ORACLE
INSURANCE

Oracle® Documaker

Fonts Reference

version 12.1

Part number: E22582-01

June 2012

THIRD PARTY SOFTWARE NOTICES

This product includes software developed by Apache Software Foundation (http://www.apache.org/).

THIS SOFTWARE IS PROVIDED "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 AUTHOR 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.

This product includes software distributed via the Berkeley Software Distribution (BSD) and licensed for binary distribution under the Generic BSD license.

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.

This product includes software developed by the JDOM Project (http://www.jdom.org/).

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED 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 JDOM AUTHORS OR THE PROJECT 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.

This product includes software developed by the Massachusetts Institute of Technology (MIT).

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.

This product includes software developed by 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.

This software is based in part on the work of the Independent JPEG Group (http://www.ijg.org/).

This product includes software developed by the Dojo Foundation (http://dojotoolkit.org).

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.

Copyright (c) 2005-2009, The Dojo Foundation. All rights reserved.

This product includes software developed by W3C.

Copyright © 2009 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 product includes software developed by Mathew R. Miller (http://www.bluecreststudios.com).

Copyright (c) 1999-2002 ComputerSmarts. All rights reserved.

This product includes software developed by Shaun Wilde and distributed via Code Project Open License (http://www.codeproject.com).

THIS WORK IS PROVIDED "AS IS", "WHERE IS" AND "AS AVAILABLE", WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OR GUARANTEES. YOU, THE USER, ASSUME ALL RISK IN ITS USE, INCLUDING COPYRIGHT INFRINGEMENT, PATENT INFRINGEMENT, SUITABILITY, ETC. AUTHOR EXPRESSLY DISCLAIMS ALL EXPRESS, IMPLIED OR STATUTORY WARRANTIES OR CONDITIONS, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, MERCHANTABLE QUALITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT, OR THAT THE WORK (OR ANY PORTION THEREOF) IS CORRECT, USEFUL, BUG-FREE OR FREE OF VIRUSES. YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

This product includes software developed by Chris Maunder and distributed via Code Project Open License (http://www.codeproject.com).

THIS WORK IS PROVIDED "AS IS", "WHERE IS" AND "AS AVAILABLE", WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OR GUARANTEES. YOU, THE USER, ASSUME ALL RISK IN ITS USE, INCLUDING COPYRIGHT INFRINGEMENT, PATENT INFRINGEMENT, SUITABILITY, ETC. AUTHOR EXPRESSLY DISCLAIMS ALL EXPRESS, IMPLIED OR STATUTORY WARRANTIES OR CONDITIONS, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, MERCHANTABLE QUALITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT, OR THAT THE WORK (OR ANY PORTION THEREOF) IS CORRECT, USEFUL, BUG-FREE OR FREE OF VIRUSES. YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

This product includes software developed by PJ Arends and distributed via Code Project Open License (http://www.codeproject.com).

THIS WORK IS PROVIDED "AS IS", "WHERE IS" AND "AS AVAILABLE", WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OR GUARANTEES. YOU, THE USER, ASSUME ALL RISK IN ITS USE, INCLUDING COPYRIGHT INFRINGEMENT, PATENT INFRINGEMENT, SUITABILITY, ETC. AUTHOR EXPRESSLY DISCLAIMS ALL EXPRESS, IMPLIED OR STATUTORY WARRANTIES OR CONDITIONS, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, MERCHANTABLE QUALITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT, OR THAT THE WORK (OR ANY PORTION THEREOF) IS CORRECT, USEFUL, BUG-FREE OR FREE OF VIRUSES. YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

This product includes software developed by Erwin Tratar. This source code and all accompanying material is copyright (c) 1998-1999 Erwin Tratar. All rights reserved.

THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT EXPRESS OR IMPLIED WARRANTY. USE IT AT YOUR OWN RISK! THE AUTHOR ACCEPTS NO LIABILITY FOR ANY DAMAGE/LOSS OF BUSINESS THAT THIS PRODUCT MAY CAUSE.

This product includes software developed by Sam Leffler of Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE

Copyright (c) 1988-1997 Sam Leffler
Copyright (c) 1991-1997 Silicon Graphics, Inc.

This product includes software developed by Guy Eric Schalnat, Andreas Dilger, Glenn Randers-Pehrson (current maintainer), and others. (http://www.libpng.org)

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

This product includes software components distributed by the Cryptix Foundation.

THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED 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 CRYPTIX FOUNDATION LIMITED 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

Copyright © 1995-2005 The Cryptix Foundation Limited. All rights reserved.

This product includes software components distributed by Sun Microsystems.

This software is provided "AS IS," without a warranty of any kind. ALLEXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANYIMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL NOT BELIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Copyright (c) 1998 Sun Microsystems, Inc. All Rights Reserved.

This product includes software components distributed by Dennis M. Sosnoski.

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.

Copyright © 2003-2007 Dennis M. Sosnoski. All Rights Reserved

It also includes materials licensed under Apache 1.1 and the following XPP3 license

THIS SOFTWARE IS PROVIDED "AS IS'" AND ANY EXPRESSED 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.

Copyright © 2002 Extreme! Lab, Indiana University. All Rights Reserved

---

This product includes software components distributed by CodeProject. This software contains material that is © 1994-2005 The Ultimate Toolbox, all rights reserved.

This product includes software components distributed by Geir Landro.

Copyright © 2001-2003 Geir Landro (drop@destroydrop.com) JavaScript Tree - www.destroydrop.com/hjavascripts/tree/version 0.96

This product includes software components distributed by the Hypersonic SQL Group.

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

This product includes software components distributed by the International Business Machines Corporation and others.

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.

This product includes software components distributed by the University of Coimbra.

University of Coimbra distributes this software in the hope that it will be useful but DISCLAIMS ALL WARRANTIES WITH REGARD TO IT, including all implied warranties of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. In no event shall University of Coimbra be liable for 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 this software.

This product includes software components distributed by Steve Souza.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR 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.

This product includes software developed by the OpenSymphony Group (http://www.opensymphony.com/.)"

CONTENTS

Preface

Oracle Documaker is a powerful, adaptive enterprise document automation platform used worldwide to acquire, create, manage, and present structured, on-demand, and interactive customer communications. It is designed to put power in the hands of business users, giving them the flexibility to create interactive, dynamic documents on demand.

Spanning the entire business life cycle, Oracle Documaker helps you manage customer communications across the entire enterprise—including document production, correspondence, and cross-selling campaigns—across all locations and lines of business. The industry-leading platform offers a cost-effective way to address the design, production, and multichannel distribution of a broad spectrum of customer-facing documents. With robust functionality and cutting-edge technical capabilities, it maximizes efficiencies, ensures compliance, and enhances customer service.

Oracle Documaker is based on open standards and integrates easily into today's service-oriented architecture environments. It integrates with any type of system across the enterprise. It can even be integrated with your self-service Web portal so stakeholders can get immediate access to up-to-date information. Oracle Documaker provides the agility and flexibility you need to roll out new products quickly and remain competitive.

Oracle offers proven tools and migration methods, along with experienced, highly trained technical personnel to ease conversions while maintaining the intelligence of your data. Leveraging Oracle Documaker as a single system can dramatically reduce costs. One insurance customer recouped the full cost of an Oracle Documaker implementation within nine months.

Business users can easily author content in Oracle Documaker Studio using Microsoft Word through a plug-in that leverages the power of Documaker Studio in the background. For even more capability, Oracle Documaker's intuitive, easy-to-use design tool, Documaker Studio, empowers business users to create powerful, persuasive content minimizing their reliance on IT, so you can produce dynamic, *intelligent* transactional documents that transmit data and content.

Audience

This document is intended to help you understand how Documaker uses fonts. It will be useful to those who design forms, maintain font cross-reference files, and who want to understand how fonts are used in the Documaker system.

Documentation Accessibility

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, refer to the following Oracle resources:

- Documaker Installation Guide
- Documaker Administration Guide
- Documaker Studio User Guide
- Rules Reference
- DAL Reference
- Printers Reference
- Unicode Reference
- Utilities Reference
- Documaker Desktop Installation Guide
- Documaker Workstation Administration Guide

Conventions

The following text conventions are used in this document:

Convention	Description
bold	Indicates information you enter.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands, URLs, code in examples, and text that appears on the screen.

Chapter 1

Understanding Fonts

A font is a collection of letters, symbols, and numbers which share a particular design. This reference guide provides general information on font concepts and types.

A font is a collection of letters, symbols, and numbers which share a particular design. Studio provides font management tools which let you organize sets of fonts for section creation and printing needs.

This guide provides general information on font concepts and types and includes the following topics

Note The Documaker system also includes several utilities you can use to work with fonts. These utilities are mentioned where appropriate throughout this chapter and are discussed in detail in the Utilities Reference.

General Font Concepts

Font Terminology

The following is a glossary of some common typographic terms you may encounter when working with fonts.

Typography is the art and technique of selecting and arranging type styles, point sizes, line lengths, line spacing, character spacing, and word spacing for typeset applications.

A *typeface* is a unique design of upper- and lower-case characters, numerals, and special symbols. Times-Roman, Arial-Italic, Courier-Bold are examples of typefaces.

A *font* is the implementation, for a specific device, of one typeface. A font contains a group of characters (letters, numbers, punctuation, and so on) which have a specific form and size. As you can see below, a Courier font is one which is designed to look like it was produced by a typewriter.

```
Courier fonts look like text produced by a typewriter.
```

A *font family* is family of related font typefaces. Times-Roman, Times-Bold, Times-Italic, and Times-BoldItalic are typefaces which belong to the Times font family.

Font size refers to the vertical point size of a font, where a point is about 1/72 of an inch.

There are several other terms used to describe the characteristics of a font, including:

- Ascender
- Baseline
- Descender

The *ascender* is the portion of a lowercase character that extends above its main body, as in the vertical stem of the character *b*.

The *baseline* is an imaginary line upon which the characters in a line of type rest.

The *descender* is the portion of a lowercase character that extends below the baseline, as in *y* or *g*.

Kerning is the process of decreasing space between two characters for improved readability, such as tucking a lowercase *o* under an uppercase *T*. A variation of kerning, called *tracking*, involves decreasing the amount of space between all characters by a specified percentage.

Leading is the amount of vertical space between lines of text. Leading (pronounced *ledding*) is measured from baseline to baseline. On old hot-type printing presses, this was done by inserting strips of lead between the cast type.

Fonts are measured in *points*. A point is a typographical unit of measure which equals about 1/72 of an inch. For example, this is a 16 point font while the rest of the line uses a 10 point font.

A *pica* is another typographical unit of measurement equal to 12 points. There are about 6 picas in one inch.

A *twip* is yet another typographical unit of measurement equal to 1/20th of a point. There are 1440 twips to one inch, 567 twips to one centimeter.

Pitch refers to the amount of horizontal space used for each character of fixed-width fonts. This is often specified in characters-per-inch (CPI). Typically, 10-pitch equals 12 point, 12-pitch equals 10 point, and 15-pitch equals 8 point type, but some fonts use other equivalencies.

Sans serif means without serifs and refers to a character (or typeface) that lacks serifs, such as Arial or Helvetica.

A *serif* is an ornamental aspect of a character. A serif typeface is one whose characters contain serifs (such as Times Roman or Courier).

Spacing can either be fixed or proportional. In a fixed font, such as Courier, every character occupies the same amount of space. In a proportional font, such as Arial or Times, characters have different widths.

Stroke weight refers to the heaviness of the stroke for a specific font. This is usually indicated in font names by including words such as Light, Regular, Book, Demi, Heavy, Black, and Extra Bold.

The *style* of a font is whether it is plain, bold, or italic.

National Language Terminology

Here are some additional terms you may encounter when working with fonts and supporting international languages.

National character handling is dependent on both the language used, and on the country. In many cases, the language is used only in one country (such as Japanese in Japan). In other cases, there is a national variant of the language (such as Canadian French).

Term	Definition
AFM	AFM is an extension used with Adobe® PostScript© font files. It stands for Adobe Font Metrics. AFM files are text files that describe a PostScript font.
ANSI	ANSI is an acronym for the American National Standards Institute. The Windows ANSI character set is based on code page ISO 8859-x plus additional characters based on an ANSI draft standard.
ASCII	ASCII is an acronym for the American Standard Code for Information Interchange. ASCII is a 7-bit code that is a US national variant of ISO 646.
Bi-directional (BIDI) languages	Bi-directional (BIDI) languages or Extended SBCS languages are languages which display text in a right-to-left manner and numbers in a left-to-right manner. Hebrew and Arabic are BIDI languages.
Byte Order Mark (BOM)	A Byte Order Mark (BOM) is a special Unicode character (0xFEFF) that is placed at the beginning of Unicode text files to indicate that the text is in Unicode format. For UTF-8 encoded files, the Byte Order Mark character is encoded as the following three bytes: 0xEF, 0xBB, 0xBF. The BOM will not display in text editors that support UTF-8. For text editors that do not support UTF-8 encoding, the BOM would appear as shown here: ï»¿
Character set	A character set defines which characters must be supported for a specific language.
Code page	A table that describes a character set for a particular speaking language. It is used by the operating system and applications to display and print a language properly. The code page defines 256 characters based on the 256 possible combinations in a single byte. Each character is mapped to a unique hexadecimal number, called a *code point*. There are three families of code pages: EBCDIC, ASCII, and ISO. For most code pages, the first 128 characters conform to the ASCII standard.
Code point	In character encoding terminology, a code point is the unique hexadecimal number that maps to a character in a code page. For example, ASCII comprises 128 code points in the range 0x00 to 0x7F, Extended ASCII comprises 256 code points in the range 0x00 to 0xFF, and Unicode comprises 1,114,112 code points in the range 0x00 to 0x10FFFF.
Double-byte character sets (DBCS)	Double-byte character sets (DBCS) are character sets which contain so many characters that they require two bytes to define the valid code point range. Languages which require a DBCS are Japanese (Kanji), Korean, and Chinese (both Traditional and Simplified). For example, the Kanji character set uses approximately 6,700 characters out of a total of 65,000 valid code points provided by a DBCS code page.
Enabled	Enabled is a term used to indicate an application that has been altered to handle input, display, and editing of double byte languages (such as Japanese) and bi-directional languages (such as Arabic).

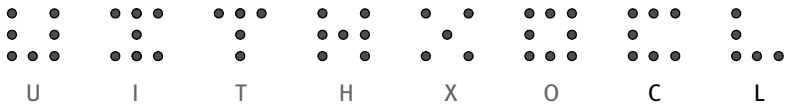
Term	Definition
Multiple Byte Character Sets (MBCS)	Multiple byte character sets (MBCS) use both single and double byte code points. This is also referred as a combined code page. For example, the combined Japanese code page 932 consists of a SBCS code page 897 and a DBCS code page 301. These code pages use the Shift JIS encoding defined by the Japanese Industry Standard Association, and contains Kanji, Hiragana, and Katakana characters.
Program Integrated Information (PII)	Program Integrated Information (PII) includes all text in messages, menus, and reports which is displayed to the user. To provide national language support, all PII text must be isolated for easy translation.
Single-byte Character Sets (SBCS	Single-byte character sets (SBCS) are character sets which can be defined using a single byte code point (code points range from 0 to 255). Most languages can be defined using an SBCS.
Translated	Translated is a term used to indicate an application which has been enabled and has had its Program Integrated Information translated into the national language. A translated application must also support various country settings, such as time, date, currency, and sorting.
Unicode	Unicode is a character coding system designed to support the interchange, processing, and display of the written texts of the diverse languages of the modern world. These characters cover the principal written languages of the Americas, Europe, the Middle East, Africa, India, Asia, and Pacifica. Support for Unicode is growing among operating systems, such as Windows, and programming languages, such as Java. For specific information on how to use Unicode, see the *Unicode Reference.*
UTF-8	UTF-8 is the variable-length byte-oriented encoding form of Unicode. UTF-8 uses only one byte for the ASCII characters (code points below 128). For code points 128 and higher, either two, three, or four bytes are used to represent each character.

How Characters are Represented

Fonts can use different methods of internally representing characters. Two categories of representing characters in fonts are known as bitmap fonts and scalable fonts.

Bitmap Fonts

Bitmap fonts describe each character as a pattern of black dots. Bitmap fonts were originally used for printer and screen devices because these devices were only capable of drawing dots. Below is crude representation of how the certain letters could be drawn as a series of dots in a 3x3 grid.



Essentially, this is what happens when a character is drawn to the screen or printed on paper. Fortunately, screen and printer fonts use a whole lot more dots per inch so that the distance between the dots becomes nearly invisible to the naked eye. By the way, this is also the reason why printed text looks better that text on the screen. Printed text often uses 300 or 600 dots per inch while your screen's resolution might be 96 dots per inch.

A different font file is required for each point size and different font files are required for different device resolutions (VGA vs. Super-VGA monitors, 300 dpi vs. 600 dpi printers).

Bitmap fonts are used primarily by printers. Bitmap fonts used by printers cannot be used for displaying text on screens because there are different internal formats and different resolutions. Printers which use bitmap fonts include HP® laser printers, IBM® AFP printers, and Xerox® Metacode printers

Scalable Fonts

A scalable font can be scaled to any size needed. Characters of scalable fonts are internally represented as outlines (a series of straight lines and curves). These outlines can be scaled to allow characters to be rendered at different resolutions and point sizes. For example, the letter *O* may be represented as outer and inner circular lines whose interior is filled.

Two types of scalable fonts are TrueType and PostScript fonts.

TrueType

TrueType was designed and developed by Apple Computer and Microsoft for use on the Macintosh computer and PCs running Microsoft Windows. TrueType provides a number of advantages over bitmap fonts. TrueType is WYSIWYG (what you see is what you get). The same font can be used with printers and video displays. Typically, TrueType font files have a file extension of TTF.

PostScript

PostScript fonts were designed and developed by Adobe Systems Incorporated. PostScript fonts are a special implementation of a PostScript language program. PostScript fonts are scalable fonts. PostScript fonts describe each character as a series of straight-line and curved-line segments. These segments (also known as an outline) along with a flexible coordinate structure allow PostScript fonts to be scaled easily and used on different devices (video monitors and printers). PostScript printers support the PostScript language and fonts. There are several types of PostScript fonts:

- PostScript Type 1

  When someone refers to a PostScript font, this is the type of font most often referred to. Typically, Type 1 font files have a file extension of PFB.

- PostScript Type 3

  A Type 3 font is one whose behavior is determined entirely by the PostScript language procedures built into the font. These fonts are typically larger files than Type 1 fonts and do not take advantage of special algorithms built into the PostScript interpreter for rendering characters. This usually results in inferior output at small sizes and low resolution.

- PostScript Type 0

  A Type 0 (zero) font is a composite font program that can contain several thousand characters, accessed by multibyte codes. They can be used for non-Roman scripts, such as Japanese kanji.

- PostScript Multiple Master

  Multiple master font programs are an extension of the Type 1 font format. Multiple master font programs contain a wide variety of typeface variations, such as multiple weights, character widths, and so on.

How Computers and Printers Use Fonts

What happens to make the letter *A* show up on the screen or print on a printer?

When you press the letter *A* on the keyboard, the keyboard sends a number to computer. On a PC, this number is usually 65 for the letter *A*. The computer uses this number to produce the letter *A*. For simplicity, let's assume you have a bitmap screen font.

As stated before, bitmap fonts describe each character as a pattern of black dots. Let's assume these patterns are stored in the font as a series of slots where slot 0 is followed by slot 1 which is followed by slot 2, and so on. For the number 65 (letter *A*), the computer simply draws the pattern of dots stored in slot 65. When the bitmap is drawn on the screen, you see what looks like the letter *A*.

If you print the letter *A* with a bitmap font, the concept is essentially the same. The printer receives the number 65 and prints the series of dots stored in slot 65 of the printer font.

The numbers which the computer uses to represent characters are called *code points*.

USING CODE PAGES

A code page is a table which defines the mapping in a computer of each of these characters to a unique hexadecimal number, called a *code point*. There are three families of code pages: EBCDIC, ASCII, and ISO.

A code page is a table that defines how the characters in a language or group of languages are encoded. A specific value is given to each character in the code page. For example, in code page 850 the letter *ñ* (lowercase) is encoded as hex A4 (decimal 164), and the letter *Ñ* (uppercase) is encoded as hex A5 (decimal 165). Of particular interest are these code pages:

- Code page 850

  Code page 850 is also called the Latin-1, multilingual code page. This code page supports the alphabetic characters of the Latin-1-based languages.

- Code page 437

  Code page 437 is the standard personal computer code page. The lower 128 characters are based on the 7-bit ASCII code. The upper 128 characters contain characters from several European languages (including part of the Greek alphabet) and various graphic characters. However, some of the accented characters, such as those used in the Nordic countries, are not represented. The missing characters are available in other code pages (code page 850 will usually contain the desired characters). It contains characters required by 13 languages used in approximately 40 countries.

- Code page 1004

  Code page 1004 is the equivalent of the Windows ANSI code page 1252. It contains more international characters than the multilingual code page 850. This character set contains all characters necessary to type all major (West) European languages. This encoding is also the preferred encoding on the Internet.

ISO 8859-x character sets use code points 128 through 255 to represent national characters, while the characters in the 32 to 127 range are those used in the US-ASCII (ISO 646) character set. Thus, ASCII text is a proper subset of all ISO 8859-X character sets.

The code points 128 through 159 are typically used as extended control characters, and are not used for encoding characters. These characters are not currently used to specify anything. This character set is also used by AmigaDOS, Windows, VMS (DEC MCS is practically equivalent to ISO 8859-1) and (practically all) UNIX implementations. MS-DOS normally uses a different character set and is not compatible with this character set.

ASCII Code Pages

ASCII is an acronym for the American Standard Code for Information Interchange. ASCII code pages are used on the PC platform. Code points below 32 for ASCII code pages are considered control characters for internal uses. These code points are usually not displayable characters. Code points from 32 to 127 are usually the same in ASCII code pages and are used for English letters, numbers, and punctuation.

Where ASCII code pages differ is in the characters assigned to code points 128-255. Code points 128-255 are used for international characters, math symbols, and so on. The characters for these code points vary in other code pages.

The characters used in code points below 128 use the English letters, numbers, and punctuation commonly found in ASCII code pages. The upper 128 code points are used for characters from several European languages (including part of the Greek alphabet) and various graphic characters. However, some of the accented characters, such as those used in the Nordic countries, are not represented.

Code page 437 is known as the standard personal computer code page. These characters were originally used in the original IBM PC. This code page is still used today in U.S. English versions of DOS and Windows. The primary code page used for these platforms is also known as the OEM code page.

Code page 850 is also called the *multilingual code page*. This code page supports many of the characters of the Latin-based alphabet and is shown below. To determine the code point associated with a character, use the numbers in the first row and column in the following table. For example, the letter *A* has a code point of 65 (64 + 1) and the space character has a code point of 32 (32 + 0).

Code Page 850

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
16																
32		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
128	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
144	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
160	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
176						Á	Â	À	©					¢	¥	
192							ã	Ã								¤
208	ð	Ð	Ê	Ë	È		Í	Î	Ï					¦	Ì	
224	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
240		±		¾	¶	§	÷	¸	°	¨	•	¹	²	³		

There are many more ASCII code pages which are targeted for a specific country and or language. For example, code page 863 is used for Canadian French.

Code page 1004 is the IBM equivalent of the Windows ANSI code page 1252. It contains more international characters than the multilingual code page 850. It contains characters required by 13 languages used in approximately 40 countries. Windows uses the ANSI code page to support most of the languages used in the Western Hemisphere and Western Europe. Keystrokes are translated by Windows from the primary (OEM) code page into the ANSI code page.

The following page shows the Windows ANSI code page 1252. To determine the code point associated with a character, use the numbers in the first row and column in the following table. For example, the letter *A* has a code point of 65 (64 + 1) and the space character has a code point of 32 (32 + 0).

Code Page 1004 (Windows Code Page 1252)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
16																
32		!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
128	€		,	*ƒ*	„	…	†	ç	ˆ	‰	Š	‹	Œ		Ž	
144		‘	’	“	”	•	–	—	˜	™	š	›	œ		ž	Ÿ
160		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
176	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
192	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
208	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
224	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
240	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

EBCDIC Code Pages

EBCDIC is an acronym for the Extended Binary Coded Decimal Interchange Code. EBCDIC code pages are used on mainframe (z/OS) and mini computers (AS400). There are many EBCDIC code pages. EBCDIC code pages usually share the same code points for English letters, numbers, and punctuation characters. However, EBCDIC code pages use different code points than ASCII code pages for the same English letters, numbers, and punctuation characters. Code points below 64 for EBCDIC code pages are considered control characters for internal uses. These code points are usually not displayable characters.

Code page 37 is an EBCDIC code page used on many z/OS and AS400 systems. Although the code points are completely different, code page 37 shares most of the same characters as code page 1004 (ANSI). The characters associated with code points 128-159 in the ANSI code page are not defined in code page 37.

Note The system uses some undefined code points (below 64) in code page 37 to try represent these characters. For maximum portability, avoid using code points 128-159 of the ANSI code page when composing forms.

The following page shows a table of code page 37. To determine the code point associated with a character, use the numbers in the first row and column in the following table. For example, the letter *A* has a code point of 193 (192 + 1) and the space character has a code point of 64 (64 + 0).

Code Page 37

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
16																
32																
48																
64			â	ä	à	á	ã	å	ç	ñ	¢	.	<	(	+	|
80	&	é	ê	ë	è	í	î	ï	ì	ß	!	$	*	)	;	¬
96	-	/	Â	Ä	À	Á	Ã	Å	Ç	Ñ	¦	,	%	_	>	?
112	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	`	:	#	@	'	=	“
128	Ø	a	b	c	d	e	f	g	h	I	«	»	ð	ý	þ	±
144	°	j	k	l	m	n	o	p	q	r	ª	º	æ	¸	Æ	¤
160	µ	~	s	t	u	v	w	x	y	z	¡	¿	Ð	Ý	Þ	®
176	^	£	¥	•	©	§	¶	¼	½	¾	[	]	¯	¨	´	×
192	{	A	B	C	D	E	F	G	H	I	-	ô	ö	ò	ó	õ
208	}	J	K	L	M	N	O	P	Q	R	¹	û	ü	ù	ú	ÿ
224	\	÷	S	T	U	V	W	X	Y	Z	²	Ô	Ö	Ò	Ó	Õ
240	0	1	2	3	4	5	6	7	8	9	³	Û	Ü	Ù	Ú	

Character Sets

You may have noticed that the largest code point shown in the earlier code page tables is 255 (240 + 15). The reason for this is that 255 is the largest value which can fit into a byte of memory. Code pages like this are said to have a single byte character set (SBCS). Some Asian languages, like Japanese and Chinese, contain so many characters that they must be represented by a double byte character set (DBCS) or a multiple byte character set (MBCS).

Code Page Names

One confusing thing about code pages is that different organizations have different names for the same code pages. IBM, Microsoft, and the International Organization for Standardization (ISO) all use different names for essentially the same code page. You may hear a code page referred to by its IBM, Microsoft, or ISO name. For example, the ANSI code page is the same as IBM code page 1004, Microsoft code page 1252, and ISO code page 8859-1.

The following table shows a list of commonly used code pages. For more information, see these books:

- Developing International Software, Second Edition - Microsoft Press
- National Language Design Guide Volume 2 - IBM

		Code Pages		
Language	**Country**	**Windows**	**OEM**	**z/OS**
U.S. English	USA	1252 (ANSI)	437, 850	037
Canadian English	Canada	1252 (ANSI)	437, 850	037
Western Hemisphere and Western Europe SBCS Code Pages				
U.K. English	UK	1252 (ANSI)	850, 437	?
Brazilian Portuguese	Brazil	1252 (ANSI)	850, 437	?
Canadian French	Canada	1252 (ANSI)	850, 863	?
Danish	Denmark	1252 (ANSI)	850	?
Dutch	Netherlands	1252 (ANSI)	850, 437	?
Finnish	Finland	1252 (ANSI)	850, 437	?
French	France	1252 (ANSI)	850, 437	?
German	Germany	1252 (ANSI)	850, 437	?
Italian	Italy	1252 (ANSI)	850, 437	?
Norwegian	Norway	1252 (ANSI)	850	?
Portuguese	Portugal	1252 (ANSI)	850, 860	?
Spanish	Spain	1252 (ANSI)	850, 437	?
Swedish	Sweden	1252 (ANSI)	850, 437	?
Welsh	Wales	1252 (ANSI)	850	?
Eastern Europe SBCS Code Pages				

		Code Pages		
Language	**Country**	**Windows**	**OEM**	**z/OS**
Russian	Russia	1251 (Cyrillic)	866, 850	?
Bosnian	Bosnia	?	852, 850	?
Bulgarian	Bulgaria	1250 (Eastern Europe)	?	?
Croatian	Croatia	1250 (Eastern Europe)	852, 850	?
Czech	Czech	1250 (E.E.)	852, 850	?
Estonian	Estonia	1257 (Baltic)	?	?
Greek	Greece	1253 (Greek)	?	?
Hungarian	Hungary	1250 (E.E.)	852, 850	?
Latvian	Latvia	1257 (Baltic)	?	?
Lithuanian	Lithuania	1257 (Baltic)	?	?
Polish	Poland	1250 (E.E.)	852, 850	?
Romanian	Romania	1250 (E.E.)	852, 850	?
Serbian-Latin	Serbia	1250 (E.E.)	852, 850	?
Slovak	Slovak	1250 (E.E.)	852, 850	?
Slovenian	Slovenia	1250 (E.E.)	852, 850	?
Turkish	Turkey	1254 (Turkish)	857, 850	?
Extended SBCS Code Pages				
Arabic	Arabic speaking	1256 (Arabic)	864, 850, 437	?
Hebrew	Israel	1255 (Hebrew)	862, 850, 437	?
Thai	Thailand	874	874, 437	?
Asian DBCS Code Pages				
Japanese	Japan	932	932, 942, 437, 850	?
Korean	Korea	949	949, 850, 437	?
Simplified Chinese	PRC, Singapore	936	1381, 437, 850	?
Traditional Chinese	Taiwan, Hong Kong	950	938, 948, 437, 850 950, 437, 850	? ?

TYPES OF FONTS

Studio uses screen and printer fonts for displaying and printing text on forms. The Family field in the FXR file contains the name of the screen font to use for displaying text under Windows.

The Font File fields in the FXR file contain the names of the printer fonts to use when printing text. The FXR file provides attributes of the fonts and cross references the various font file names and parameters for different printers. The FXR file does not contain any printer or screen fonts, only information about printer and screen fonts. FXR files are referred to in this section but are discussed in detail in the *Using Font Cross-Reference Files* on page 24.

USING SCREEN FONTS

Font Substitution in Windows

If the system cannot find a matching screen font using the information in the FXR file, it will attempt to substitute a different Windows font. For Windows, the system will automatically try to substitute the following fonts for these missing fonts:

If this font is missing...	The system will substitute this font...
Courier	Courier New
Helv	Arial
Letter Gothic	Courier New
MICR	Courier New (fixed pitch) or Arial (proportional)
OCR A	Courier New (fixed pitch) or Arial (proportional)
OCR B	Courier New (fixed pitch) or Arial (proportional)
Times	Times New Roman
Times Roman	Times New Roman
Tms Rms	Times New Roman
Univers	Arial
Thorndatle AMT	Times New Roman
Andale DouSpace WT	Courier New
Cumberland AMT	Courier New

Use the WINDOW32SUBS INI control group to define substitute font names. Here is an example, which shows the default settings:

```
< Window32Subs >
   Univers              = Arial
   Helv                 = Arial
   Letter Gothic        = Courier New
   Courier              = Courier New
   Tms Rms              = Times New Roman
   Times Roman          = Times New Roman
   Times                = Times New Roman
   Thorndale AMT        = Times New Roman
```

```
Andale DuoSpace WT   = Courier New
Cumberland AMT       = Courier New
```

In this example, the system substitutes the native Windows 32-bit font, Times New Roman, if the Times family font is not found. Likewise, it substitutes Courier New for Letter Gothic and Arial for Univers. If you do not have a font installed which matches the original or substituted fonts, a default font will be used instead (usually Courier).

Installing Screen Fonts in Windows

To avoid these font substitutions, you can install fonts into Windows using the Fonts folder (usually located in the Control Panel). After opening the Fonts folder, select the File, Install New Font option. The Add Fonts window appears and asks for the drive and directory in which the new TrueType font files are located. When you finish selecting the fonts you want to install, click Ok to install them.

For the system to correctly match the fonts installed under Windows, the family and face name must be spelled exactly the same as they appear on the Names tab of the Properties window for the font. Use FXR settings for FAP height, FAP width, and so on, to customize the display of a font.

Using Printer Fonts

The system supports printer fonts for AFP, Xerox Metacode, OpenType, PCL, and PostScript printers. Here is some background information you should know about each of these print platforms.

AFP

AFP fonts are designed solely for IBM's AFP printers. In AFP terminology, a font is described by these components:

Coded fonts

A coded font file contains references to specific character set and specific code page. Coded font files always begin with the letter *X*, such as *X0DATIN8*.

Code pages

In IBM AFP terminology, a code page file maps code points to an AFP character name in a character set file. Code page files always begin with the letter *T*, such as *T1DOC037*.

Character sets

A character set file contains the bitmap image of each character in the character set. Character set files always begin with the letter *C,* such as *C0FATIN8.240* or *C0FATIN8.300*. The character set file name extension (240 or 300) indicates whether the bitmap images are drawn at 240 or 300 dots per inch. Each character is given a eight letter AFP character name. For example, the letter *A* has an AFP character name of LA020000.

Metacode

Metacode fonts are designed solely for Xerox Metacode printers. Metacode fonts are bitmap fonts. Typically, Metacode font files have a file extension of FNT, such as FXTIN8.FNT. Characters are accessed by code points.

OpenType

OpenType® is a format for scalable computer fonts which was built on its predecessor, TrueType. An OpenType font can be in one of these formats:

- TrueType format (which usually have a .TTF file extension)
- Compact Font Format (which usually have an .OTF file extension).

The Documaker PDF Print Driver and Documaker Studio support OpenType fonts using the TrueType format. You must acquire OpenType fonts with the proper licensing for your intended use. For more information, see this web site:

http://www.microsoft.com/typography/RedistributionFAQ.mspx

PCL

PCL® is the Printer Control Language developed by Hewlett Packard for its LaserJet (and compatible) printers. PCL bitmap fonts are used by the system. PCL bitmap fonts can have any file name extension. The system provides PCL fonts with an extension of PCL, such as *FPTIN8.PCL*. Like Metacode fonts, PCL characters are accessed by code points.

PostScript Fonts

PostScript fonts were designed and developed by Adobe Systems Incorporated. PostScript fonts are actually a special implementation of a PostScript language program. PostScript fonts are scalable fonts and there are several types of PostScript fonts, PostScript Type 1 fonts are most common and are the only type supported by the system. Typically, Type 1 font files have a file extension of PFB, such as *COURIER.PFB*.

Each character in a PostScript font has a PostScript character name. When used as a screen font, the operating system associates code points in a code page with the appropriate PostScript character names.

Note The system uses the CODEPAGE.INI file to associate code points with the appropriate PostScript characters.

TrueType Fonts

TrueType® is a scalable font designed and developed by Apple Computer and Microsoft for use on the Macintosh computer and on PCs running Microsoft Windows. The same font can be used with printers and video displays. Typically, TrueType font files have a file extension of TTF.

Adding Printer Fonts to a Font Cross-reference File

Fonts are added to an FXR file using Documaker Studio's Font manager. You can insert OpenType, TrueType, PCL, AFP, Xerox Metacode, certain FormMaker II files, and other FXR files into a font cross-reference file.

Note For more information on adding fonts to an FXR file, see the Documaker Studio User Guide. The FXR files included with the system do not contain OpenType fonts.

USING SYSTEM FONTS

Oracle Insurance has licensed for use and distribution with the system the following TrueType fonts:

Font	Description
Albany	Albany is a proportional/sans serif font (similar to Arial).
Cumberland	Cumberland is a fixed pitch/serif font (similar to Courier).
Thorndale	Thorndale is a proportional/serif font (similar to Times New Roman).
Andale Duospace WT	Andale Duospace WT is a fixed pitch/sans serif font (similar to Letter Gothic).
MICR MT	MICR MT is a symbol font using for check printing.

In addition, you can use the included AFP, Metacode, PCL, and PostScript versions of these fonts for printing.

From these fonts, we have created fonts to use with AFP (240 and 300 DPI), PCL, and Xerox printers. These fonts let you print nearly identical forms on any supported printer. This file naming convention is used for AFP, PCL, and Xerox printer fonts:

F T F1 F1 S P

For example, a 10 point bold Albany Xerox font would be named *FXAAB0.FNT*.

F	Standard Documaker system font
T	Printer type where A = AFP, P = PCL, X = Xerox 0 degree, 9 = Xerox 90 degree, 1 = Xerox 180 degree, 2 = Xerox 270 degree
F1	Two-character family name where AA = Albany, AD = Andale Duospace, CU = Cumberland, DM = Data Matrix for Documaker, TH = Thorndale, MR = MICR MT
S	Style where B = Bold, I = Italic, O = Bold Italic, N = Normal/Medium
P	Point size where 1 - 9 = point sizes 1-9 and 0 = point size 10 A - Z = point sizes 11-36

Font Cross-reference Files for the System Fonts

Here are the font cross-reference files for the system fonts:

Font cross-reference file	For
REL121.FXR	Windows code page 1252 (N. America and Western Europe)
EASTEUR.FXR	Windows code page 1250 (Eastern Europe)
CYRILLIC.FXR	Windows code page 1251 (Russia)
GREEK.FXR	Windows code page 1253 (Greece)
TURKISH.FXR	Windows code page 1254 (Turkey)
BALTIC.FXR	Windows code page 1257 (Estonia, Latvia, Lithuania)

You can also download these Asian FXRS and font packages from the Oracle Software Delivery Cloud at:
https://edelivery.oracle.com

REL121J.FXR	Japanese
REL121K.FXR	Korean
REL121SC.FXR	Simplified Chinese
REL121TC.FXR	Traditional Chinese

Here is a list of the TrueType fonts included in the REL121.FXR file:

TrueType Font	TrueType Font Name
aduo.ttf	Andale Duospace
aduob.ttf	Andale Duospace Bold
albw.ttf	Albany AMT
albwb.ttf	Albany AMT Bold
albwi.ttf	Albany AMT Italic
albwbi.ttf	Albany AMT Bold Italic
cumbwr__.ttf	Cumberland AMT
cumbwb__.ttf	Cumberland AMT Bold
cumbwi__.ttf	Cumberland AMT Italic
cumbwbi_.ttf	Cumberland AMT Bold Italic
dm_____.ttf	Data Matrix for Documaker
micr____.ttf	Micr MT
pdf417__.ttf	Pdf417
thowr___.ttf	Thorndale AMT
thowb___.ttf	Thorndale AMT Bold
thowi___.ttf	Thorndale AMT Italic
thowbi__.ttf	Thorndale AMT Bold Italic

USING CUSTOM FONTS

To the system, custom fonts are simply fonts which are not based on the ANSI code page. This means that the font contains characters which have different code points or which do not exist in the ANSI code page. If you cannot use the system fonts (or at least ANSI code page based fonts), you will need to consider these possible issues:

- Viewing Forms

  Viewing forms may be the first problem since the characters in the original printer font do not match the characters used in displaying text on the screen. This problem will be seen during forms composition. This will also be a problem if the you have licensed the Entry or Archive Retrieval modules. Keyboard entry becomes a training issue as well. Under Windows, you must use 4-digit Alt key sequences to prevent code point translation.

  If possible, you should convert any custom fonts to TrueType fonts for Windows and install the fonts into your operating system. If the font cross-reference file is properly modified to specify these screen fonts, the system will display your forms correctly. However, these characters may not display properly in Documaker Workstation.

- PDF Incompatibility

  In addition to the Entry and Archive module problem, PDF or Acrobat files created for Internet archive retrieval use the ANSI code page for displaying forms. Therefore, archived forms based on custom fonts may not display correctly when retrieved through Docupresentment.

- Printing Forms

  Another problem concerns using custom fonts on multiple (ASCII and EBCDIC) platforms. The system performs ASCII/EBCDIC translation based on the assumption that the ASCII code page is the ANSI code page and that the EBCDIC code page is code page 37. The system also assumes that PCL, PostScript, and Metacode printers use ASCII (hence ANSI) fonts. The system assumes AFP printers use EBCDIC fonts. The following table shows when the system will translate text (from FAP files) and variable data (from extract files) when printed under different platforms and printers.

Platform / Printer	ASCII (Windows 32-bit) ASCII FAP files and Extract data	EBCDIC (z/OS, AS400) EBCDIC FAP files and Extract data
AFP	ASCII to EBCDIC translation	No translation
PCL	No translation	EBCDIC to ASCII translation
PostScript	No translation	EBCDIC to ASCII translation
Xerox Metacode	No translation	EBCDIC to ASCII translation

On AFP printers

On a PC, text will be translated when printing to an AFP printer. Therefore, the code points used in text or variable data on forms are very important. After these code points are translated to the EBCDIC (code page 37), they must match the code points associated with the desired characters in the AFP code page which will be used.

On EBCDIC platforms, such as z/OS, AS400, text is assumed to be EBCDIC and will not be translated when you print to an AFP printer. The key to correct printing is to make sure the text (FAP files) and variable data (extract files) use the code points associated with the desired characters in the AFP code page you will use. Since FAP files are created as ASCII files on a PC, they will need to be transferred to the EBCDIC platform. Since you are using custom fonts, it is quite likely the file transfer software will not perform the proper code point translation. In this case, you may need to upload the files without translation and use the CPCNV utility to translate the files. This may require defining a special code page in the CODEPAGE.INI file for the CPCNV utility to use to do the proper translation.

On Xerox Metacode printers

On a PC, text (code points) will not be translated when printing to a Metacode printer.

On EBCDIC platforms (z/OS, AS400), text is assumed to be EBCDIC and will be translated to ASCII (ANSI code page) when printing to a Metacode printer. Therefore, the EBCDIC code points used in text or variable data on forms are very important. Since the FAP files are ASCII files created on a PC, they will need to be transferred to the EBCDIC platform. Since you are using custom fonts, it is quite likely that the file transfer software will not perform the proper code point translation. In this case, you may need to upload the files without translation and use the CPCNV utility to translate the files. This may require defining a special code page in the CODEPAGE.INI file for the CPCNV utility to use to do the proper translation.

On PCL printers

On a PC, text (code points) will not be translated when printing to a PCL printer. On EBCDIC platforms (z/OS, AS400), PCL print is not supported.

On PostScript printers

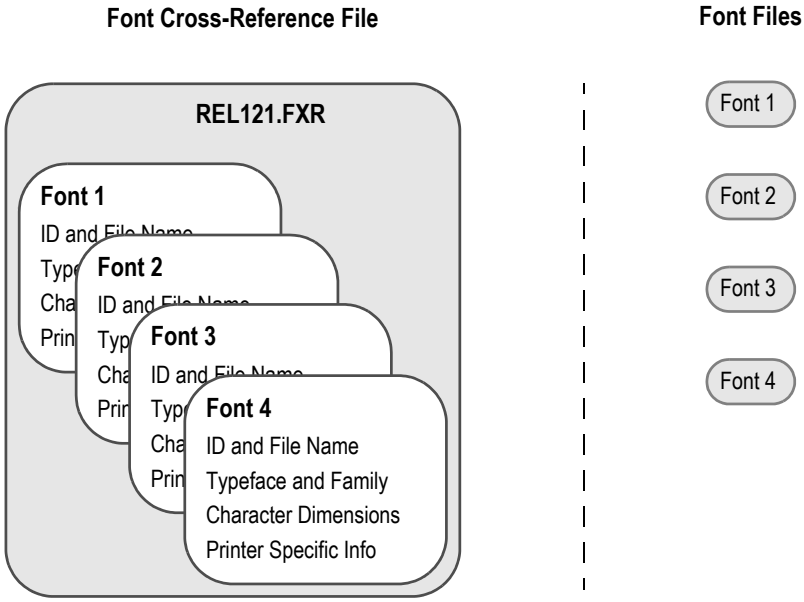
On a PC, text (code points) will not be translated when printing to a PostScript printer. On EBCDIC platforms (z/OS, AS400), PostScript print is not supported.

USING FONT CROSS-REFERENCE FILES

The font cross-reference (FXR) file lets you organize the fonts you use for display and printing. The FXR file provides the system with all the necessary font information. It does not contain the actual font files; rather, it contains information about the font attributes. Font attribute information includes formatting styles (bold, italic, and so on), point size (10 point, 14 point, and so on), and font stroke weight (heavy, light, and so on).

Note Storing the cross-reference information separately from the physical fonts affords greater flexibility in printer and font usage. You can convert virtually any font for your individual printer environment, provided you obtain appropriate license agreements for the fonts.

Let's examine the organization of the font cross-reference file and the font files. The illustration below depicts a font cross-reference file named REL121.FXR. This file contains a single font set. It includes all the crucial information for each font in the font set. The actual font files are physically separate from the font cross-reference file.



As shown above, the font files are distinct from the font cross-reference file. When you work with the font cross-reference file you affect the stored font information. You do not affect the separate and independent font files. The number of available fonts is limited only by your needs and storage space. If you keep this organizational structure in mind you can easily work with the font cross-reference file.

The font cross-reference file provides the names of your independent font files, but it is more than a simple listing of fonts. The file contains crucial font attribute information along with information specific to your printer types. The printer information is crucial because sections are compiled based on your printer environment.

The font cross-reference file ends in the extension *FXR* (for font cross-reference). The system includes these font cross-reference files:

Font cross-reference file	For
REL121.FXR	Windows code page 1252 (N. America and Western Europe)
EASTEUR.FXR	Windows code page 1250 (Eastern Europe)
CYRILLIC.FXR	Windows code page 1251 (Russia)
GREEK.FXR	Windows code page 1253 (Greece)
TURKISH.FXR	Windows code page 1254 (Turkey)
BALTIC.FXR	Windows code page 1257 (Estonia, Latvia, Lithuania)

Keep in mind these points concerning the FXR file:

- Contains one font set

  The font set is the specific group of fonts you choose to include in your font cross-reference file. Each font cross-reference file contains a single font set. You assign each font cross-reference file and font set a unique name. For example, you might organize a font set for creating and printing accounting forms in a font cross-reference file called ACCOUNT.FXR.

- Contains information on multiple fonts

  A font set contains numerous fonts. For example, a font set might contain Times New Roman fonts and Gothic fonts of multiple point sizes with bold and italic attributes. A second font set might contain Courier fonts and Helvetica fonts, also of multiple point sizes with bold, italic and regular attributes.

- Independent of your font files

  The font cross-reference file works with the printer and window font files. Remember that the font files are separate files from the font cross-reference file.

How FXR Settings Affect Display and Print Quality

Certain attributes in the FXR file affect how the system displays text. For example, when the system displays text, it uses scalable font technology which exists in Adobe Type 1 Postscript fonts and TrueType fonts. All versions of Windows support TrueType fonts. Windows 2000 and higher also support PostScript fonts.

These fonts are selected via the family name specified in our FXR, and scaled according to point size, height and width parameters in the FXR. The fonts are spaced according to the character widths specified in the FXR.

Once the font is selected, then it can be zoomed in and out, or additionally scaled as required. Bitmap fonts do not have this scaling ability, which is why scalable fonts are used for display purposes, rather than bitmap fonts.

This means that when the system displays text on the screen, it attempts to mirror how it will look on paper. To achieve the same look on the screen as on paper, the parameters in the FXR are critical. The more accurate the FXR is, the more likely the display will mirror the printed document. The printed document is the standard for the screen display.

Since the system includes TrueType fonts which match its printer fonts, if you install these fonts on a Windows system, what you see on your screen will more closely match what you print out. The keys are to closely match the printer's fonts and to have the best possible information in the FXR file.

Creating a font cross-reference file is usually done by importing a printer font file using Font manager. Since the font cross-reference file is a representation of information contained in the printer font file, modification of its fields usually does not affect the printed output. However, modifying these FXR fields can improve the system's ability to display forms.

MAINTAINING FXR FILES

Use the Font manager to maintain FXR files. You can start this tool in Studio using the Manage, System, Fonts option. Font manager makes it easy to insert, edit, copy or delete font information in the FXR file.

INTERNATIONAL LANGUAGE SUPPORT

The goal for international language support is to support the languages you are most likely to need, such as those languages used in the Western Hemisphere and Western Europe.

If you need support for Far Eastern languages like Chinese, Japanese, or Korean or if you need support for Eastern European languages, you must use version 10.2 or higher. See the Unicode Reference for more information.

USING ANSI CODE PAGES ON WINDOWS PLATFORMS

Documaker supports the languages you are most likely to need, such as those languages used in the Western Hemisphere and Europe. If you need support for Far Eastern languages like Chinese, Japanese, Korean, or Vietnamese, you must use the Unicode functionality found in version 10.2 or higher. See the Unicode Reference for more information.

Microsoft has defined a number of code pages known as Windows ANSI code pages. Documaker has support for the following ANSI code pages:

Code page	Description
1250	Central and East European Latin. Code Page 1250 is used for languages such as Polish, Czech, Slovak, Hungarian, Slovene, Bosnian, Croatian, Serbian (Latin script), Romanian and Albanian.
1251	Cyrillic. Code Page 1251 is used for languages such as Russian, Bulgarian, and Serbian Cyrillic.
1252	West European Latin. Code Page 1252 is used for languages such as Afrikaans, Basque, Catalan, Danish, Dutch, English, Faroese, Finnish, French, Galician, German, Icelandic, Indonesian, Italian, Malay, Norwegian, Portuguese, Spanish, Swahili, Swedish
1253	Greek. Code Page 1253 is used for the modern Greek language
1254	Turkish. Code Page 1254 is used for the Turkish language
1257	Baltic. Code Page 1257 is used for the Estonian, Latvian and Lithuanian languages

Code Page 1004 is IBM's equivalent code page to the Windows code page 1252.

Documaker provides FXR files and fonts for each of these code pages.

Note See *Using International Characters on page 29* for more information.

USING CODE PAGE 37 FOR EBCDIC PLATFORMS

For international languages on EBCDIC platforms, such as z/OS and AS400, use EBCDIC code page 37 as the standard EBCDIC code page. Code page 37 is the native code page for many z/OS systems. By using code page 37, you receive these benefits:

- Code page 37 supports languages used in Europe and North and South America, such as French, Spanish, Italian, German, Portuguese, and Danish.
- This reduces or eliminates the need to convert extract files containing international characters on z/OS.
- This helps reduce or eliminate the need to convert resources before uploading to EBCDIC platforms from Windows.
- Using code page 37 for EBCDIC platforms creates compatibility problems with resources created in earlier versions. This only affects resources created in an earlier version which contain international or desktop publishing characters.
- All characters defined in code page 37 are also contained in code page 1004, the standard ASCII code page. There are, however, characters in code page 1004 which are not in code page 37—mainly desktop publishing characters from code point 128 to 159. To support these characters, Documaker uses undefined code points in code page 37 (code points below 64). For maximum portability, *avoid* using characters not defined in code page 37.

AFP print output and resource files normally use EBCDIC characters. The other supported printers, such as Metacode, PCL, and PostScript, normally use ASCII characters.

Note Documaker includes a set of AFP fonts that can be used for languages represented by code pages 1250, 1251, 1252, 1253, 1254, and 1257.

USING INTERNATIONAL CHARACTERS

One method for entering international characters is to install a country/language specific version of Windows. These language-specific versions of Windows map characters from the keyboard differently so that it is easier to enter characters common to that language. In the simplest case, a single keystroke will generate an international character.

For example, if you have a Canadian French version of Windows, pressing the slash character (/) on a U.S. keyboard produces an e-acute letter (*é*). Many international characters require a two-character keystroke combination. Again using the Canadian French keyboard setup, you must press the left square bracket ([ ) followed by the letter *e* to generate an e-circumflex letter (*ê*).

Having to install a special version of Windows would be difficult for those in the U.S. who are trying to compose forms with French characters. Fortunately, there is a simpler solution.

Using the numeric keypad on the right side of your keyboard, you can hold down the Alt key and enter a three-digit number to enter an international character. For example, if your primary (OEM) code page is 437 or 850, you can enter the letter *ñ* (lowercase) by pressing the Alt key while you type 164 on the numeric keypad. When you release the Alt key, the code point 164 will be generated by the keyboard, which Windows will display as the letter *ñ*.

Note If you look at the code page 1004 table you will see that on the ANSI code page code point 164 is not the letter *ñ*. So why is the letter *ñ* being displayed? Windows recognizes that a code point of 164 has been generated by the keyboard and it is associated with the OEM code page (437 or 850). For this code page, code point 164 maps to the letter *ñ*. In Windows, the code point from the keyboard is translated from 164 to 241. A Windows program will actually receive a keystroke code point of 241 instead and that code point will display as the desired letter ñ.

You can also use the numeric keypad to enter ANSI code points directly. Using the numeric keypad on the right side of your keyboard, you can press the ALT key and type a four-digit number to key in an international character. For example, you can enter in the letter ñ by pressing the Alt key and typing 0241 on the numeric keypad. Entering a four-digit number beginning with a zero tells Windows you are entering a code point for the ANSI code page. Therefore, Windows does not need to translate the code point and passes the keystroke code point directly to the Windows application.

By standardizing on the ANSI code page, a document containing several languages can be read and written by a number of people from different countries. The keystroke code point translation lets Windows support many OEM code pages and keyboard settings.

Note You can use any Windows text editor, such as Notepad, to edit resource files since Windows also uses the ANSI code page.

Converting Text Files from one Code Page to Another

There are two situations where you may need to convert text files from one code page to another.

- If the customer's data (extract) file is not in the ANSI code page and the file contains international characters, you will need to convert the customer data file to use the ANSI code page.
- If you need to upload system resource files, such as FAP, INI, and menu resource (MEN.RES) files, which contain international characters to an EBCDIC platform, such as z/OS, and the file transfer software cannot convert ANSI code page file to EBCDIC code page 37.

To convert a file from one code page to another, you can use the CPCNV code page conversion utility. For more information, see the Utilities Reference.

SETTING UP POSTSCRIPT FONTS

The system includes two PostScript fonts, PDF417, and Data Matrix for Documaker. These fonts reside in the FAP\MSTRRES\FMRES\DEFLIB\ directory with the sample forms included with Documaker Studio. The following naming conventions are used for the bitmap printer fonts that are created from the PostScript fonts supported by the system. PostScript fonts are easily converted to Xerox, AFP, and PCL formats.

Note When you create bitmap printer fonts from PostScript fonts, follow the naming convention outlined in the table below. This will make it easier to track and identify those fonts.

A standard font has a six-character name. Each character indicates a specific piece of data that describes the font. For example, you may take a PostScript font such as Data Matrix for Documaker (DM_____.PFB), convert the font to Metacode format, and change the name to the standard bitmap font name (FXDMN6). The font name characters designate the following:

Character	Definition
1	Converted PostScript fonts always begin with the letter F, indicating a system supported font.
2	Indicates the printer platform associated with the converted font: X = Xerox, A = AFP, P = PCL
3 and 4	Indicate the font family, such as Times Roman, Courier, and so on. AA = Albany, AD = Andale Duospace, CU = Cumberland, DM = Data Matrix for Documaker,TH = Thorndale, MR = MICR MT
5	Indicates the style of the font: N = Normal (no attributes), B = Bold, I = Italic, O = Bold, Italic
6	Indicates the point size of the font. Use numbers 1 through 9 for point sizes 1 through 9. 0 (zero) = 10 point A = 11 point B = 12 point C = 13 point--through--Z = 36 point

FONTS FOR PDF FILES

When you are creating PDF files, keep in mind that the following fonts are included with Adobe Acrobat Reader and do not have to be embedded.

Fixed Pitch Fonts	Proportional Fonts
Courier	Helvetica
Courier-Bold	Helvetica-Bold
Courier-Oblique	Helvetica-Oblique
Courier-BoldOblique	Helvetica-BoldOblique
	Times-Roman
	Times-Bold
	Times-Italic
	Times-BoldItalic
	Symbol
	ZapfDingbats

Importing PostScript Symbol Fonts

You can select a code page when importing PostScript symbol fonts, such as Euro Sans and ITD Zapf Dingbats, which contain characters that do not adhere to a standard Windows code page.

In Font manager, select *9999,WD* as the code page when importing these types of PostScript fonts.

Note For normal fonts, you should continue to select *1004,W1* as the code page.

If you import a PostScript font using code page 1004,W1 and the system produces a font record with only a few non-zero character widths or produces an internal error, try using code page 9999,WD to import the font.

For instance, importing Euro Sans and ITC Zapf Dingbats using code page 1004,W1 produces a font record where only the space and hard space characters (code points 32 and 160) contain non-zero character widths. Importing the same fonts using code page 9999,WD produces a font record with non-zero character widths for virtually every code point from 32 to 255.

When you use the PS2PCL utility to convert PostScript symbol fonts to PCL, specify the symbol set by setting the /S parameter to *WD*. This tells the utility that these PostScript fonts that contain characters that do not adhere to a standard Windows code page.

Note When converting normal text fonts with the PS2PCL utility, continue to set the /S parameter to *W1*.

FONT NAMING CONVENTIONS

The following FXR files are available for your use:

Font cross-reference file	Description
REL121.FXR	Windows code page 1252 (N. America and Western Europe)
EASTEUR.FXR	Windows code page 1250 (Eastern Europe)
CYRILLIC.FXR	Windows code page 1251 (Russia)
GREEK.FXR	Windows code page 1253 (Greece)
TURKISH.FXR	Windows code page 1254 (Turkey)
BALTIC.FXR	Windows code page 1257 (Estonia, Latvia, Lithuania)

Each font is identified by a font ID or five digit number. The FXR files use a naming convention to identify the font name, code page supported, font style, and point size. Here is a summary of the naming convention:

- The first three digits identify the font name and font style.
- The last two digits represent the point size of the font, such as 09 point, 12 point, and so on.

Font IDs	Font
EASTEUR.FXR - Central and East European Languages	
20004 to 20036	Albany AMT (4 to 36 point sizes)
20104 to 20136	Albany AMT Bold
20204 to 20236	Albany AMT Italic
20304 to 20336	Albany AMT Bold Italic
20404 to 20436	Cumberland AMT
20504 to 20536	Cumberland AMT Bold
20604 to 20636	Cumberland AMT Italic
20704 to 20736	Cumberland AMT Bold Italic
20804 to 20836	Thorndale AMT
20904 to 20936	Thorndale AMT Bold
21004 to 21036	Thorndale AMT Italic
21104 to 21136	Thorndale AMT Bold Italic
21204 to 21236	Andale Duospace WT
21304 to 21336	Andale Duospace WT Bold

Font IDs	Font
CYRILLIC.FXR - Russia	
22004 to 22036	Albany AMT (4 to 36 point sizes)
22104 to 22136	Albany AMT Bold
22204 to 22236	Albany AMT Italic
22304 to 22336	Albany AMT Bold Italic
22404 to 22436	Cumberland AMT
22504 to 22536	Cumberland AMT Bold
22604 to 22636	Cumberland AMT Italic
22704 to 22736	Cumberland AMT Bold Italic
22804 to 22836	Thorndale AMT
22904 to 22936	Thorndale AMT Bold
23004 to 23036	Thorndale AMT Italic
23104 to 23136	Thorndale AMT Bold Italic
23204 to 23236	Andale Duospace WT
23304 to 23336	Andale Duospace WT Bold
REL121.FXR - Western Hemisphere / Western Europe	
24004 to 24036	Albany AMT (4 to 36 point sizes)
24104 to 24136	Albany AMT Bold
24204 to 24236	Albany AMT Italic
24304 to 24336	Albany AMT Bold Italic
24404 to 24436	Cumberland AMT
24504 to 24536	Cumberland AMT Bold
24604 to 24636	Cumberland AMT Italic
24704 to 24736	Cumberland AMT Bold Italic
24804 to 24836	Thorndale AMT
24904 to 24936	Thorndale AMT Bold
25004 to 25036	Thorndale AMT Italic
25104 to 25136	Thorndale AMT Bold Italic
25204 to 25236	Andale Duospace WT

Font IDs	Font
25304 to 25336	Andale Duospace WT Bold
GREEK.FXR - Greece	
26004 to 26036	Albany AMT (4 to 36 point sizes)
26104 to 26136	Albany AMT Bold
26204 to 26236	Albany AMT Italic
26304 to 26336	Albany AMT Bold Italic
26404 to 26436	Cumberland AMT
26504 to 26536	Cumberland AMT Bold
26604 to 26636	Cumberland AMT Italic
26704 to 26736	Cumberland AMT Bold Italic
26804 to 26836	Thorndale AMT
26904 to 26936	Thorndale AMT Bold
27004 to 27036	Thorndale AMT Italic
27104 to 27136	Thorndale AMT Bold Italic
27204 to 27236	Andale Duospace WT
27304 to 27336	Andale Duospace WT Bold
TURKISH.FXR - Turkey	
28004 to 28036	Albany AMT (4 to 36 point sizes)
28104 to 28136	Albany AMT Bold
28204 to 28236	Albany AMT Italic
28304 to 28336	Albany AMT Bold Italic
28404 to 28436	Cumberland AMT
28504 to 28536	Cumberland AMT Bold
28604 to 28636	Cumberland AMT Italic
28704 to 28736	Cumberland AMT Bold Italic
28804 to 28836	Thorndale AMT
28904 to 28936	Thorndale AMT Bold
29004 to 29036	Thorndale AMT Italic
29104 to 29136	Thorndale AMT Bold Italic

Font IDs	Font
29204 to 29236	Andale Duospace WT
29304 to 29336	Andale Duospace WT Bold
BALTIC.FXR - Estonia, Latvia, Lithuania	
30004 to 30036	Albany AMT (4 to 36 point sizes)
30104 to 30136	Albany AMT Bold
30204 to 30236	Albany AMT Italic
30304 to 30336	Albany AMT Bold Italic
30404 to 30436	Cumberland AMT
30504 to 30536	Cumberland AMT Bold
30604 to 30636	Cumberland AMT Italic
30704 to 30736	Cumberland AMT Bold Italic
30804 to 30836	Thorndale AMT
30904 to 30936	Thorndale AMT Bold
31004 to 31036	Thorndale AMT Italic
31104 to 31136	Thorndale AMT Bold Italic
31204 to 31236	Andale Duospace WT
31304 to 31336	Andale Duospace WT Bold
All FXR files contain the following common fonts:	
911	PDF417 240 DPI 9x12 - 2.16 Pt
912	PDF417 300 DPI 9x12 - 2.16 Pt
1215	PDF417 240 DPI 12x16 - 2.88 Pt
1216	PDF417 300 DPI 12x16 - 2.88 Pt
13504 to 13536	Data Matrix for Documaker (4 to 36 point sizes)
32004 to 32036	MICR MT (4 to 36 point sizes)

Note You can only use a font ID from 00001 to 32767 and the font ID must be numeric not alphanumeric.

MAPPING FONTS FOR FILE CONVERSIONS

When converting a file from one format to another, you may need to convert the fonts used in the document. You can use INI control groups and options to map fonts in a source document to the fonts you want to use in the destination document. For instance, if you are converting an RTF file into a FAP file, you can use the following control group:

```
< RTFFontMAP >
   Arial = Swiss
```

This tells the system to convert all Arial fonts into Swiss fonts. Use this control group when converting DCD files into FAP files:

```
< FontFamilyMatching >
   Arial = Swiss
```

Place these control groups and options in the FAPCOMP.INI file.

RTF and DCD files contain font information about the generic font families used. For example, Arial and Univers, both san serif proportional fonts, belong to a generic font family called *Swiss*.

The RTF and DCD converters in the system use the RTFFontMap and FontFamilyMatching control groups to assign a font when other means of mapping fonts from the RTF or DCD file fails.

In Windows environments, there are several generic font families, as shown in this table:

Family	Description
Decorative	Specifies a novelty font, such as Old English.
Dontcare	Specifies a generic family name. This name is used when information about a font does not exist or does not matter. The default font is used.
Modern	Specifies a monospace font with or without serifs. Monospace fonts are usually modern fonts, such as Pica, Elite, and Courier New.
Roman	Specifies a proportional font with serifs, such as Times New Roman.
Script	Specifies a font that is designed to look like handwriting, such as Script and Cursive.
Swiss	Specifies a proportional font without serifs, such as Arial.

Here is a list of the system fonts:

This typeface	Style
Albany	Proportional/Sans Serif
Albany Bold	Proportional/Sans Serif
Albany Italic	Proportional/Sans Serif
Albany Bold Italic	Proportional/Sans Serif
Cumberland	Fixed Pitch/Serif

This typeface	Style
Cumberland bold	Fixed Pitch/Serif
Cumberland italic	Fixed Pitch/Serif
Cumberland bold italic	Fixed Pitch/Serif
Thorndale	Proportional/Serif
Thorndale bold	Proportional/Serif
Thorndale italic	Proportional/Serif
Thorndale bold italic	Proportional/Serif
Andale Duospace WT	Fixed Pitch/Sans Serif
Andale Duospace WT Bold	Fixed Pitch/Sans Serif
MICR MT	Symbol font (check printing)

INDEX