsers](#) (see chapter 8).
- The user locale that corresponds to the active system locale will always be available, and does not need to be enabled. For example, with the system locale set to 'English-US', the corresponding user locale will always be available, even if its 'Enabled' status on the Localization tab of System Properties has been set to 'False.'
- See the *Content Server Troubleshooting Guide* for more internationalization issues.

SEARCH SOLUTION SETTINGS

OVERVIEW

Several options are available to provide searching and indexing capabilities for Content Server 8.x. New installations are set up to use the database to provide searching and indexing. The out-of-the-box database configuration provides metadata-only searching, but you can modify the default configuration for some of the supported databases to enable full-text searching. For further details refer to the Content Server installation guides.

You can also use an external search engine in place of the content database to provide full-text and/or metadata search functionality to the content server. Integration modules are available for the following external search solutions:

- ❖ Verity K2 (www.autonomy.com)
- ❖ FAST InStream (www.fastsearch.com)



Note: A useful document in this respect is *Choosing a Search Solution*, which discusses the various search options that can be used with Content Server, as well as the factors to consider when selecting a solution. It is available as a PDF file on the Content Server documentation CD (*integrator\search_solutions_80en.pdf*), and is also included in Content Server's online help system for administrators.



Tech Tip: If you upgraded to Content Server 8.x from an earlier release with Verity K2 4.5.1, the Verity search engine will continue to be used unless you modify the configuration.

The settings of the search solution used with your content server may affect the content server's ability to handle multilingual content. The considerations in this area depend on the search solution used:

- ❖ [Database](#) (page 7-2)
- ❖ [Verity](#) (page 7-4)
- ❖ [FAST InStream](#) (page 7-11)

DATABASE

If the database is used for metadata and/or full-text searching, the following considerations should be taken into account with regard to international environments, depending on the database:

- ❖ [SQL Server](#) (page 7-2)
- ❖ [Oracle](#) (page 7-3)
- ❖ [DB2](#) (page 7-3)
- ❖ [Sybase](#) (page 7-4)
- ❖ [PostgreSQL](#) (page 7-4)

SQL Server

Please note the following internationalization considerations when using SQL Server to provide searching capabilities to the content server:

- ❖ Both metadata and full-text searching can be provided. For new Content Server 8.x installations, only metadata searching is enabled by default, but the configuration can be modified to offer full-text searching as well. See the Content Server installation guides for further details.
- ❖ Content in different languages can be indexed, and users can search across documents in different languages. The following languages are supported: English, German, French, Spanish, Portuguese, Italian, Greek, Dutch, Danish, Swedish, Norwegian (both Bokmål and Nynorsk), Finnish, Hungarian, Polish, Czech, Russian, Japanese, Korean, and Chinese (both traditional and simplified).
- ❖ Languages are automatically detected.

Oracle

Please note the following internationalization considerations when using Oracle to provide searching capabilities to the content server:

- ❖ Both metadata and full-text searching can be provided. For new Content Server 8.x installations, only metadata searching is enabled by default, but the configuration can be modified to offer full-text searching as well. See the Content Server installation guides for further details.
- ❖ Content in different languages can be indexed, and users can search across documents in different languages. The following languages are supported: English, German, French, Spanish, Portuguese, Italian, Greek, Dutch, Danish, Swedish, Norwegian (both Bokmål and Nynorsk), Finnish, Hungarian, Polish, Czech, Russian, Japanese, Korean, and Chinese (both traditional and simplified).
- ❖ Languages are automatically detected in Oracle 10g. Oracle9i requires a language column in the database to support multiple languages.
- ❖ All Oracle NLS character sets are supported (including ASCII, UTF-8, JA16SJIS, GBK, BIG5, etc.).
- ❖ Mixed case is supported for Chinese and Japanese.

DB2

Please note the following internationalization considerations when using DB2 to provide searching capabilities to the content server:

- ❖ Both metadata and full-text searching can be provided. For new Content Server 8.x installations, only metadata searching is enabled by default, but the configuration can be modified to offer full-text searching as well. See the Content Server installation guides for further details.
- ❖ Content in different languages can be indexed, and users can search across documents in different languages. The following languages are supported: English, German, French, Spanish, Portuguese, Italian, Greek, Dutch, Danish, Swedish, Norwegian (both Bokmål and Nynorsk), Finnish, Hungarian, Polish, Czech, Russian, Japanese, Korean, and Chinese (both traditional and simplified).
- ❖ Languages are not automatically detected.

Sybase

Please note the following internationalization considerations when using Sybase to provide searching capabilities to the content server:

- ❖ Only metadata searching is provided.
- ❖ Metadata in different languages can be indexed, and users can search across documents in different languages. The following languages are supported: English, German, French, Spanish, Portuguese, Italian, Greek, Dutch, Danish, Swedish, Norwegian (both Bokmål and Nynorsk), Finnish, Hungarian, Polish, Czech, Russian, Japanese, Korean, and Chinese (both traditional and simplified).

PostgreSQL

Please note the following internationalization considerations when using PostgreSQL to provide searching capabilities to the content server:

- ❖ Only metadata searching is provided.
- ❖ Metadata in different languages can be indexed, and users can search across documents in different languages. The following languages are supported: English, German, French, Spanish, Portuguese, Italian, Greek, Dutch, Danish, Swedish, Norwegian (both Bokmål and Nynorsk), Finnish, Hungarian, Polish, Czech, Russian, Japanese, Korean, and Chinese (both traditional and simplified).

VERITY

Verity is one of the external search solutions that can be used in conjunction with Content Server. For instructions on how to set up your content server for use with the Verity search engine, refer to the *Verity Integration Guide*.

If Verity is used for metadata and/or full-text searching and a content item is checked in, its text and metadata are passed through to the embedded search engine, extracted, and then indexed. When users search for content by metadata or keywords, a query is issued against the search index, not the database. The results can be sorted by any of the metadata fields or based on a relevancy score assigned by the search engine.

The Verity search engine integration can be started and stopped with the content server. It can be customized using administrative tools. Rebuilds of the index can be performed within the Content Server administration utilities.

The following topics are important with regard to international environments:

- ❖ [Verity Locales](#) (page 7-5)
- ❖ [Supported Verity Locales](#) (page 7-5)
- ❖ [Changing the Verity Locale](#) (page 7-7)
- ❖ [Checking the Verity Version](#) (page 7-8)
- ❖ [Verity Considerations](#) (page 7-8)

Verity Locales

Verity is one of the indexing and search engines that can be used with Content Server to provide full-text search capability. There are a number of “language locales” for Verity, which are used to make sure language-specific characters in text are indexed correctly.

All content server instances (including proxied servers) have their own Verity locales. There can only be one Verity locale per content server instance. The default Verity locale for an instance is set to ‘englishx’ upon installation of the Content Server software. This is a locale that supports the ISO-8859-1 code page, which means that it can handle special characters in most Western European languages, such as ä, Ö, ß, etc. (German), é, â, ç, etc. (French), and ó, ñ, á, etc. (Spanish).

If you want to index text in other languages than English, German, French, Spanish, Italian, Portuguese, Dutch, Danish, Swedish, Norwegian, and Finnish (all sharing the ISO-8859-1 file encoding), you need to add a parameter to Content Server’s *[Install_Dir]/bin/intradoc.cfg* configuration file (where *[Install_Dir]* is the installation directory of the content server instance). See [Changing the Verity Locale](#) (page 7-7) for further details.

Supported Verity Locales

The following table shows the Verity locales that are supported by Content Server:

Verity locale	Languages that Verity index engine can handle
english	English (2)

Verity locale	Languages that Verity index engine can handle
englishv (= default)	All Western European languages (1)
frenchv	All Western European languages (1)
germanv	All Western European languages (1)
spanishv	All Western European languages (1)
italianv	All Western European languages (1)
portugv	All Western European languages (1)
dutchv	All Western European languages (1)
danishv	All Western European languages (1)
swedishv	All Western European languages (1)
bokmalv	All Western European languages (1)
finnishv	All Western European languages (1)
japanb	Japanese + English (2)
koreab	Korean + English (2)
simpcb	Simplified Chinese + English (2)
tradcb	Traditional Chinese + English (2)
uni (3)	All languages and language combinations. In addition, this includes Arabic, Czech, Greek, Hebrew, Hungarian, Nynorsk, Polish, Russian, and Turkish.

Notes:

1. ‘Western European languages’ in the table refers to the following languages (in addition to English): German, French, Spanish, Italian, Portuguese, Dutch, Danish, Swedish, Norwegian, and Finnish (all sharing the ISO-8859-1 encoding.)
The differences between the locales are related to the language in which Verity-related messages are presented and the way results are presented in the search results pages, etc.

2. ‘English’ in the table essentially refers to all ASCII characters—i.e., a-z, A-Z, 0-9, and common punctuation marks (comma, colon, question mark, etc.), but not special letters such as é, Å, ö, ñ, ß, etc.
3. The ‘uni’ locale is not supported by Verity VDK 4.5.1; it requires VDK 5.x or higher. If required, contact [Oracle support](#) for assistance.



Note: As shown in the table above, the ‘uni’ Verity locale (UTF-8) provides the most comprehensive indexing and searching capabilities. If you anticipate that the content server will handle multilingual content from various language groups, it is recommended that you use this Verity locale.

Changing the Verity Locale

You generally only need to modify the default Verity locale (‘englishv’) if you want the content server (and Verity) to handle non-Western European content (for example, Asian languages and/or Eastern European languages). In that case, you need to add a parameter to one of the configuration files of Content Server:

1. Go to the file `[Install_Dir]/bin/intradoc.cfg` (where `[Install_Dir]` is the installation directory of the content server instance), and open it in a text editor.
2. Add the following entry:

```
VerityLocale=[value]
```

 where *[value]* is any of the Verity locales in the table in [Supported Verity Locales](#) (page 7-5), for example:

```
VerityLocale=japanb
```
3. Save the modified configuration file and exit the text editor. (If a warning is displayed about saving the file to a text-only format, go ahead and confirm the action.)
4. Restart the content server instance.



Caution: If the new Verity locale does not use the same encoding scheme as the old one (for example, from ‘englishx’ to ‘japanb’), you need to rebuild the search index. This may be a very time-consuming process, depending on the number of content items managed by your content server instance. It is therefore recommended that you perform the index rebuild during off-peak hours of content server use (typically at night or on the weekend).

Checking the Verity Version

To find out what Verity version is currently running with a particular content server instance, complete the following steps:

1. Log on to the content server instance as an administrator.
2. Check a new document into the content server, or (temporarily) update the metadata of an existing document (for example, its title).
3. Go to the Administration page.
4. Click on **Admin Server**.
5. Click on the appropriate server button to the right of the start–stop–restart icons.
6. In the options menu on the left, click **View Server Output**.
7. The output will show a number of lines that start with the word ‘indexer’.
One of these lines will contain the version of the Verity engine used, for example:
Verity K2 version 4.5.1

Verity Considerations

Please note the following important considerations with regard to the Verity search engine:

- ❑ You can only use one Verity locale per content server instance. If you want Verity to handle content in languages that do not belong to the same language group, you *must* use ‘uni’ as the Verity locale. For example:
 - combinations of Western European languages and Asian languages (for example, English and Japanese)
 - combinations of Western and Eastern European languages (for example, English, German, and Russian)
 - combinations of Asian languages (for example, Japanese and Korean)



Important: The ‘uni’ locale is not supported by Verity VDK 4.5.1; it requires VDK 5.x or higher. If required, contact [Oracle support](#) for assistance.

- ❑ You can use search operators to broaden or narrow your Verity full-text search. Some common search operators include *AND*, *OR*, and *NOT*. Verity supports localized (i.e., non-English) search operators. However, you need to enclose these in angle brackets, for example: “système <ET><SAUF> gestion.”

The table below shows some localized versions of the AND, OR, and NOT search operators.

Language	Search Operator		
English	AND	OR	NOT
French	<ET>	<OU>	<SAUF>
German	<UND>	<ODER>	<NICHT>
Spanish	<Y>	<O>	<EXCEPTO>
Portuguese	<E>	<OU>	<SALVO>



Tech Tip: If you are using a non-English [Verity locale](#), you can still use English search operators by enclosing them in angle brackets and using a hash symbol (for example, “système <#AND><#NOT> gestion.”)

- ❑ The content ID must be supplied in single-byte characters. Multi-byte characters for the content ID are not supported by Verity. It is recommended that you autogenerate your content IDs (Administration—Admin Server—*[Instance Name]* button—General Configuration—Automatically assign a content ID on check-in).

You can also hide the content ID field on the content check-in page by adding the following parameter to *[Install_Dir]/config/config.cfg* (where *[Install_Dir]* is the installation directory of the content server instance):

```
dDocName:isHidden=1
```

- ❑ **Verity 4.5 only:** Full-text indexing of PDF files is not always handled correctly for multi-byte and bidirectional languages if the default Verity settings are used. To fix this, you must edit Verity’s *style.uni* to use the keyview filter for PDFs:
 1. Add the entry `UseVdk4CustomStyle=true` to *[CS_Instance_Dir]/config/config.cfg*.
 2. Save the file, and restart the content server.
 3. Create a directory called `vdk4_custom_style` under the *[CS_Instance_Dir]/search* directory.
 4. Copy all files from the *[CS_Instance_Dir]/shared/search/style/basic* directory to the *[CS_Instance_Dir]/search/vdk4_custom_style* directory.
 5. Start a text editor and open *style.uni* in the *[CS_Instance_Dir]/search/vdk4_custom_style* directory.

6. Comment out the line `/format-filter = "flt_pdf"` under `type: "application/pdf"` and uncomment `/format-filter = "flt_kv"`, for example:

```
type: "application/pdf"
# /format-filter = "flt_pdf"
/format-filter = "flt_kv"
```

7. Save and rebuild the search index.



Note: Even using the “flt_kv” filter, full-text indexing of PDF files is not always indexed correctly for multi-byte characters. You can fix this by using the `TextIndexerFilter` component.

- Verity 4.5 only:** Search word highlighting is not supported for multi-byte locales. This feature should be turned off in Content Server if your [system locale](#) is a multi-byte language (for example, Japanese or Korean).

To do this, proceed as follows:

1. **UNIX:** Start the System Properties utility by running the `System_Properties` script, which is located in the `bin` subdirectory of the Content Server installation directory.

Windows: Choose Start—Programs—Content Server—*[Instance Name]*—Tools—System Properties (make sure you have administrator rights).

2. Make sure the Options tab is opened.
 3. Make sure the “Enable search keyword highlighting” check box is cleared, and click **OK**.
 4. Restart the content server instance.
- If you are having problems finding full-size and half-size Japanese katakana in your checked-in documents, you may need a patch from Verity and/or some modifications to the Verity configuration. Please contact support for more information.
 - If PDF bookmarks for Word documents with multi-byte characters in their headings are not displayed correctly, make sure that you enable the “Unicode signatures for PDF bookmarks” option in Inbound Refinery (local configuration setting). This setting specifies that Unicode character coding should be used when creating PDF bookmarks for Microsoft Word documents rather than ASCII. Unicode uses 16 bits, which means that non-ASCII characters can be used in the PDF bookmarks.

FAST InSTREAM

FAST InStream is one of the external search solutions that can be used in conjunction with Content Server. For instructions on how to set up your content server for use with the FAST search engine, refer to the *FAST Integration Guide*.

If FAST is used for metadata and/or full-text searching and a content item is checked in, its text and metadata are passed through to the embedded search engine, extracted, and then indexed. When users search for content by metadata or keywords, a query is issued against the search index, not the database.

Important Considerations

Please note the following considerations with regard to the FAST InStream search engine:

- ❖ Metadata in different languages can be indexed, and users can search across documents in different languages. The following languages are supported: English, German, French, Spanish, Portuguese, Italian, Dutch, Swedish, Norwegian (both Bokmål and Nynorsk), Finnish, Hungarian, Polish, Russian, Japanese, and Korean.
- ❖ Languages are automatically detected.
- ❖ PDF highlighting is supported.
- ❖ Both metadata and full-text searching can be handled.

WEB BROWSER SETTINGS

OVERVIEW

This section covers the following topics:

- ❖ [Adding Language Support](#) (page 8-1)
- ❖ [Browser Considerations](#) (page 8-2)

ADDING LANGUAGE SUPPORT

Administrators, consumers, and contributors use a web browser to interface with the content server. To correctly display web pages encoded in any language, they need to set up their web browsers to support the appropriate language(s):

- ❖ In **Internet Explorer**, this is done through Tools—Internet Options—General—Languages.
- ❖ In **Netscape**, this is done through Edit—Preferences—Navigator—Languages.
- ❖ In **Mozilla**, this is done through Edit—Preferences—Navigator—Languages.
- ❖ In **Firefox**, this is done through Tools—Options—Advanced—General tab—Edit Languages.
- ❖ In **Safari**, this is done through Safari—Preferences—Appearance—Default Encoding.

For further details, refer to the help files or documentation of your web browser.

You can add as many languages as you want. You can move languages up and down in the list. The list of languages is processed from top to bottom, which means that languages higher in the list take precedence over languages lower in the list.



Important: Adding languages does not guarantee that your computer has a font that can display web pages in your preferred languages. If a required font is missing, characters are typically displayed as little squares. Depending on your web browser, you may be prompted when fonts need to be added. For details refer to your web browser's documentation.



Note: If you are using a web browser in the same language as the content you want to access, you typically do not need to make any changes to the settings in your web browser.

BROWSER CONSIDERATIONS

Please note the following important considerations with regard to web browsers:



Note: For more troubleshooting information related to internationalization, refer to the *Content Server Troubleshooting Guide*.

Text in Forms With Re-used Data Is Displayed in &#xxxx; Form

In Internet Explorer, text in form fields that contain “re-used” data—for example, prefilled data fields on the “check-in similar” form or document update form—are displayed in Unicode form (&#xxxx;).

If you type a character into an Internet Explorer form and the current encoding of that page does not support that character, Internet Explorer does not stop you from actually entering the character. Instead, when you perform the post, it will encode the character in its Unicode encoding (&#xxxx;). This works fine as long as you view the same content inside an HTML page for a browser. However, problems arise if you try to use the content in a JavaScript construction, try to repurpose the values into other formats (such as text files), try to search for it, or try to make sure there is enough room in a database field to hold it.

You can prevent problems like this by using Unicode encoding in the [database](#) (see chapter 5).

Some Bookmarked URLs Don't Work in Internet Explorer

Some bookmarked URLs in Internet Explorer do not work, even though the links themselves are valid.

If your content server has been configured to use UTF-8 file encoding and you have a URL that has UTF-8 multi-byte sequences in it that have not been URL-escaped (%xx escape sequence), then the URL will work in Internet Explorer if clicked inside UTF-8 pages, but it will not work when bookmarked. Please note that this is not a problem with Mozilla-based browsers.

If you set the `XmlEncodingMode=full` parameter in `<CS_Instance_Dir>/config/config.cfg`, it will reduce the consequences of this problem because many more parts of the content server CGI URL parameters will be fully 7-bit encoded. However, some content server URL constructions will not URL-escape the `dDocName` parameters (document name).

If you wish to avoid this problem with a UTF-8 content server altogether, then you will need to either do some customization of the content server pages or make sure your all `dDocName` values are clean 7-bit.

File Name Characters in "Save As" Dialogs Not Displayed Correctly

If you use Internet Explorer to download a file from the content server and you go through the "Save As" dialog, Internet Explorer assumes that the file name given to it by the content server is in the default encoding of the client desktop. This is a particular problem for UTF-8 systems, where all non 7-bit characters tended to get garbled in the "Save As" dialog. In systems where the content server and desktop seemingly have the same encoding, small discrepancies can occur. The typical difference is the difference between a server whose operating system is in the UNIX family and a client desktop running Windows. As a particular example, the default implementation of `shift_jis` on Linux is an encoding called `shift_jis`. This varies by a few critical characters from MS932, the encoding used by Windows to implement `shift_jis`. Even if you train the server to explicitly use MS932, it will still be short a few characters (but fewer) from the MS932 on Windows.

In Content Server 7.5 and higher, the issue is partially addressed by using the user's locale to guess at the current desktop encoding. For example, if the user's locale is Japanese and the Japanese locale is associated with the `shift_jis` encoding, then that encoding is associated with that particular logged-in user. When it comes time to produce a suggested filename, the content server looks up the default internal Java encoding used to implement `shift_jis`, finds this to be MS932, and uses the MS932 encoding to produce the bytes to

send for the suggested filename. Different logged-in users can potentially get different encodings for creating the suggested filenames.

Please note that this is not a problem with Mozilla-based browsers.



Note: This issue also applies to the Outlook Express e-mail client.

ENCODING COMPATIBILITY

ENCODING COMPATIBILITY

It is crucial that the file encodings of the content server, database, and Verity search engine (if used) are all compatible. There is a lot of data traffic between these three system components, and incompatible file encodings will result in communication errors, display problems, etc.

The table on the next page provides recommended file encodings for various scenarios, depending on the content language(s) that need to be supported by the content server.



Note: You can ignore the Verity Encoding column if you are using database searching or the FAST search engine. They are preconfigured to support all languages and do not use locales.

As shown in the table below, the most comprehensive language support is achieved by using UTF-8 as the Content Server file encoding, Unicode as the database encoding, and 'uni' as the Verity encoding (if applicable). This basically enables Content Server to work with content in virtually any language in the world.

Language(s) to be supported by Content Server	Content Server encoding (3)	Database encoding	Verity encoding (4)
English only (1)	no changes required to default setting (= ISO-8859-1)	ISO-8859-1	englishv (= default) or english

Language(s) to be supported by Content Server	Content Server encoding (3)	Database encoding	Verity encoding (4)
English (1) + multiple Western European languages (2)	no changes required to default setting (= ISO-8859-1)	ISO-8859-1	englishv (= default)
English + one Western European language (2)	no changes required to default setting (= ISO-8859-1)	ISO-8859-1	frenchv, germanv, italianv, spanishv, portugv, dutchv, danishv, swedishv, bokmalv, finnishv (depending on the language) (5)
English + Russian only	ISO-8859-5	ISO-8859-5	uni (6)
English + Czech only	UTF-8	Unicode or UTF-8	uni (6)
English + Greek only	ISO-8859-7	ISO-8859-7	uni (6)
English + Turkish only	ISO-8859-9	ISO-8859-9	uni (6)
English + Hungarian only	ISO-8859-2	Unicode or UTF-8	uni (6)
English + Polish only	ISO-8859-2	Unicode or UTF-8	uni (6)
English + any combination of Western European languages and Russian/Czech/Greek/Turkish/Hungarian/Polish	UTF-8	Unicode or UTF-8	uni (6)
English + Japanese only (on Sun Solaris)	UTF-8	UTF-8 (7)	japanb
English + Japanese only (on other UNIX flavors and Windows)	SJIS	SJIS	japanb
English + Korean only	KSC5601	KSC5601	koreab

Language(s) to be supported by Content Server	Content Server encoding (3)	Database encoding	Verity encoding (4)
English + Traditional Chinese only	BIG5	Unicode or UTF-8	tradcb
English + Simplified Chinese only	GB2312	Unicode or UTF-8	simpcb
English + all European languages (incl. Russian, Greek, and/or Turkish)	UTF-8	Unicode	uni (6)
English + one or more Asian language(s) + one or more European language(s)	UTF-8	Unicode	uni (6)
English + multiple Asian languages	UTF-8	Unicode	uni (6)

Notes:

1. ‘English’ in the table essentially refers to all ASCII characters—i.e., a-z, A-Z, 0-9, and common punctuation marks (comma, colon, question mark, etc.), but not special letters such as é, Å, ö, ñ, ß, etc.
2. ‘Western European languages’ in the table refers to the following languages (in addition to English): German, French, Spanish, Italian, Portuguese, Dutch, Danish, Swedish, Norwegian, and Finnish (all sharing the ISO-8859-1 encoding.)
3. The Content Server file encoding is controlled by the `FileEncoding` parameter in `[Install_Dir]/bin/intradoc.cfg` (where `[Install_Dir]` is the installation directory of the content server instance), for example:

```
FileEncoding=UTF-8
```

See [Chapter 4 \(Content Server File Encoding\)](#) for further details.

4. The Verity language encoding (“locale”) is controlled by the `VerityLocale` parameter in `[Install_Dir]/bin/intradoc.cfg` (where `[Install_Dir]` is the installation directory of the content server instance), for example:

```
VerityLocale=japanb
```

For further details refer to [Chapter 5 \(Database Data Encoding\)](#) and [Chapter 7 \(Search Solution Settings\)](#).

5. These Verity locales all support the ISO-8859-1 code page, which means they can all handle special characters in most Western European languages (for example, ä, ß, é, ç, ó, ñ, æ). The differences between these locales are related to the language in which Verity-related messages are presented and the way results are presented in the search results pages, etc.
6. The 'uni' locale is not supported by Verity VDK 4.5.1; it requires VDK 5.x or higher. If required, contact [Oracle support](#) for assistance.
7. The web server *must* also use UTF-8 encoding.

Chapter 10

INTERNATIONAL SCENARIOS

OVERVIEW

By default, the Content Server 8.x software is installed with all U.S. English settings, i.e.:

- ❖ The [system locale](#) is set to English-US.
- ❖ English-US is the only [user locale](#) available.



Note: If the Verity integration module is installed, the [Verity locale](#) is set to ‘englishx’ by default.

You may need to modify the default configuration settings, depending on the language environment that Content Server is to work in:

- ❖ [English-Only Environments](#) (page 10-1)
- ❖ [European Language Environments](#) (page 10-2)
- ❖ [Asian and Other Non-European Language Environments](#) (page 10-4)
- ❖ [Mixed Language Environments](#) (page 10-10)

ENGLISH-ONLY ENVIRONMENTS

In English-only environments, Content Server only handles English files, content, and metadata, and only English text needs to be included in the full-text search index.

The Content Server software is generally installed on an English-language operating system.

English-only environments like these typically do not require any special modifications to the file and language encoding settings of the out-of-the-box content server.

EUROPEAN LANGUAGE ENVIRONMENTS

The European language environments can be divided into two groups:

- ❖ [Western European languages](#)
- ❖ [Other European languages](#)

Western European Languages

In Western European language environments, Content Server typically handles files, content, and metadata in English, German, French, Spanish, Italian, Portuguese, Dutch, Danish, Swedish, Norwegian and/or Finnish, and text in these languages needs to be included in the full-text search index. The Content Server software can be installed on an English-language or localized operating system (for example, the German or French version of Microsoft Windows).

Western European environments like these typically do not require any special modifications to the file and language encoding settings of the out-of-the-box content server. All Western European languages share the same file encoding (ISO-8859-1), which means they can generally be mixed without any issues.

Search Solution

If you are using the Verity search engine to provide search capability to the content server, the Verity locale is set to 'englishx' by default. This Verity locale supports full-text indexing of special characters in all Western European languages (for example, ä, ß, é, ç, ó, ñ, æ). Each language also has its own Verity locale (for example, 'germanx', 'frenchx', etc.). If you want, you can configure Verity to use the language-specific locales. All European Verity locales ending in 'x' can handle European special characters such as ä, ß, é, ç, ó, ñ, æ. The differences between these locales are related to the language in which Verity-related messages are presented and the way results are presented in the search results pages, etc. For details on changing the Verity locale refer to [Changing the Verity Locale](#) (page 7-7).

If you are using the database or FAST to provide search capability to the content server, no changes to the out-of-the-box content server configuration are necessary.

Other European Languages

There may also be European language environments in which Content Server needs to handle files, content, and metadata in one or more non-Western European languages (for example, Russian, Greek, or Turkish)—possibly in addition to Western European languages—and text in these languages needs to be included in the full-text search index. The Content Server software can be installed on an English-language or localized operating system (for example, the Russian or Turkish version of Microsoft Windows).

Environments like these may require some modifications to the file and language encoding settings of the out-of-the-box content server. The exact modifications depend on the setup and combination of languages to be supported.

Modifications to Configuration File

If you want the content server to only handle a single non-Western European language in addition to English (for example, only Russian and English), you could choose to use the file and language encodings specific to that language. However, this will typically limit the character sets that the content server can work with. It is generally better to use UTF-8 and Unicode encodings as these provide the most comprehensive character set support and allow virtually any languages to be mixed without any issues.

Whichever path you choose, you need to make some modifications to the `[Install_Dir]/bin/intradoc.cfg` configuration file (where `[Install_Dir]` is the installation directory of the content server instance):

- ❖ You need to change the **Content Server file encoding** by adding the following line to the `intradoc.cfg` file:

```
FileEncoding= [value]
```

(where `[value]` is the [appropriate value for the language](#) as provided in the table on page 4-3.)



Note: If you are running any refinery products, you also need to add the `FileEncoding` entry for each connection. Add the parameter to the file `[Ref_Dir]/Connections/[Connection_Name]/intradoc.cfg` (where `[Ref_Dir]` is the installation directory of Inbound Refinery and `[Connection_Name]` the name of the refinery connection). Make sure that you restart the content server instance that the refinery connection links to.

- ❖ If you are using the Verity search engine, you also need to change the **Verity locale** by adding the following line to the `intradoc.cfg` file:

```
VerityLocale= [value]
```

(where *[value]* is the [appropriate value for the language](#) as provided in the table on page 7-5.)

Save the *intradoc.cfg* file and make sure that you restart the content server and Admin server.

Example

If, for example, you want the content server to support English and Russian (and no other languages), you would modify the *intradoc.cfg* configuration file to include these two lines:

```
FileEncoding=ISO-8859-5  
VerityLocale=russian [if you are using the Verity search engine]
```

If you want the content server to support English, Russian, and other languages (European or non-European), you would modify the *intradoc.cfg* configuration file to include these two lines:

```
FileEncoding=UTF8  
VerityLocale=uni [if you are using the Verity search engine]
```



Note: In the second example, you also need to make sure the database is configured to support Unicode (see chapter 5).

Important Considerations

Please note the following considerations for pan-European environments:

- After modifying the VerityLocale parameter, you need to rebuild the search index. This may be a very time-consuming process, depending on the number of content items managed by your content server instance. It is therefore recommended that you perform the index rebuild during off-peak hours of content server use (typically at night or on the weekend).
- The ‘uni’ locale is not supported by Verity VDK 4.5.1; it requires VDK 5.x or higher. If required, contact [Oracle support](#) for assistance.

ASIAN AND OTHER NON-EUROPEAN LANGUAGE ENVIRONMENTS

In Asian and other non-European language environments, Content Server needs to handle files, content, and metadata in one or more non-European languages (for example,

Japanese, Korean, or Arabic)—possibly in addition to (Western) European languages—and text in these languages needs to be included in the full-text search index. The Content Server software can be installed on an English-language or localized operating system (for example, the Japanese or Korean version of Microsoft Windows).

Environments like these require some modifications to the file and language encoding settings of the out-of-the-box content server. The exact modifications depend on the setup and combination of languages to be supported.

Modifications to Configuration File

If you want the content server to only handle a single non-European language in addition to English (for example, only Korean and English), you could choose to use the file and language encodings specific to that language. However, this will typically limit the character sets that the content server can work with. It is generally better to use UTF-8 and Unicode encodings as these provide the most comprehensive character set support and allow virtually any languages to be mixed without any issues.

Whichever path you choose, you need to make some modifications to the *[Install_Dir]/bin/intradoc.cfg* configuration file (where *[Install_Dir]* is the installation directory of the content server instance):

- ❖ You need to change the **Content Server file encoding** by adding the following line to the *intradoc.cfg* file:

```
FileEncoding= [value]
```

(where *[value]* is the [appropriate value for the language](#) as provided in the table on page 4-3.)

- ❖ If you are using the Verity search engine, you also need to change the **Verity locale** by adding the following line to the *intradoc.cfg* file:

```
VerityLocale= [value]
```

(where *[value]* is the [appropriate value for the language](#) as provided in the table on page 7-5.)

Save the *intradoc.cfg* file and make sure that you restart the content server and the Admin server.

Example

If, for example, you want the content server to support English and Japanese (and no other languages), you would modify the *intradoc.cfg* configuration file to include these two lines:

```
FileEncoding=SJIS  
VerityLocale=japanb [if you are using the Verity search engine]
```

If you want the content server to support English, Japanese, and other languages (European or non-European), you would modify the *intradoc.cfg* configuration file to include these two lines:

```
FileEncoding=UTF8  
VerityLocale=uni [if you are using the Verity search engine]
```



Note: In the second example, you also need to make sure the database is configured to support Unicode (see chapter 5).

Important Considerations

Please note the following considerations for Asian and other non-European environments:

- ❑ If you have installed the Content Server software on a non-Asian operating system and you want it to support Asian languages, you need to install the correct language packs for the operating system.
- ❑ If you are using Apache 2.x as the web server, you may need to modify Apache's AddDefaultCharset configuration setting. Some browsers (particularly, Mozilla-based ones) may not show an HTML page in the character set specified in the page. This is because Apache has a configuration entry which augments the HTTP Content-Type return header to specify the character set of every page that Apache returns, and some browsers will trust the header instead of the character set as specified in the page. This can be a particular problem for the Content Server help pages, which use UTF-8 as their standard internal character set encoding.

To work around this issue, use the Apache configuration entry AddDefaultCharset to make it agree with the character set as specified in the web pages. This value can be determined by looking for the value %pagecharset% in the HTML tag:

```
<meta http-equiv="Content-Type" content="text/html; charset=%pagecharset%">
```

If the website is returning pages in more than one character set, then set AddDefaultCharset to the value "Off" or use the Location or Directory directives to scope its usage. Please note that the content server may be using different character sets depending on the source of the content in the pages. Here is an example configuration for a Japanese content server that is using the native MS932 (or SJIS) character set:

```
<Location /idcm1>  
AddDefaultCharset shift_jis  
</Location>
```



```
<Location /idcm1/help>
AddDefaultCharset utf-8
</Location>
```

Note that the help pages have to be scoped separately to deliver UTF-8 encoded pages.

- ❑ After modifying the VerityLocale parameter, you need to rebuild the search index. This may be a very time-consuming process, depending on the number of content items managed by your content server instance. It is therefore recommended that you perform the index rebuild during off-peak hours of content server use (typically at night or on the weekend).
- ❑ The ‘uni’ locale is not supported by Verity VDK 4.5.1; it requires VDK 5.x or higher. If required, contact [Oracle support](#) for assistance.
- ❑ If you are running any refinery products, you also need to add the FileEncoding entry for each connection. Add the parameter to the file *[Ref_Dir]/Connections/[Connection_Name]/intradoc.cfg* (where *[Ref_Dir]* is the installation directory of Inbound Refinery and *[Connection_Name]* the name of the refinery connection). Make sure that you restart the content server instance that the refinery connection links to.
- ❑ The content ID must be supplied in single-byte characters. Multi-byte characters for the content ID are not supported by Verity. It is recommended that you autogenerate your content IDs (Administration—Admin Server—*[Instance Name]* button—General Configuration—Automatically assign a content ID on check-in).

You can also hide the content ID field on the content check-in page by adding the following parameter to *[Install_Dir]/config/config.cfg* (where *[Install_Dir]* is the installation directory of the content server instance):

```
dDocName:isHidden=1
```

- ❑ **Verity 4.5 only:** The Verity search engine does not handle Japanese hankaku (half-size) katakana and zenkaku (full-size) katakana the same. This means that not all instances of katakana text may be found in a full-text search. To ensure that all katakana text in documents is indexed and handled correctly, make sure that you use Verity VDK version 5.x or higher.



Important: In light of this, it is strongly recommended that you use Verity VDK version 5.x or higher if your content server is to handle Japanese documents.

- ❑ Full-text indexing of PDF files is not always handled correctly for multi-byte and bidirectional languages if the default Verity settings are used. To fix this, you must edit Verity’s *style.uni* to use the keyview filter for PDFs.

1. Add the entry `UseVdk4CustomStyle=true` to `[CS_Instance_Dir]/config/config.cfg`.
2. Save the file, and restart the content server.
3. Create a directory called `vdk4_custom_style` under the `[CS_Instance_Dir]/search` directory.
4. Copy all files from the `[CS_Instance_Dir]/shared/search/style/basic` directory to the `[CS_Instance_Dir]/search/vdk4_custom_style` directory.
5. Start a text editor and open `style.uni` in the `[CS_Instance_Dir]/search/vdk4_custom_style` directory.
6. Comment out the line `/format-filter = "flt_pdf"` under `type: "application/pdf"` and uncomment `/format-filter = "flt_kv"`, for example:


```
type: "application/pdf"
# /format-filter = "flt_pdf"
/format-filter = "flt_kv"
```
7. Start a text editor and open `config.cfg` in the `[CS_Instance_Dir]/config` directory.
8. Make sure this file contains the following line (add it if necessary):


```
UseVdk4CustomStyle=true
```
9. Save the changes to `config.cfg` and exit the text editor.
10. Restart the content server.
11. Save and rebuild the search index. (This may take a long time, depending on the number of content items already in the system.)



Note: Even using the “flt_kv” filter, full-text indexing of PDF files is not always indexed correctly for multi-byte characters. You can fix this by using the `TextIndexerFilter` component.

- ❑ Search word highlighting is not supported for multi-byte locales. This feature should be turned off in Content Server if your [system locale](#) is a multi-byte language (for example, Japanese or Korean).

To do this, proceed as follows:

1. **UNIX:** Start the System Properties utility by running the `System_Properties` script, which is located in the `bin` subdirectory of the Content Server installation directory.

Windows: Choose Start—Programs—Content Server—`[Instance Name]`—Tools—System Properties (make sure you have administrator rights).

2. Make sure the Options tab is opened.

3. Make sure the “Enable search keyword highlighting” check box is cleared, and click **OK**.
4. Restart the content server instance.

- ❑ If you have difficulty viewing multi-byte output with a Netscape browser, update your browser to version 6 or higher. Older versions of Netscape do not support the display of multi-byte characters reliably.
- ❑ To view multi-byte characters within applets on a non-native operating system (for example, Japanese on an English operating system), you must install Sun’s JVM 1.4 or higher and configure your browser to use this virtual machine.
- ❑ Most versions of Internet Explorer corrupt specific characters in multi-byte and bidirectional languages (for example, Korean, Hebrew, and Arabic) in URLs if users are on a native operating system and the content server file encoding is set to UTF-8 (i.e., `FileEncoding=UTF8` in `[CS_Instance_Dir]/config/config.cfg`). To work around this issue, include the following entry in `[CS_Instance_Dir]/config/config.cfg`:

```
UrlDefaultEncodingMode=full
```

This forces any non-ASCII characters in URLs to be %xx encoded, which bypasses Internet Explorer’s misinterpretation of some multi-byte and bidirectional UTF-8 characters in URLs.



Note: This also affects saved queries in Content Server as the query expressions are included in the URL when the queries are being executed.

- ❑ Users may receive workflow notifications by e-mail. If these e-mail messages are in Korean or Japanese and users view them on systems running a non-Korean or non-Japanese operating system, the subject line of the e-mail will not be displayed correctly and is unreadable. The e-mail content as such is displayed correctly.



Note: This is not an Oracle-related issue, but a limitation of the mail server software.

- ❑ If PDF bookmarks for Word documents with multi-byte characters in their headings are not displayed correctly, make sure that you enable the “Unicode signatures for PDF bookmarks” option in Inbound Refinery (local configuration setting). This setting specifies that Unicode character coding should be used when creating PDF bookmarks for Microsoft Word documents rather than ASCII. Unicode uses 16 bits, which means that non-ASCII characters can be used in the PDF bookmarks.

MIXED LANGUAGE ENVIRONMENTS

Content Server can work with a large number of different languages. If you anticipate that your content server will need to support multiple languages (especially if these languages use different encodings), it is strongly recommended that you do the following:

- ❖ Make sure the **Content Server file encoding** is set to UTF-8. You do this by adding the following line to the `[Install_Dir]/bin/intradoc.cfg` configuration file (where `[Install_Dir]` is the installation directory of the content server instance):

```
FileEncoding=UTF8
```

For details refer to [Chapter 4 \(Content Server File Encoding\)](#)



Note: If you are running any refinery products, you also need to add the `FileEncoding` entry for each connection. Add the parameter to the file `[Ref_Dir]/Connections/[Connection_Name]/intradoc.cfg` (where `[Ref_Dir]` is the installation directory of Inbound Refinery and `[Connection_Name]` the name of the refinery connection). Make sure that you restart the content server instance that the refinery connection links to.

- ❖ If you are using the Verity search engine, make sure the **Verity locale** is also set to support UTF-8. You do this by adding the following line to the `[Install_Dir]/bin/intradoc.cfg` configuration file (where `[Install_Dir]` is the installation directory of the content server instance):

```
VerityLocale=uni
```

For details refer to [Verity](#) (page 7-4).



Important: The 'uni' locale is not supported by Verity VDK 4.5.1; it requires VDK 5.x or higher. If required, contact [Oracle support](#) for assistance.

- ❖ Make sure the **database** is set to support Unicode. This is something that you specify during the installation of the Content Server software.

For details refer to [Chapter 5 \(Database Data Encoding\)](#)



DATE AND TIME FORMAT

OVERVIEW

This section covers the following topics:

- ❖ [Date and Time Format](#) (page A-1)
- ❖ [Time Zone Format](#) (page A-5)

DATE AND TIME FORMAT

One of the things defined by the [system locale](#) (and hence the [user locale](#)) is the date and time format that is used by the content server. All predefined locales come with predefined date and time formats which are optimized for the respective language (or rather, the area in which the language is spoken). For example, the default date format for the English-US locale displays dates as 8/26/06, whereas the same date with the German locale would be represented as 26.8.2006.

Date/Time Format Definition

The date/time format definition (used on the Localization tab of System Properties) might look something like this:

```
M/d/yy {h:mm[:ss] {aa} [zzz] }!mAM, PM
```

The date/time format is typically separated into three distinct areas:

- ❖ the date—for example, “M/d/yy”
- ❖ the time—for example, “{h:mm[:ss] {aa}[zzz]}”
- ❖ any additional formatting—for example, “!mAM,PM”

The following characters have a special meaning:

Character	Description
[]	The data inside braces is optional and is not displayed when a date is formatted. For example, in the above example, the seconds (although tracked) are hidden in the display, as is the time zone.
{ }	The data inside brackets is optional, but is displayed when a date is formatted.
!m	Non-pattern escape for AM/PM symbols. This escape sequence defines what “before-noon” and “after-noon” text indicators should be used for the ‘aa’ parameter in the date/time format definition (typically “AM,PM”).
!t	Non-pattern escape for the time zone name (for example, “!tAmerica/Chicago”). This escape sequence specifies the time zone that the system should consider the time to be in. The ‘!t’ value must match a time zone name defined in <i>[Install_Dir]/shared/config/resources/std_locale.htm</i> . It is typically not used for display purposes, but rather to enable the system to relate times to each other.

Administrators can change the pattern assigned to the setting to display a different date/time format. The patterns used in the date/time format come directly from Java and can use the following options:

Symbol	Meaning	Presentation	Example
G	Era designator	Text	AD
y	Year	Number	06 (yy), 2006 (yyyy) (see important note below)

Symbol	Meaning	Presentation	Example
M	Month in year	Text & Number	11 (M), Nov (MMM), November (MMMM)
d	Day in month	Number	6 (d), 06 (dd)
h	Hour in AM/PM (1~12)	Number	8 (h), 08 (hh)
H	Hour in day (0~23)	Number	8 (H), 08 (HH), 15 (H or HH)
m	Minutes in hour	Number	8 (m), 08 (mm), 15 (m or mm)
s	Second in minute	Number	8 (s), 08 (ss), 15 (s or ss)
S	Milliseconds	Number	978
E	Day in week	Text	Tue (EEE), Tuesday (EEEE)
D	Day in year	Number	189
a	AM/PM marker	Text	PM
k	Hour in day (1~24)	Number	15
K	Hour in AM/PM (0~11)	Number	0
z	Time zone	Text	PST (zz), Pacific Standard Time (zzzz)
'	Escape for text	Delimiter	(none)
'	Single quote	Literal	'



Important: If a locale uses two digits to represent the year ('yy'), then a year is interpreted to be between 1969 and 2068. In other words, '65' is interpreted as 2065, not 1965. To interpret years before 1969 correctly, you need to make sure that the locale uses four digits to represent years ('yyyy'). Please note that the default English-US locale uses two digits.

The number of symbol letters you specify also determines the format. For example, if the “zz” pattern results in “PST,” then the “zzzz” pattern generates “Pacific Standard Time.” The following table summarizes these rules:

Pattern Rule		
Presentation	Number of Symbols	Result
Text	1 – 3	Abbreviated form, if one exists—for example, “Tue”.
Text	4 or more	Full form—for example, “Tuesday”.
Number	Minimum number of digits is required.	Lower numbers are padded with leading zeros: 04 vs. 4. Years are truncated to last two digits: 04 vs. 2004.
Text & Number	1 – 2	Number form—for example, “11”.
Text & Number	3	Short text form—for example, “Nov”.
Text & Number	4 or more	Long text form—for example, “November”.

A pattern containing any invalid pattern letter results in a thrown exception during formatting or parsing.



Tech Tip: Using a `SystemDateFormat` setting in Content Server’s `config/config.cfg` file will change the system date. However, this may impact the way Content Server will serialize dates into files. Using the System Properties utility to change system locales and user locales is the recommended method and allows for implementing one or more date format views.



Tech Tip: You can also set a display date format for a locale in the System Properties utility. This extra data/time format is used for display purposes, and is used by the Idoc Script functions `formatDateDisplay`, `formatDateOnlyDisplay`, and `formatTimeOnlyDisplay`.

TIME ZONE FORMAT

All time zones are contained in the file *std_locale.htm*, which is located in the directory *[Install_Dir]/shared/config/resources* (where *[Install_Dir]* is the installation directory of the content server instance). The time zone format definition used in Content Server might look something like this:

```
STZ:-5.0,3,1,1,2.0,9,-1,1,2.0
```

Each time zone is defined by a total of nine parameters, which are separated by commas:

- ❖ The given base time zone offset to Greenwich Mean Time (GMT), in floating-point hours. A negative number means it is earlier than GMT, and a positive number it is later than GMT. For example, '-5.0' means the time zone is five hours earlier than GMT.
- ❖ The month in which daylight savings time starts (January = 0, February = 1, etc.).
- ❖ The day of the week in the month that daylight savings time starts. For example, '1' means the first Sunday, Monday, etc. of the month.
- ❖ The day of the week that daylight savings time starts. For example, '1' means Sunday; '-1' means Saturday.
- ❖ The time of the day that daylight savings time starts, in floating-point hours. For example, '2.0' means 2:00 am.
- ❖ The month in which daylight savings time ends (January = 0, February = 1, etc.).
- ❖ This is the day of the week in the month that daylight savings time ends. For example, '-1' means the last Sunday, Monday, etc. of the month.
- ❖ This is the day of the week that daylight savings time ends. For example, '1' means Sunday; '-1' means Saturday.
- ❖ This is the time of the day that daylight savings time ends, in floating-point hours. For example, '2.0' means 2:00 am.

THIRD PARTY LICENSES

OVERVIEW

This appendix includes a description of the Third Party Licenses for all the third party products included with this product.

- ❖ [Apache Software License](#) (page B-1)
- ❖ [W3C® Software Notice and License](#) (page B-2)
- ❖ [Zlib License](#) (page B-3)
- ❖ [General BSD License](#) (page B-4)
- ❖ [General MIT License](#) (page B-5)
- ❖ [Unicode License](#) (page B-5)
- ❖ [Miscellaneous Attributions](#) (page B-7)

APACHE SOFTWARE LICENSE

- * Copyright 1999-2004 The Apache Software Foundation.
- * Licensed under the Apache License, Version 2.0 (the "License");
- * you may not use this file except in compliance with the License.
- * You may obtain a copy of the License at
- * <http://www.apache.org/licenses/LICENSE-2.0>
- *

- * Unless required by applicable law or agreed to in writing, software
- * distributed under the License is distributed on an "AS IS" BASIS,
- * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- * See the License for the specific language governing permissions and
- * limitations under the License.

W3C® SOFTWARE NOTICE AND LICENSE

- * Copyright © 1994-2000 World Wide Web Consortium,
- * (Massachusetts Institute of Technology, Institut National de
- * Recherche en Informatique et en Automatique, Keio University).
- * All Rights Reserved. <http://www.w3.org/Consortium/Legal/>
- *
- * This W3C work (including software, documents, or other related items) is
- * being provided by the copyright holders under the following license. By
- * obtaining, using and/or copying this work, you (the licensee) agree that
- * you have read, understood, and will comply with the following terms and
- * conditions:
- *
- * Permission to use, copy, modify, and distribute this software and its
- * documentation, with or without modification, for any purpose and without
- * fee or royalty is hereby granted, provided that you include the following
- * on ALL copies of the software and documentation or portions thereof,
- * including modifications, that you make:
- *
- * 1. The full text of this NOTICE in a location viewable to users of the
- * redistributed or derivative work.
- *
- * 2. Any pre-existing intellectual property disclaimers, notices, or terms
- * and conditions. If none exist, a short notice of the following form
- * (hypertext is preferred, text is permitted) should be used within the
- * body of any redistributed or derivative code: "Copyright ©
- * [\$date-of-software] World Wide Web Consortium, (Massachusetts

* Institute of Technology, Institut National de Recherche en
* Informatique et en Automatique, Keio University). All Rights
* Reserved. <http://www.w3.org/Consortium/Legal/>"
*
* 3. Notice of any changes or modifications to the W3C files, including the
* date changes were made. (We recommend you provide URIs to the location
* from which the code is derived.)
*
* THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS
* MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT
* NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR
* PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE
* ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
*
* COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
* CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR
* DOCUMENTATION.
*
* The name and trademarks of copyright holders may NOT be used in advertising
* or publicity pertaining to the software without specific, written prior
* permission. Title to copyright in this software and any associated
* documentation will at all times remain with copyright holders.
*

ZLIB LICENSE

* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005

Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages

arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org

Mark Adler madler@alumni.caltech.edu

GENERAL BSD LICENSE

Copyright (c) 1998, Regents of the University of California

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

"Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

"Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

"Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GENERAL MIT LICENSE

Copyright (c) 1998, Regents of the Massachusetts Institute of Technology

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

UNICODE LICENSE

UNICODE, INC. LICENSE AGREEMENT - DATA FILES AND SOFTWARE

Unicode Data Files include all data files under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>. Unicode Software includes any source code published in the Unicode Standard or under the directories <http://www.unicode.org/Public/>, <http://www.unicode.org/reports/>, and <http://www.unicode.org/cldr/data/>.

Third Party Licenses

NOTICE TO USER: Carefully read the following legal agreement. BY DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING UNICODE INC.'S DATA FILES ("DATA FILES"), AND/OR SOFTWARE ("SOFTWARE"), YOU UNEQUIVOCALLY ACCEPT, AND AGREE TO BE BOUND BY, ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE, DO NOT DOWNLOAD, INSTALL, COPY, DISTRIBUTE OR USE THE DATA FILES OR SOFTWARE.

COPYRIGHT AND PERMISSION NOTICE

Copyright © 1991-2006 Unicode, Inc. All rights reserved. Distributed under the Terms of Use in <http://www.unicode.org/copyright.html>.

Permission is hereby granted, free of charge, to any person obtaining a copy of the Unicode data files and any associated documentation (the "Data Files") or Unicode software and any associated documentation (the "Software") to deal in the Data Files or Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Data Files or Software, and to permit persons to whom the Data Files or Software are furnished to do so, provided that (a) the above copyright notice(s) and this permission notice appear with all copies of the Data Files or Software, (b) both the above copyright notice(s) and this permission notice appear in associated documentation, and (c) there is clear notice in each modified Data File or in the Software as well as in the documentation associated with the Data File(s) or Software that the data or software has been modified.

THE DATA FILES AND SOFTWARE ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE DATA FILES OR SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in these Data Files or Software without prior written authorization of the copyright holder.

Unicode and the Unicode logo are trademarks of Unicode, Inc., and may be registered in some jurisdictions. All other trademarks and registered trademarks mentioned herein are the property of their respective owners

MISCELLANEOUS ATTRIBUTIONS

Adobe, Acrobat, and the Acrobat Logo are registered trademarks of Adobe Systems Incorporated.

FAST Instream is a trademark of Fast Search and Transfer ASA.

HP-UX is a registered trademark of Hewlett-Packard Company.

IBM, Informix, and DB2 are registered trademarks of IBM Corporation.

Jaws PDF Library is a registered trademark of Global Graphics Software Ltd.

Kofax is a registered trademark, and Ascent and Ascent Capture are trademarks of Kofax Image Products.

Linux is a registered trademark of Linus Torvalds.

Mac is a registered trademark, and Safari is a trademark of Apple Computer, Inc.

Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.

MrSID is property of LizardTech, Inc. It is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Oracle is a registered trademark of Oracle Corporation.

Portions Copyright © 1994-1997 LEAD Technologies, Inc. All rights reserved.

Portions Copyright © 1990-1998 Handmade Software, Inc. All rights reserved.

Portions Copyright © 1988, 1997 Aladdin Enterprises. All rights reserved.

Portions Copyright © 1997 Soft Horizons. All rights reserved.

Portions Copyright © 1995-1999 LizardTech, Inc. All rights reserved.

Red Hat is a registered trademark of Red Hat, Inc.

Sun is a registered trademark, and Sun ONE, Solaris, iPlanet and Java are trademarks of Sun Microsystems, Inc.

Sybase is a registered trademark of Sybase, Inc.

UNIX is a registered trademark of The Open Group.

Verity is a registered trademark of Autonomy Corporation plc



#

&#xxxx; character format, 8-2

A

AddDefaultCharset setting (Apache), 10-6
 adding a system locale, 6-4
 adding language support to operating system, 3-3
 Apache
 AddDefaultCharset, 10-6
 Asian languages
 content server environments, 10-4
 encoding compatibility, 9-2

B

batchload file, file encoding of --, 4-5
 bidirectional languages, 2-3

C

c shell, 3-1
 changing the system locale, 6-2
 changing the Verity locale, 7-7
 checking the Verity version, 7-8
 Chinese
 encoding compatibility, 9-3
 configuration variables
 XmlEncodingMode, 8-3
 content IDs
 multibyte --, 7-9, 10-7
 Content Server
 database language encoding, 9-1
 file encoding, 9-1
 language packs, 6-10
 using -- in international environments, 1-2

D

database encoding, 9-1
 native, 5-2
 Unicode, 5-2
 date format, 6-7
 definition of --, A-1
 two-digit years, A-3
 DBCS codepages, 2-2
 double-byte languages, 2-2

E

Eastern European languages, 10-3
 editing a system locale, 6-4
 enabling user locales, 6-8
 encoding
 Content Server, 9-1
 database, 9-1
 native file --, 4-2
 Verity, 7-5, 9-1, 10-2
 encoding compatibility
 Asian languages, 9-2
 European languages, 9-1
 encoding of locale, 6-7
 English-only environments, 10-1
 englishx (Verity locale), 7-5, 10-2
 European language support, 4-2
 European languages, 10-2

F

file encoding
 batchload file, 4-5
 Content Server, 9-1
 setting the Content Server --, 4-2
 user passwords, 4-5
 FileEncoding parameter, 4-2, 10-3, 10-5

H

headings in Word documents, 7-10, 10-9
highlighting of search words, 10-8

I

installation
 system locale, 6-1
international environments, 1-2
 Content Server file encoding, 9-1
 database language encoding, 9-1
 English-only, 10-1
 mixed language environments, 10-10
 pan-European languages, 10-3, 10-4
 system locale, 6-1
 user locales, 6-8
 Verity, 7-5, 10-2
 Verity encoding, 9-1
 Western European languages, 10-2
Intradoc.cfg
 FileEncoding parameter, 4-2, 10-3, 10-5
 VerityLocale parameter, 7-7, 10-3, 10-5
ISO-8859 code page, 7-5, 9-1

J

Japanese
 encoding compatibility, 9-2
 katakana, 10-7

K

katakana (in Japanese), 10-7
Korean
 encoding compatibility, 9-2
KSC5601 code page, 9-2

L

language code of locale, 6-6
language encoding
 database, 9-1
 UNIX, 3-1
 Verity, 9-1
 Windows, 3-3
language packs, 6-10
languages
 adding language support to Windows, 3-3
 Asian and other non-European environments, 10-4
 bidirectional --, 2-3

double-byte --, 2-2
English-only environments, 10-1
mixing --, 10-10
pan-European environments, 10-3
single-byte --, 2-1
Western European environments, 10-2

locale properties

date/time format, 6-7, A-1
encoding, 6-7
language code, 6-6
time zone, 6-7, A-5
Verity locale, 6-7

Locale property, 6-6

locales

changing the system locale, 6-2
changing the Verity locale, 7-7
checking the Verity version, 7-8
considerations for system --, 6-7
considerations for user --, 6-10
considerations for Verity --, 7-8
enabling user --, 6-8
resource file directory, 6-5
resource files, 6-5
setting the system locale, 6-2
system --, 6-1
user --, 6-8
Verity --, 6-7, 7-5, 10-2

localized user interface, 6-8, 6-9

localized Verity search operators, 7-8

M

Microsoft Windows
 adding language support, 3-3
 language encoding, 3-3
mixed language environments, 10-10
multi-byte languages
 full-text indexing of PDF files, 10-7
 search word highlighting, 10-8
multibyte content IDs, 7-9, 10-7

N

native database encoding, 5-2
native operating system encoding, 4-2
non-European languages, 10-4
nvarchar data type, 5-2

P

pan-European language environments, 10-3
passwords of users, 4-5

PDF bookmarks, 7-10, 10-9
 PDF file indexing
 bidirectional languages, 10-7
 multi-byte languages, 10-7

R

resource file directory, 6-5
 resource files, 6-5

S

SBCS codepages, 2-1
 search index locale, 7-5, 10-2
 search operators (Verity), 7-8
 search word highlighting, 10-8, 10-8
 setting the system locale, 6-2
 single-byte languages, 2-1
 SJIS code page, 9-2
 Solaris
 determining current locale, 3-2
 setting locale, 3-2
 system locale, 6-1
 adding a new --, 6-4
 available --s, 6-3
 changing the --, 6-2
 considerations, 6-7
 date format, 6-7, A-1
 editing an existing --, 6-4
 language code, 6-6
 resource file directory, 6-5
 resource files, 6-5
 setting the --, 6-2
 time format, 6-7, A-1
 time zone, 6-7, A-5

T

time format, 6-7
 definition of --, A-1
 time zone, 6-7
 definition of -- format, A-5
 two-digit years, A-3

U

Unicode database encoding, 5-2
 UNIX
 language encoding, 3-1
 user locales, 6-8
 available --, 6-9
 considerations, 6-10
 enabling --, 6-8
 user password and file encoding, 4-5
 UTF-8, 4-2

V

varchar data type, 5-2
 Verity, 10-8
 checking -- version, 7-8
 katakana, 10-7
 search operators, 7-8
 Verity language encoding, 9-1
 Verity locale, 6-7, 7-5, 9-1, 10-2
 changing the --, 7-7
 considerations, 7-8
 default, 7-5
 VerityLocale parameter, 7-7, 10-3, 10-5

W

Western European languages, 10-2
 Word documents, headings in --, 7-10, 10-9

X

x (at end of Verity locale names), 7-5
 XmlEncodingMode (configuration variable), 8-3

Y

year as two digits, A-3

