

The logo for Documaker, featuring the word "Documaker" in a white serif font with a trademark symbol, set against a solid blue rectangular background.

Documaker™

Documaker

Version 11.2

Features and Enhancements

Docucorp International, a subsidiary of Skywire Software, announces Documaker version 11.2. This document introduces version 11.2 and describes its features and enhancements.

To receive the full benefits of the new product features included in this and earlier releases, Skywire Software's Education Services Group offers a comprehensive range of training classes. Classes are taught hands-on at Skywire Software's training facilities in Atlanta. For a list of courses, including fees and availability, please call 770.859.9900.

Installation

Please follow the installation instructions included in the readme file on the CD or in the installation guide. All documentation is stored on the CD, in the \doc\rel112 directory.

You can also visit the following internet site to see the latest documentation and to install software:

www.docucorp.com/support

You must register to use this site.

Skywire Software, L.L.C.
2401 Internet Boulevard
Suite 201
Frisco, Texas 75034
www.skywiresoftware.com

Phone: (U. S.)972 377 1110
(EMEA)+44 (0) 1372 366 200
FAX: (U. S.)972 377 1109
(EMEA)+44 (0) 1372 366 201
Support: (U. S.) 866 844 8780
(EMEA) +44 (0) 1372 366 222

contactus@skywiresoftware.com

PUBLICATION COPYRIGHT NOTICE

Copyright © 2007 Skywire Software, L.L.C. All rights reserved.

Printed in the United States of America.

This publication contains proprietary information which is the property of Skywire Software or its subsidiaries. This publication may also be protected under the copyright and trade secret laws of other countries.

TRADEMARKS

Skywire® is a registered trademark of Skywire Software, L.L.C.

Docucorp®, its products (Docucreate™, Documaker™, Docupresentment™, Docusave®, Documanager™, Poweroffice®, Docutoolbox™, and Transall™), and its logo are trademarks or registered trademarks of Skywire Software or its subsidiaries.

The Docucorp product modules (Commcommander™, Docuflex®, Documerge®, Docugraph™, Docusolve®, Docuword™, Dynacomp®, DWSD™, DBL™, Freeform®, Grafxc commander™, Imagecreate™, I.R.I.S.™, MARS/NT™, Powermapping™, Printcommander®, Rulecommander™, Shuttle™, VLAM®, Virtual Library Access Method™, Template Technology™, and X/HP™ are trademarks of Skywire Software or its subsidiaries.

Skywire Software (or its subsidiaries) and Mynd Corporation are joint owners of the DAP™ and Document Automation Platform™ product trademarks.

Docuflex is based in part on the work of Jean-loup Gailly and Mark Adler.

Docuflex is based in part on the work of Sam Leffler and Silicon Graphic, Inc.

Copyright © 1988-1997 Sam Leffler.

Copyright © 1991-1997 Silicon Graphics, Inc.

Docuflex is based in part on the work of the Independent JPEG Group.

The Graphic Interchange Format® is the Copyright property of CompuServe Incorporated. GIFSM is a Service Mark property of CompuServe Incorporated.

Docuflex is based in part on the work of Graphics Server Technologies, L.P.

Copyright © 1988-2002 Graphics Server Technologies, L.P.

All other trademarks, registered trademarks, and service marks mentioned within this publication or its associated software are property of their respective owners.

SOFTWARE COPYRIGHT NOTICE AND COPY LIMITATIONS

Your license agreement with Skywire Software or its subsidiaries, authorizes the number of copies that can be made, if any, and the computer systems on which the software may be used. Any duplication or use of any Skywire Software (or its subsidiaries) software in whole or in part, other than as authorized in the license agreement, must be authorized in writing by an officer of Skywire Software or its subsidiaries.

PUBLICATION COPY LIMITATIONS

Licensed users of the Skywire Software (or its subsidiaries) software described in this publication are authorized to make additional hard copies of this publication, for internal use only, as long as the total number of copies does not exceed the total number of seats or licenses of the software purchased, and the licensee or customer complies with the terms and conditions of the License Agreement in effect for the software. Otherwise, no part of this publication may be copied, distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, manual, or otherwise, without permission in writing by an officer of Skywire Software or its subsidiaries.

DISCLAIMER

The contents of this publication and the computer software it represents are subject to change without notice. Publication of this manual is not a commitment by Skywire Software or its subsidiaries to provide the features described. Neither Skywire Software nor its subsidiaries assume responsibility or liability for errors that may appear herein. Skywire Software and its subsidiaries reserve the right to revise this publication and to make changes in it from time to time without obligation of Skywire Software or its subsidiaries to notify any person or organization of such revision or changes.

The screens and other illustrations in this publication are meant to be representative, not exact duplicates, of those that appear on your monitor or printer.

MONOTYPE FONT(S) SOFTWARE LICENSE AGREEMENT

- 1 "Software" shall mean the digitally encoded, machine readable, scalable outline Intellifont/ data as encoded in a special format as well as the Intellifont Software.
- 2 You agree to accept a non-exclusive license to use the Software to reproduce and display weights, styles and versions of letters, numerals, characters and symbols ("Typefaces") solely for your own customary business purposes which shall not include use or access by entities other than you. Under the terms of this License Agreement, you have the right to use the Monotype Imaging, Inc., Typefaces only in conjunction with the Docucorp International, Inc., a subsidiary of Skywire Software, LLC, Documentation Software licensed by you. Monotype (formerly Agfa) retains all rights, title and interest to the Software and Typefaces and no rights are granted to you other than a License to use the Software on the terms expressly set forth in this Agreement.
- 3 To protect proprietary rights of Monotype, you agree to maintain the Software and other proprietary information concerning the Typefaces in strict confidence and to establish reasonable procedures regulating access to and use of the Software and Typefaces.
- 4 You agree not to duplicate or copy the Software or Typefaces, except that you may make one backup copy. You agree that any such copy shall contain the same proprietary notices as those appearing on the original.
- 5 This License shall continue until the last use of the Software and Typefaces, unless sooner terminated. This License may be terminated by Monotype if you fail to comply with the terms of this License and such failure is not remedied within thirty (30) days after notice from Monotype. When this License expires or is terminated, you shall either return to Monotype or destroy all copies of the Software and Typefaces and documentation as requested.
- 6 You agree that you will not modify, alter, disassemble, decrypt, reverse engineer or decompile the Software.
- 7 Monotype warrants that for ninety (90) days after delivery, the Software will perform in accordance with Monotype-published specifications and diskette will be free from defects in material and workmanship. Monotype does not warrant that the Software is free from all bugs, errors and omissions.
- 8 THE PARTIES AGREE THAT ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE AND MERCHANTABILITY, ARE EXCLUDED.
- 9 Your exclusive remedy and the sole liability of Monotype in connection with the Software and Typeface is repair or replacement of defective parts, upon their return to Monotype.
- 10 IN NO EVENT WILL MONOTYPE IMAGING, INC., BE LIABLE FOR LOST PROFITS, LOST DATA, OR ANY OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES CAUSED BY THE ABUSE OR MISAPPLICATION OF THE SOFTWARE AND TYPEFACES.
- 11 Massachusetts U.S.A. Law governs the Agreement.
- 12 You shall not sublicense, sell, lease, or otherwise transfer the Software and/or Typefaces without the prior written consent of Monotype.
- 13 Use, duplication or disclosure by the Government is subject to restrictions as set forth in the Rights in Technical Data and Computer Software clause at FAR 252-227-7013, subdivision (b)(3)(ii) or subparagraph (c)(1)(ii), as appropriate. Further use, duplication or disclosure is subject to restrictions applicable to restricted rights software as set forth in FAR 52.227-19 (c) (2).
- 14 YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. NEITHER PARTY SHALL BE BOUND BY ANY STATEMENT OR REPRESENTATION NOT CONTAINED IN THIS AGREEMENT. NO CHANGE IN THIS AGREEMENT IS EFFECTIVE UNLESS WRITTEN AND SIGNED BY PROPERLY AUTHORIZED REPRESENTATIVES OF EACH PARTY. BY OPENING THIS DISKETTE PACKAGE, AND YOU AGREE TO ACCEPT THE TERMS AND CONDITIONS OF THIS AGREEMENT.

CONTENTS

Summary of Features

- 2 Using the Feature Summary
- 3 Important Considerations when Upgrading
- 4 Summary of Features

List of Features

- 10 Overview
 - 16 Using the FAP2PDF Utility
 - 16 Purging Information from the Library Log
 - 17 300 DPI Support for AFP Files
 - 18 Adding Hypertext Links
 - 20 Improved Support for Color and Graphics
 - 21 Importing RTF Files
 - 21 RTF2FAP utility
 - 21 Image Editor
 - 21 Limitations
 - 22 Logging INI File Names and Options in the TRACE File
 - 22 Moving Through the List of Fonts
 - 23 Support Added to Vectors for the Pre-printed Option
 - 24 GDI Print Supports GenPrint Batch Printing to Files
 - 25 Easier Handling of Overflow
 - 29 Overflow Examples
 - 29 Image-level overflow
 - 29 Form-level overflow
 - 30 Form set (Image)-level overflow
 - 31 New User Functions
 - 33 Overflow Variables
 - 33 Automatically Re-Paginating Images
 - 35 Adding Security at the Resource Level
 - 36 Storing the WIP Index and WIP Data in Database Tables
 - 37 Using this Feature
 - 38 Setting Up Your Database
 - 38 Example WIP.DAL file

39	Example WIP.DFD file
44	Example WIPDATA.DAL file
44	Example WIPDATA.DFD file
45	Changing the CARDATA field
47	Example INI4WIP.DAL file
48	Manually updating INI files
52	Extracting EDL Forms from a VLAM File
54	Converting FAP and FOR Files into XML Files
57	Using the New Image Usage Report
60	Generating the USPS Intelligent Mail Barcode (4-State Customer Bar Code)
61	Using the New Undo/Redo Option
62	Viewing Field Information
63	Specifying Return Values
63	Using the New ResetImageDimension Rule Options
64	Retaining the Duplex Value when Converting Normalized Metacode Files
64	Controlling Access to the Field Dictionary Manager
65	Using the New Toolbar Buttons
66	Docking and Automatically Hiding Windows
68	Setting the Default Effective Date for New Resources
69	Highlighting Selected Objects
69	Using the Form Manager Navigation Tree
70	Showing the Object Name on the Status Bar
71	Easier Navigation in Multi-Page FAP Files
72	Adding Code Page and Character Set Names to the FXR
72	Using the Char and CharV DAL Functions
72	Char
72	CharV
73	Editing LOG Files from within Image Manager
74	Including RDP Information on Group Reports
74	Using Superscript and Subscript in Text Labels, Text Areas, and Variable Fields
76	Dragging and Dropping Objects
76	Customizing the Library Manager Window
77	Using the New ProcessTriggers Rule
78	Displaying Page Numbers
79	Better Display when Resizing Lines
79	Progress Bars During Report Generation

80	Printing a Specific Page of a Form Set
81	LBYPROC Utility Enhancements
81	Miscellaneous Enhancements to the Studio Interface
82	Miscellaneous Business Definition Manager Changes
83	Miscellaneous Enhancements to Studio's Group Manager
84	Miscellaneous Changes to Studio's Image Manager
84	Miscellaneous Changes to Studio's Script Manager
85	Miscellaneous Changes to Studio's Library Manager
86	Miscellaneous Changes to Studio's Dictionary Managers
87	Miscellaneous Changes to Studio's Conversion Manager
87	Miscellaneous Changes to Studio's Reports Manager
88	Miscellaneous Changes to Studio's Test Manager
88	Miscellaneous Enhancements to Deployment Manager
88	New Default for the CheckNextRecip Option
89	Using the New BDF2FDT Utility
90	Converting EPS Files into LOG Files
90	The AFP Loader Now Uses Character Set And Code Page Font Information
91	Prompting for Search Criteria
91	Creating Smaller PostScript Output
92	Specifying a DFD File for the Library Data Table
92	Specifying a Catalog Table for Each Library
92	Suppressing LOG Files When Converting AFP to FAP or PDF
93	Outputting AFP Character Set and Code Page Information
94	Support for Version 1 Xerox FRM Files
94	Enhanced Error Checking for the MRGCHK Utility
94	Marking Objects as Pre-printed
96	Enhanced DAL Profile Output
98	Importing CSV Files into an Extract Dictionary
99	Creating a CSV File
100	Getting the JDEName, JDLName, CodeDef, and CommentRecords Options
102	Using the New FONTLIST Utility
104	Improved Support for PostScript Symbol Fonts
104	Generating DDL Files from DFD Files
105	Enhanced Support for Color Logos
106	Enhanced Help System
108	Mapping Alternate WIP Index Columns from Imported WIP Data
108	Suppressing Duplicate Form Descriptions

109	Showing Field Information in PDF Files
110	Using Mixed Mode AFP Datasets
110	Making Banner Forms Compatible with Multi-file Print
112	Support for QuarkXPress Version 7.0
112	Defining the Content in the MultiFilePrint Log
113	New Look and Feel for Documaker Studio
114	Using the New CheckZeroFontID Rule
115	Additional Documanage Data Type Support in Documaker Server Archive
116	Automatic Category Overrides
117	Truncating Fields
117	Creating FOR Files when Converting FAP Files
119	Importing TGA Files into Your Extract Dictionary
120	Paginating Forms Using DAL Functions
120	PaginateForm
120	AddImage
121	DellImage
122	Controlling Pagination when Editing a Form Set
122	Automatically Saving Form Sets in iDocumaker Workstation
123	Using the New Font Usage Report
124	Generating PDF417 Barcodes
125	Opening Documaker Studio Files
125	Defaults for the Module and PrintFunc Options
126	Searching for DAL Scripts
127	Using the New STRCompare Function
128	Loading Logos from the Library when Converting AFP or Metacode Files
128	Paper Stock Support in the Metacode Print Driver
129	New Ways to Select Objects
130	Using the New Complete and Exit Option
131	Enhanced AFP Conversions
134	Copying the EDL Form Description into the Library Description Field
135	Improved WIP Selection Performance During Batch Processing
136	Expanded Search Capability
136	Creating an Image from an Area
137	Using the Measurement Tool
138	Using the New Font Cross-Reference (FXR) Files
139	Using the FDT2DB Utility's New Purge Parameter
139	Using the New RunTriggers Rule

CHAPTER 1

Summary of Features

Version 11.2 includes features and enhancements that improve the functionality and ease of use of our suite of products. This document provides detailed information on the specific features and enhancements in these areas:

- [Using the Feature Summary on page 2](#)
- [Summary of Features on page 4](#)

USING THE FEATURE SUMMARY

The following feature summary lists enhancements and certain new features of the Documaker suite of related products. Some of the enhancements or new features have already been made available as patches to the current shipping version. Where applicable, a feature number is listed for reference.

The features are listed in feature number order and with licensing information in chapter two. The [Summary of Features on page 4](#) organizes the features in functional areas and provides links to the detailed feature descriptions.

NOTE: If you have any questions about your license, please contact your sales representative.

IMPORTANT CONSIDERATIONS WHEN UPGRADING

When upgrading to version 11.2, please keep these changes in mind...

- Feature 2045 changes the default for the CheckNextRecip option. This option enables recipient batch records for the same transaction that occur in sequence in a recipient batch file to be queued and printed in one call to the print driver. The default for this option has changed from Yes to No, as shown here:

```
< RunMode >  
    CheckNextRecip = No
```

The new default minimizes the likelihood you would have to change the option's setting. For instance, this option must be set to No (now the default) when you are using the GenPrint program with print drivers that must print to a separate file for each recipient, such as the PDF, XML, HTML, and RTF print drivers.

For more information, see [New Default for the CheckNextRecip Option on page 88](#).

- Version 11.2 includes new font cross-reference (FXR) files. These include PDF417 fonts and corrects some problems with character widths and font heights.

For more information, see [Using the New Font Cross-Reference \(FXR\) Files on page 138](#)

- Documaker Studio and Image Editor can now produce 300 DPI output for Documaker AFP print streams and normalized AFP files. For Documerge customers who compose AFP forms at 300 dots per inch (DPI) via Printcommander 5.x/6.x, you no longer have to convert your AFP forms to 240 DPI to use Image Editor.

The Resolution option in your AFP printer control group controls the resolution of the coordinates used to produce Documaker AFP print streams and normalized AFP files for Documerge. The Resolution option also controls the resolution of the bitmap data produced when using inline logos.

For more information, see [300 DPI Support for AFP Files on page 17](#)

SUMMARY OF FEATURES

This table summarizes all the new features in this release. The features are grouped in these functional areas:

- [Archive Enhancements on page 4](#)
- [DAL Enhancements on page 4](#)
- [Studio Enhancements on page 4](#)
- [Docutoolbox enhancements on page 6](#)
- [Documaker Workstation enhancements on page 7](#)
- [Printer and font enhancements on page 7](#)
- [Rules processing enhancements on page 8](#)
- [Miscellaneous enhancements on page 8](#)

Feature	For more information, see...
Archive Enhancements	
2130	Additional Documanager Data Type Support in Documaker Server Archive on page 115
2131	Automatic Category Overrides on page 116
DAL Enhancements	
2143	Paginating Forms Using DAL Functions on page 120
2158	Using the New STRCompare Function on page 127
Studio Enhancements	
1861	Moving Through the List of Fonts on page 22
1866	Easier Handling of Overflow on page 25
1869	Adding Security at the Resource Level on page 35
1888	Extracting EDL Forms from a VLAM File on page 52
1891	Converting FAP and FOR Files into XML Files on page 54
1892	Using the New Image Usage Report on page 57
1901	Using the New Undo/Redo Option on page 61
1906	Viewing Field Information on page 62
1911	Specifying Return Values on page 63
1918	Retaining the Duplex Value when Converting Normalized Metacode Files on page 64
1932	Controlling Access to the Field Dictionary Manager on page 64

Feature	For more information, see...
1937	Using the New Toolbar Buttons on page 65
1938	Docking and Automatically Hiding Windows on page 66
1950	Setting the Default Effective Date for New Resources on page 68
1966	Highlighting Selected Objects on page 69
1971	Using the Form Manager Navigation Tree on page 69
1972	Showing the Object Name on the Status Bar on page 70
1981	Easier Navigation in Multi-Page FAP Files on page 71
1997	Editing LOG Files from within Image Manager on page 73
1999	Including RDP Information on Group Reports on page 74
2003	Dragging and Dropping Objects on page 76
2005	Customizing the Library Manager Window on page 76
2015	Displaying Page Numbers on page 78
2017	Better Display when Resizing Lines on page 79
2019	Progress Bars During Report Generation on page 79
2023	Printing a Specific Page of a Form Set on page 80
2026	LBYPROC Utility Enhancements on page 81
2028	Miscellaneous Enhancements to the Studio Interface on page 81
2029	Miscellaneous Business Definition Manager Changes on page 82
2030	Miscellaneous Enhancements to Studio's Group Manager on page 83
2032	Miscellaneous Changes to Studio's Image Manager on page 84
2034	Miscellaneous Changes to Studio's Script Manager on page 84
2035	Miscellaneous Changes to Studio's Library Manager on page 85
2036	Miscellaneous Changes to Studio's Dictionary Managers on page 86
2038	Miscellaneous Changes to Studio's Conversion Manager on page 87
2039	Miscellaneous Changes to Studio's Reports Manager on page 87
2041	Miscellaneous Changes to Studio's Test Manager on page 88
2042	Miscellaneous Enhancements to Deployment Manager on page 88
2091	Importing CSV Files into an Extract Dictionary on page 98

Feature	For more information, see...
2095	Getting the JDName, JDLName, CodeDef, and CommentRecords Options on page 100
2111	Enhanced Help System on page 106
2127	New Look and Feel for Documaker Studio on page 113
2141	Creating FOR Files when Converting FAP Files on page 117
2142	Importing TGA Files into Your Extract Dictionary on page 119
2147	Using the New Font Usage Report on page 123
2152	Opening Documaker Studio Files on page 125
2156	Searching for DAL Scripts on page 126
2163	New Ways to Select Objects on page 129
2165	Enhanced AFP Conversions on page 131
2166	Copying the EDL Form Description into the Library Description Field on page 134
2168	Expanded Search Capability on page 136
2170	Creating an Image from an Area on page 136
2171	Using the Measurement Tool on page 137
Docutoolbox enhancements	
1140	Using the FAP2PDF Utility on page 16
1498	Purging Information from the Library Log on page 16
1633	Importing RTF Files on page 21
1990	Adding Code Page and Character Set Names to the FXR on page 72
2026	LBYPROC Utility Enhancements on page 81
2061	Using the New BDF2FDT Utility on page 89
2063	Converting EPS Files into LOG Files on page 90
2086	Enhanced Error Checking for the MRGCHK Utility on page 94
2087	Marking Objects as Pre-printed on page 94
2096	Using the New FONTLIST Utility on page 102
2107	Improved Support for PostScript Symbol Fonts on page 104

Feature	For more information, see...
2108	Generating DDL Files from DFD Files on page 104
2119	Using Mixed Mode AFP Datasets on page 110
2122	Support for QuarkXPress Version 7.0 on page 112
2183	Using the FDT2DB Utility's New Purge Parameter on page 139
Documaker Workstation enhancements	
1868	Automatically Re-Paginating Images on page 33
2067	Prompting for Search Criteria on page 91
2112	Mapping Alternate WIP Index Columns from Imported WIP Data on page 108
2113	Suppressing Duplicate Form Descriptions on page 108
2144	Controlling Pagination when Editing a Form Set on page 122
2146	Automatically Saving Form Sets in iDocumaker Workstation on page 122
2164	Using the New Complete and Exit Option on page 130
Printer and font enhancements	
1543	300 DPI Support for AFP Files on page 17
1864	Support Added to Vectors for the Pre-printed Option on page 23
1865	GDI Print Supports GenPrint Batch Printing to Files on page 24
2064	The AFP Loader Now Uses Character Set And Code Page Font Information on page 90
2068	Creating Smaller PostScript Output on page 91
2077	Suppressing LOG Files When Converting AFP to FAP or PDF on page 92
2080	Outputting AFP Character Set and Code Page Information on page 93
2085	Support for Version 1 Xerox FRM Files on page 94
2109	Enhanced Support for Color Logos on page 105
2116	Showing Field Information in PDF Files on page 109
2123	Defining the Content in the MultiFilePrint Log on page 112
2159	Loading Logos from the Library when Converting AFP or Metacode Files on page 128
2160	Paper Stock Support in the Metacode Print Driver on page 128

Feature	For more information, see...
2179	Using the New Font Cross-Reference (FXR) Files on page 138
Rules processing enhancements	
1916	Using the New ResetImageDimension Rule Options on page 63
2011	Using the New ProcessTriggers Rule on page 77
2045	New Default for the CheckNextRecip Option on page 88
2121	Making Banner Forms Compatible with Multi-file Print on page 110
2128	Using the New CheckZeroFontID Rule on page 114
2140	Truncating Fields on page 117
2150	Generating PDF417 Barcodes on page 124
2185	Using the New RunTriggers Rule on page 139
Miscellaneous enhancements	
1604	Adding Hypertext Links on page 18
1606	Improved Support for Color and Graphics on page 20
1859	Logging INI File Names and Options in the TRACE File on page 22
1872	Storing the WIP Index and WIP Data in Database Tables on page 36
1895	Generating the USPS Intelligent Mail Barcode (4-State Customer Bar Code) on page 60
1995	Using the Char and CharV DAL Functions on page 72
2002	Using Superscript and Subscript in Text Labels, Text Areas, and Variable Fields on page 74
2072	Specifying a DFD File for the Library Data Table on page 92
2073	Specifying a Catalog Table for Each Library on page 92
2088	Enhanced DAL Profile Output on page 96
2155	Defaults for the Module and PrintFunc Options on page 125
2167	Improved WIP Selection Performance During Batch Processing on page 135

CHAPTER 2

List of Features

Version 11.2 includes features and enhancements that improve the functionality and ease of use of our suite of products.

This document provides detailed information on the specific features and enhancements, listed in feature number order.

NOTE: You can find a [Summary of Features on page 1](#) which groups these features into functional categories, such as Archive or Documaker Studio.

OVERVIEW

The following table provides a list of the features included in the version 11.2 release and information about the type of license you must have to receive each feature.

License		Description
PPS		You must have a license for the Documaker Workstation/PPS product.
DAP		You must have a license for the DAP Development System.
RPS		You must have a license for Documaker Server.
iDMS		You must have a license for iDocumaker Workstation.

Feature	License	For more information, see...
1140	RPS	Using the FAP2PDF Utility on page 16
1498	RPS	Purging Information from the Library Log on page 16
1543	RPS	300 DPI Support for AFP Files on page 17
1604	RPS	Adding Hypertext Links on page 18
1606	DAP	Improved Support for Color and Graphics on page 20
1633	DAP	Importing RTF Files on page 21
1859	RPS	Logging INI File Names and Options in the TRACE File on page 22
1861	RPS	Moving Through the List of Fonts on page 22
1864	RPS	Support Added to Vectors for the Pre-printed Option on page 23
1865	RPS	GDI Print Supports GenPrint Batch Printing to Files on page 24
1866	RPS	Easier Handling of Overflow on page 25
1868	RPS	Automatically Re-Paginating Images on page 33
1869	RPS	Adding Security at the Resource Level on page 35
1872	RPS	Storing the WIP Index and WIP Data in Database Tables on page 36
1888	RPS	Extracting EDL Forms from a VLAM File on page 52
1891	RPS	Converting FAP and FOR Files into XML Files on page 54
1892	RPS	Using the New Image Usage Report on page 57

Feature	License	For more information, see...
1895	RPS	Generating the USPS Intelligent Mail Barcode (4-State Customer Bar Code) on page 60
1901	RPS	Using the New Undo/Redo Option on page 61
1906	RPS	Viewing Field Information on page 62
1911	RPS	Specifying Return Values on page 63
1916	RPS	Using the New ResetImageDimension Rule Options on page 63
1918	RPS	Retaining the Duplex Value when Converting Normalized Metacode Files on page 64
1932	RPS	Controlling Access to the Field Dictionary Manager on page 64
1937	RPS	Using the New Toolbar Buttons on page 65
1938	RPS	Docking and Automatically Hiding Windows on page 66
1950	RPS	Setting the Default Effective Date for New Resources on page 68
1966	RPS	Highlighting Selected Objects on page 69
1971	RPS	Using the Form Manager Navigation Tree on page 69
1972	RPS	Showing the Object Name on the Status Bar on page 70
1981	RPS	Easier Navigation in Multi-Page FAP Files on page 71
1990	RPS	Adding Code Page and Character Set Names to the FXR on page 72
1995	RPS	Using the Char and CharV DAL Functions on page 72
1997	RPS	Editing LOG Files from within Image Manager on page 73
1999	RPS	Including RDP Information on Group Reports on page 74
2002	RPS	Using Superscript and Subscript in Text Labels, Text Areas, and Variable Fields on page 74
2003	RPS	Dragging and Dropping Objects on page 76
2005	RPS	Customizing the Library Manager Window on page 76
2011	RPS	Using the New ProcessTriggers Rule on page 77
2015	RPS	Displaying Page Numbers on page 78

Feature	License	For more information, see...
2017	RPS	Better Display when Resizing Lines on page 79
2019	RPS	Progress Bars During Report Generation on page 79
2023	RPS	Printing a Specific Page of a Form Set on page 80
2026	RPS	LBYPROC Utility Enhancements on page 81
2028	RPS	Miscellaneous Enhancements to the Studio Interface on page 81
2029	RPS	Miscellaneous Business Definition Manager Changes on page 82
2030	RPS	Miscellaneous Enhancements to Studio's Group Manager on page 83
2032	RPS	Miscellaneous Changes to Studio's Image Manager on page 84
2034	RPS	Miscellaneous Changes to Studio's Script Manager on page 84
2035	RPS	Miscellaneous Changes to Studio's Library Manager on page 85
2036	RPS	Miscellaneous Changes to Studio's Dictionary Managers on page 86
2038	RPS	Miscellaneous Changes to Studio's Conversion Manager on page 87
2039	RPS	Miscellaneous Changes to Studio's Reports Manager on page 87
2041	RPS	Miscellaneous Changes to Studio's Test Manager on page 88
2042	RPS	Miscellaneous Enhancements to Deployment Manager on page 88
2045	RPS	New Default for the CheckNextRecip Option on page 88
2061	RPS	Using the New BDF2FDT Utility on page 89
2063	RPS	Converting EPS Files into LOG Files on page 90
2064	RPS	The AFP Loader Now Uses Character Set And Code Page Font Information on page 90
2067	RPS	Prompting for Search Criteria on page 91
2068	RPS	Creating Smaller PostScript Output on page 91

Feature	License	For more information, see...
2072	RPS	Specifying a DFD File for the Library Data Table on page 92
2073	RPS	Specifying a Catalog Table for Each Library on page 92
2077	RPS	Suppressing LOG Files When Converting AFP to FAP or PDF on page 92
2080	RPS	Outputting AFP Character Set and Code Page Information on page 93
2085	RPS	Support for Version 1 Xerox FRM Files on page 94
2086	RPS	Enhanced Error Checking for the MRGCHK Utility on page 94
2087	RPS	Marking Objects as Pre-printed on page 94
2088	RPS	Enhanced DAL Profile Output on page 96
2091	RPS	Importing CSV Files into an Extract Dictionary on page 98
2095	RPS	Getting the JDName, JDLName, CodeDef, and CommentRecords Options on page 100
2096	RPS	Using the New FONTLIST Utility on page 102
2107	RPS	Improved Support for PostScript Symbol Fonts on page 104
2108	RPS	Generating DDL Files from DFD Files on page 104
2109	RPS	Enhanced Support for Color Logos on page 105
2111	RPS	Enhanced Help System on page 106
2112	RPS	Mapping Alternate WIP Index Columns from Imported WIP Data on page 108
2113	RPS	Suppressing Duplicate Form Descriptions on page 108
2116	RPS	Showing Field Information in PDF Files on page 109
2119	RPS	Using Mixed Mode AFP Datasets on page 110
2121	RPS	Making Banner Forms Compatible with Multi-file Print on page 110
2122	RPS	Support for QuarkXPress Version 7.0 on page 112
2123	RPS	Defining the Content in the MultiFilePrint Log on page 112
2127	RPS	New Look and Feel for Documaker Studio on page 113

Feature	License	For more information, see...
2128	RPS	Using the New CheckZeroFontID Rule on page 114
2130	DAP	Additional Documanager Data Type Support in Documaker Server Archive on page 115
2131	RPS	Automatic Category Overrides on page 116
2140	RPS	Truncating Fields on page 117
2141	RPS	Creating FOR Files when Converting FAP Files on page 117
2142	RPS	Importing TGA Files into Your Extract Dictionary on page 119
2143	RPS	Paginating Forms Using DAL Functions on page 120
2144	RPS	Controlling Pagination when Editing a Form Set on page 122
2146	iDMS	Automatically Saving Form Sets in iDocumaker Workstation on page 122
2147	RPS	Using the New Font Usage Report on page 123
2150	RPS	Generating PDF417 Barcodes on page 124
2152	RPS	Opening Documaker Studio Files on page 125
2155	RPS	Defaults for the Module and PrintFunc Options on page 125
2156	RPS	Searching for DAL Scripts on page 126
2158	RPS	Using the New STRCompare Function on page 127
2159	RPS	Loading Logos from the Library when Converting AFP or Metacode Files on page 128
2160	RPS	Paper Stock Support in the Metacode Print Driver on page 128
2163	RPS	New Ways to Select Objects on page 129
2164	RPS	Using the New Complete and Exit Option on page 130
2165	RPS	Enhanced AFP Conversions on page 131
2166	RPS	Copying the EDL Form Description into the Library Description Field on page 134
2167	RPS	Improved WIP Selection Performance During Batch Processing on page 135
2168	RPS	Expanded Search Capability on page 136

Feature	License	For more information, see...
2170	RPS	Creating an Image from an Area on page 136
2171	RPS	Using the Measurement Tool on page 137
2179	RPS	Using the New Font Cross-Reference (FXR) Files on page 138
2183	RPS	Using the FDT2DB Utility's New Purge Parameter on page 139
2185	RPS	Using the New RunTriggers Rule on page 139

1140
RPS

USING THE FAP2PDF UTILITY

You can use the new FAP2PDF utility to convert FAP files into PDF files. To produce a PDF file from a FAP file, you must also supply an FXR file. The FXR file determines what fonts are used in the PDF files the utility produces.

The utility looks in the PDF printer control group in your FSISYS.INI file for options that tell it how to create the PDF file. These options let you include bookmarks, embed fonts, use internal fonts, and so on.

NOTE: You can use the /INI parameter to specify the INI file that contains the PDF printer control group and you can use the /P parameter to specify the actual name of the PDF printer control group. For information about the INI options you can use to customize PDF output, see Using the PDF Print Driver.

Syntax

FAP2PDF /I /X /INI /P /VF

Parameter	Description
/I	Enter the names of the FAP files you want to convert into PDF files. You can use wildcard characters (*). You do not have to specify the .FAP extension.
/X	Enter the name of the FXR file you want to use. The FXR file determines which fonts are used in the PDF file. You do not have to specify the .FXR extension.
/INI	(optional) Enter the name of the INI file that contains PDF settings. The default is FSISYS.INI.
/P	(optional) Enter the name of the PDF PrtType control group in the FSISYS.INI file. The default is PDF.
/VF	(optional) Include this parameter if you want to template fields.

If the FXR and INI files specify that you want the fonts embedded, the INI file must contain a FontLib option in the MasterResource control group that specifies the directory of the TrueType or PostScript fonts.

NOTE: To see a list of the FAP2PDF parameters, run the utility with no parameters.

1498
RPS

PURGING INFORMATION FROM THE LIBRARY LOG

You can now purge information from Studio's library log. To do so, open a library in the Library Manager and then select History from the Library menu.

Use the Filter Values to select the kind of information you want to purge. For instance, you could select only log entries that pertain to FAP files or only log entries about resources that were unlocked.

Then specify the range of log entries to purge:

Log Date Options	
<input checked="" type="checkbox"/> Purge Before	10/28/2006
<input type="checkbox"/> Purge Older Than	0
Purge Now	Purge Now

- You can enter the date before which you want log entries purged. For instance, if you want to purge entries created before October 28, 2006, you would specify 10/28/06 in the Purge Before field either by entering the date or by selecting it from the calendar that appears when you click in the field.
- You can specify the number of days before which you want log entries purged. For instance, if you want entries purged that are 30 days old or older, enter 30 or use the spin button to select 30 in the Purge Older Than field.

NOTE: You can specify either the date or the number of days, but not both.

Click Purge Now to purge the log entries. Studio lets you confirm the action before it deletes the entries. Keep in mind this *permanently deletes* log entries from the library log.

1543
RPS

300 DPI SUPPORT FOR AFP FILES

Documaker Studio and Image Editor can now produce 300 DPI output for Documaker AFP print streams and normalized AFP files. For Documerge customers who compose AFP forms at 300 dots per inch (DPI) via Printcommander 5.x/6.x, you no longer have to convert your AFP forms to 240 DPI to use Image Editor.

The Resolution option in your AFP printer control group controls the resolution of the coordinates used to produce Documaker AFP print streams and normalized AFP files for Documerge. The Resolution option also controls the resolution of the bitmap data produced when using inline logos.

```
< PrtType:AFP >
Resolution = 240
```

Option	Description
--------	-------------

Resolution	You can enter 240, 300, or 600 to indicate the resolution in dots per inch (DPI). The default is 240.
------------	---

The FAP2OVL utility has also been enhanced to support producing AFP overlays using 240, 300, or 600 DPI for coordinates and bitmap data produced when using inline logos. Use the new /R parameter to specify the resolution you want.

Parameter	Description
/R	Enter 240, 300, or 600 to indicate the resolution in dots per inch (DPI). The default is 240.

Documaker Server and Documaker Studio can produce AFP output (print streams and normalized AFP files) using either a 240 or 300 DPI coordinate system.

For reference, FAP files use a 2400 DPI coordinate system, so to position an object, such as a line or piece of text, at 1 inch by 1 inch using FAP coordinates, you would place the object at 2400 units by 2400 units.

To position an object at 1 inch by 1 inch using 300 DPI coordinates, you would place the object at 300 units by 300 units.

To position an object at 1 inch by 1 inch using 240 DPI coordinates, you would place the object at 240 units by 240 units.

In all three cases, the object is placed at 1 inch by 1 inch. The only difference is the DPI setting you use for positioning. The printed output of AFP output is identical regardless of the coordinate system used.

1604
RPS

ADDING HYPERTEXT LINKS

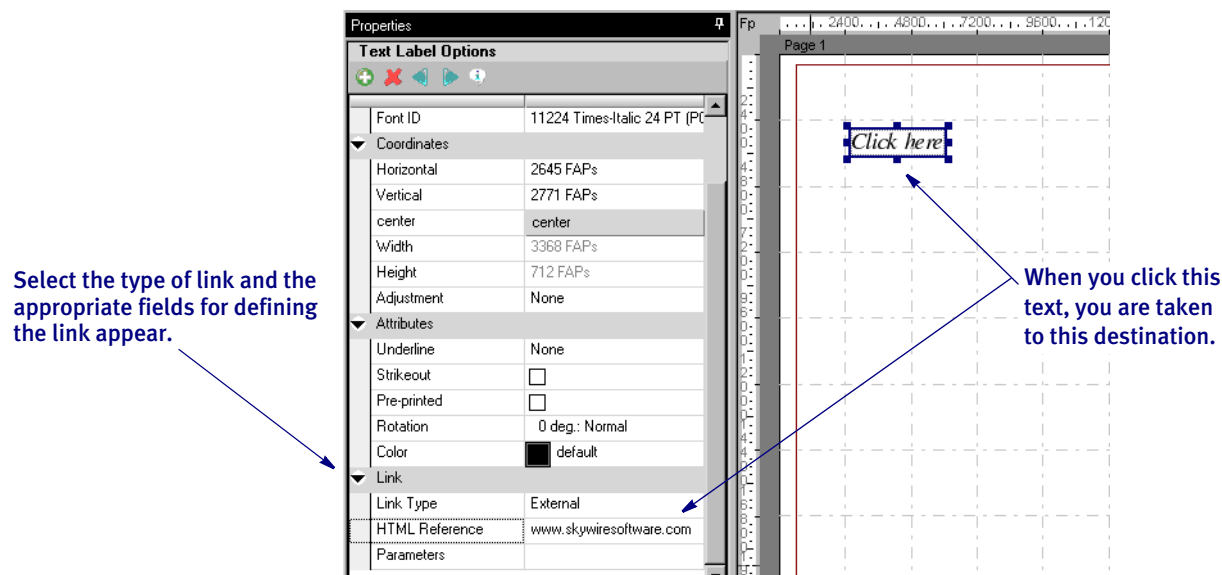
You can now produce PDF or HTML output that contains hypertext links that will launch a web browser and open a specified web site. You can create hypertext links in these types of objects:

- Text labels
- Variable fields
- Logos

NOTE: Place the hypertext link at the actual field location and not at a reference to that field. For instance, if you define a variable field with a hypertext link and embed a reference to that field within a text area, the reference will not indicate the link is there.

Also keep in mind that links are assigned as an attribute of a field. The data placed in the field is not the target of the link, but rather the data you click on to go to that target.

This example shows a text label. You use the text label's Link properties to define the link.



Once you select *External* in the Link Type field, other fields appear so you can define the link.

Field	Description
Link Type	Select External to set up a link to a web page.
HTML Reference	Enter the address of the web page. Here is an example: <code>http://www.docucorp.com</code>
Parameters	Use this field when producing HTML output to specify additional parameters to an HREF type link, such as a target frame or mouseover behavior. This example causes a web page to open in a new browser window: <code>target="new"</code>

NOTE: If you have the PDF Print Driver defined as your printer, you will see options appropriate for setting up a PDF link. For more information about licensing the PDF Print Driver, contact your sales representative.

If the example above were used on a text label containing the text *Click Here*, the system would generate the following command in an HTML print stream:

```
<a href="http://www.docucorp.com" target="new">Click Here</a>
```

When viewed in a browser, the HTML print stream would contain a link that appears as follows:

[Click Here](http://www.docucorp.com)

You can find more information on HTML hypertext links at:

<http://www.w3.org/TR/html401/>

1606
DAP

IMPROVED SUPPORT FOR COLOR AND GRAPHICS

Version 11.2 includes several enhancements that affect support for color and graphics. These include:

- The PCL print driver now supports bitmap compression
- The PostScript print driver now supports bitmap compression
- The PostScript print driver now 4-bit and 8-bit color bitmaps

NOTE: Compression is enabled by default in both the PCL and PostScript print drivers. To disable compression, add the following INI option to the PCL and PostScript printer control group:

```
< PrtType:XXX >  
Compression = No
```

The PostScript print driver now supports 4-bit and 8-bit color bitmaps. It previously only supported monochrome and 24-bit bitmaps. Color bitmaps are compressed in JPEG format.

Monochrome bitmaps are compressed using Run Length Encoding (RLE) compression. If compression or color is disabled, 4-bit and 8-bit color bitmaps are printed as monochrome bitmaps. For compatibility with previous releases, 24-bit color bitmaps are printed in color when compression is disabled and color is enabled.

PostScript print streams with bitmap compression are often smaller and may be produced faster than PostScript print streams without bitmap compression. PostScript print streams with compressed multicolor bitmaps will see the greatest reduction in terms of file size and time to produce.

The 4-bit and 8-bit color bitmaps printed in color with compression will likely produce larger print streams than 4-bit and 8-bit color bitmaps which have been converted to monochrome (black and white) bitmaps.

Keep in mind:

- For any bitmap to print in color, you must make sure the bitmap (LOG) is marked as *Print in Color* in the FAP file. Also make sure you set the SendColor option to Yes in the PCL or PostScript printer control group before printing.
- When using Forms Integrity Manager (FIM) to compare a new (version 11.2) PostScript print stream with bitmap compression against an older PostScript print stream without bitmap compression, FIM will report that some bitmaps are not identical. Older PostScript print streams without bitmap compression generated the bitmap data in multiple streams while the newer compressed bitmaps are always generated within a single stream. In this case, FIM will report the older print streams contains multiple *Overlay Images* entries while the new print streams contain a single *Overlay Images* entry. Also, FIM may report differences in some attributes (height, width, raster size, and so on) of *Overlay Images* and *Variable Images* due to differences in how bitmaps are emitted.

1633
DAP

IMPORTING RTF FILES

The RTF import feature, which converts RTF files into Docucorp's FAP file format, includes the following enhancements which result in greater fidelity.

- You can now import polyline or vector drawings.
- Objects anchored to a paragraph are positioned correctly.
- Improved font matching more closely resembles the original style and size.
- Unicode support now includes paragraphs containing mixed single-byte and multi-byte characters.
- Hyphenation settings are now retained.
- You can now convert bitmaps in the original document into external Documaker logo (LOG) references.
- Using the RTF2FAP utility, you can create form (FOR) files for Documaker Studio. Separate images are created for headers, bodies, and footers.

RTF2FAP utility

These changes include two new parameters for the RTF2FAP utility:

Parameter	Description
/L	Include this parameter to create LOG files for all bitmap images in the input file.
/F	Include this parameter and a file name to create a FOR file for use with Documaker Studio. If you omit the file name, the utility appends the FOR extension to the input file name.

Image Editor

When importing RTF files from the Image Editor, the system uses the FXR file you currently have loaded. No output file is created until you save the FAP file. One multi-page FAP file is created rather than a FAP file for each page. A Documaker Studio form file (FOR) is not created.

Limitations

There are some limitations to the RTF importer.

- If an bitmap image does not specify a resolution, the system defaults to 300 DPI. This may result in logos being sized incorrectly. The system tries to calculate a resolution based on the output size specified by the RTF file, but this may not work in every instance.
- The *linkstyles* control word causes Microsoft Word to alter a document (fonts) according to user-specific styles held in the Normal.dot template. Since Docucorp software cannot interpret the Normal.dot file — as it would vary according to the user — the *linkstyles* control word cannot be supported. If the imported FAP file has fonts that are significantly different from those in the original RTF file, this may be the cause.
- Docucorp software does not support z-ordering, the layering of objects one over the other. Documents which rely on z-ordering appear differently when imported.

- The *charscalex* and *charscaley* control words cause individual characters to be scaled horizontally or vertically, respectively. The system does not support such functionality, so these control words are not supported.
- Document layout is heavily dependent on choice of fonts. The system tries to match the fonts in the source document, but substitution can affect the layout. The best way to avoid layout changes is to include all used fonts in your font cross-reference (FXR) file. Be aware of font copyright restrictions when doing so.
- When you use the RTF import feature to convert rich text pasted into a FAP text area, the system has no information about the margins in the FAP file. This can cause some objects to shift.
- Docucorp software does not include a table object, so RTF tables are converted to columnar data. If, however, the table is embedded in a column, this results in nested columns (columns within a column), which is not supported.
- Merged table cells are not supported.

1859
RPS

LOGGING INI FILE NAMES AND OPTIONS IN THE TRACE FILE

Now you can log INI file names and options in the TRACE file during GenTrn, GenData, GenPrint, GenArc, and Documaker Studio processing.

To turn on the logging of INI file names and options, include these INI options:

```
< Debug_Switches >
  Enable_Debug_Options = Yes
  INILib               = Yes
```

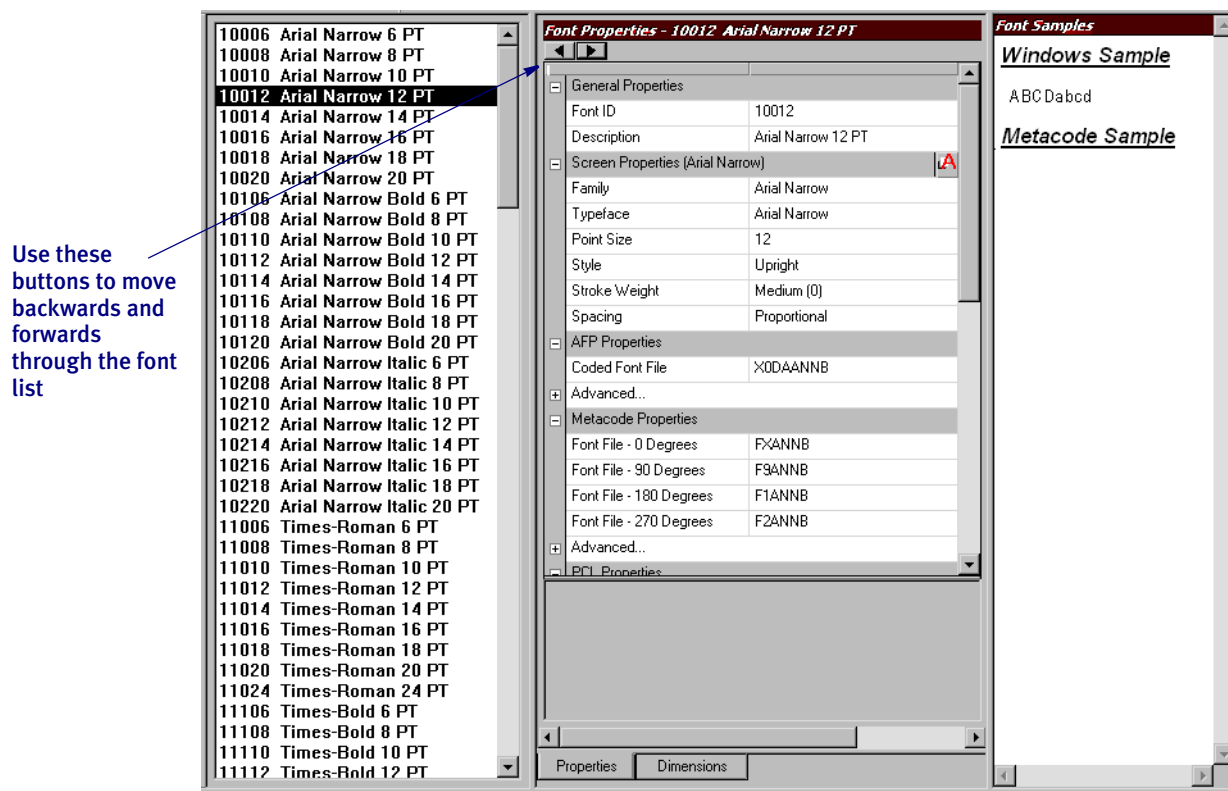
For the GenTrn, GenData, GenPrint, and GenArc programs, you can include the /L command line parameter to log these file names and options in the TRACE file.

NOTE: Logging the INI file names and options in the TRACE file replaces the writing of the INI file names and options to the LOGFILE as was done in prior releases.

1861
RPS

MOVING THROUGH THE LIST OF FONTS

You can use the new forward and backward arrow buttons on the Font Description window in Studio's Font Manager to more easily navigate from one font to the next.



1864
RPS

SUPPORT ADDED TO VECTORS FOR THE PRE-PRINTED OPTION

You can now enable the Pre-Printed option to tell the system that the vector drawing should not print when using pre-printed paper.

NOTE: The Pre-Printed attribute is assigned to a given vector via the property definitions in the Image Manager. The Pre-Printed paper option is an option you specify at print time and tells the system whether it should print the pre-printed items.

This change affects any utility or program that imports, converts, or outputs vectors. This includes the PCL and Quark conversions to FAP files, as well as all printer output drivers when printing vectors.

This lets you design a form view that looks like the final document without having to print the vector drawing — because that object is pre-printed.

NOTE: Keep in mind you cannot upgrade to get this feature and then create FAP files with vectors that will work in an earlier version of the software. Also keep in mind that the PDF Print Driver ignores the pre-printed flag for all objects.

1865
RPS

GENPRINT BATCH PRINTING TO FILES

You can now use the GDI print driver to print to a file. By adding the new PrintToFile option in your GDI printer control group, you can direct output to the path and file you specify — equivalent to checking the Print to File field on the Print window.

```
< PtrType:GDI >  
PrintToFile = Yes
```

Option	Description
--------	-------------

PrintToFile	Enter Yes to have the GDI print driver use the Port options as the output printfile names for each batch when running GenPrint. The default is No.
-------------	--

In the GenPrint program, output printfile names for each batch are specified using the Port INI option. When you use the GenPrint program with most Documaker print drivers, the Port option determines the name of the print stream created for each batch.

Normally, the GDI print driver prints directly to a Windows print driver and does not create files written to disk. By setting the PrintToFile option to Yes in your GDI printer control group, the GDI print driver creates a print stream for each batch based on the names specified in the Port options — just like the other Documaker print drivers.

Because the Documaker GDI print driver is not designed for batch print, these additional GDI print options are recommended when you set the PrintToFile to Yes:

```
< PtrType:GDI >  
...  
SuppressDialog = Yes  
GDIDevice = (Windows printer name)  
FullSupport = Yes
```

Option	Description
--------	-------------

SuppressDialog	Enter Yes to suppress the Windows Print window from appearing.
GDIDevice	Enter the name of the Windows print driver you want to use.
FullSupport	Enter Yes to tell the Windows driver to fully support duplexing, tray selection, and so on.

This feature is limited to using the GDI driver with GenPrint (multi-step batch print) to produce output print files and is limited to simple GenPrint (batch print) environments.

Keep in mind that all normal GDI print limitations (fidelity, tray selection, duplexing, and so on) apply, plus the following:

- Banner page processing may not work
- Cannot use the SetDeviceName and BreakBatch DAL functions
- Callback functions may not work

- Single step processing does not work correctly (all transactions are printed to a single file)
- Multiple driver router may not work
- Printing from the Image Editor will probably work but the Device setting will be used to create the file
- Printing from PPS may not work correctly
- Printing to fax drivers, email drivers, and so on may not work
- Other types of print or print features not previously discussed may not work

In other words, trying to use Print to File option with anything except GenPrint running in a simple batch mode using a normal Windows print driver is not supported.

1866
RPS

EASIER HANDLING OF OVERFLOW

The process of mapping fields on repeating page segments — known as overflow — has been improved. This majority of the overflow situations you will typically encounter are now easier to implement.

NOTE: Your existing overflow mappings are not affected. In fact, the prior method may prove useful to handle some specific and more complicated overflow situations, such as when the variables are not being mapped onto repeating page segments of the same name.

This new method of handling overflow is based upon the assumption that there is usually a fixed relationship between the data occurrence you want to map and the form and/or page segment (FAP image) occurrence where the fields are defined. All you have to do is set a few field and image options to define the relationship for Studio.

Keep in mind that there are two types of overflow situations, sometimes referred to as:

- Regular overflow
- Forced overflow

The main difference is in how many occurrences of similar data appear on a single page segment.

In regular overflow, there is one occurrence of each field on the form or page segment. For instance, you might have a page segment that represents an invoice line on a form. Here is an example:

Page segment 1						
Classification	Code No	Prem Basis	Rate	Prem/Ops	Commiss	Extended

In this example, seven fields are defined on the page segment. This might represent a single row of information from the input stream you are mapping from. Using this type of page segment, you would trigger one page segment for each row of data found in the extract file. For example, if your extract contained 12 rows of data, you would trigger 12 copies of this page segment. Each page segment you include will need to map a subsequent occurrence of the data.

In forced overflow, there are multiple occurrences of similar data to be mapped onto the page segment. Here is an example:

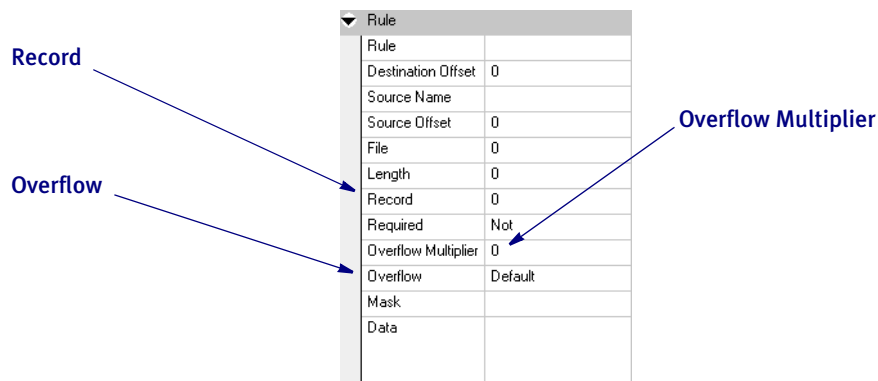
Page segment 1						
Classification1	Code No1	Prem Basis1	Rate1	Prem/Ops1	Commiss1	Extended1
Classification2	Code No2	Prem Basis2	Rate2	Prem/Ops2	Commiss2	Extended2
Classification3	Code No3	Prem Basis3	Rate3	Prem/Ops3	Commiss3	Extended3
Classification4	Code No4	Prem Basis4	Rate4	Prem/Ops4	Commiss4	Extended4
Classification5	Code No5	Prem Basis5	Rate5	Prem/Ops5	Commiss5	Extended5

Although similar in layout, in this example you see there are five pre-defined rows of fields on the page segment. For each page segment you trigger, you will be able to show five additional rows of data.

Notice that the names of the fields differ slightly on each row. Fields on a segment must have a unique name, but the underlying mapping of each similarly named field can be the same. Using the previous example of 12 rows of data in the extract file, you would trigger three occurrences of this page segment to hold that data. The last page segment would only map the first two rows of data and the remaining three rows would be empty.

Keeping these overflow examples in mind, let's look at how this feature lets you define the field mappings for these segments.

At the field level, you must tell Studio that repeating data elements will be mapped into the field's location using the Overflow, Overflow Multiplier, and Record fields.



Field	Description																		
Record	<p>Use this field in the forced overflow situation.</p> <p>Studio adds the value you enter to the current overflow calculation to reach the correct occurrence of the data to retrieve from the source file.</p> <p>If you enter 1 (or 0), Studio uses the occurrence calculation without modification — it uses the first record it finds.</p> <p>If you enter 2 or higher, it tells Studio that you want to reach the second occurrence value at the current calculation and so on.</p> <p>The exact overflow calculation is determined by using the Overflow Multiplier and Overflow fields.</p> <p>The default is zero (0).</p>																		
Overflow Multiplier	<p>When using forced overflow, there are multiple occurrences of the same data on the page segment.</p> <p>Use this field to indicate the number of times the field is repeated on the page segment. Enter a value greater than 1 to tell Studio you require forced overflow calculations.</p> <p>The resulting calculation is then added to the Record field to determine the exact overflow occurrence you are trying to map.</p> <p>The default is zero (0).</p>																		
Overflow	<p>Use this field to indicate that overflow is required.</p> <p>This applies to both regular and forced overflow. The option you select here defines how the occurrence will be calculated.</p> <div data-bbox="680 1026 1104 1220"> <table> <tr><td>Record</td><td>1</td></tr> <tr><td>Required</td><td>Not</td></tr> <tr><td>Overflow Multiplier</td><td>0</td></tr> <tr><td>Overflow</td><td>Default</td></tr> <tr><td>Mask</td><td>Default</td></tr> <tr><td>Data</td><td>Form</td></tr> <tr><td></td><td>Formset (Image)</td></tr> <tr><td></td><td>Image</td></tr> <tr><td></td><td>None</td></tr> </table> </div> <ul style="list-style-type: none"> — Default - This is the default setting. A value of Default means that this field is subject to the image-level definition for this Overflow value. For example, if you select Default and the image properties define anything other than None, Studio uses the image-level overflow definition. — Form - Choose Form if this field appears on a repeating form. This implies there is a different form for each instance of the data. — Formset (Image) - Choose Formset (Image) if the image page segment that contains the fields is repeated within the document, but is not limited to the same form. This means counting the matching image names from the start of the document, without regard to the actual form, determines the value to use in the overflow occurrence calculation. — Image - Choose Image when the repeating page segments are included on the same form. This means the count of the image within the parent form is used in calculating the correct overflow occurrence. — None - Choose None if you want to exclude this field from overflow calculations. This tells Studio that you will handle this field's mapping independent from any overflow that might be specified at the image page segment level. You do not have to set the field Overflow value to None if the image Overflow property is set to None. <p>The default is Default.</p>	Record	1	Required	Not	Overflow Multiplier	0	Overflow	Default	Mask	Default	Data	Form		Formset (Image)		Image		None
Record	1																		
Required	Not																		
Overflow Multiplier	0																		
Overflow	Default																		
Mask	Default																		
Data	Form																		
	Formset (Image)																		
	Image																		
	None																		

If you use the defaults for these fields at the field level, these fields inherit what is defined at the image level. If you choose different settings at the field level, the system will use your choices instead of the image level settings. So, unless you are mixing overflow and non-overflow variables on an image, you can just accept the defaults for the fields. If, however, you are using forced overflow, you do have to define the *record member* which is added in addition to any overflow occurrence calculated.

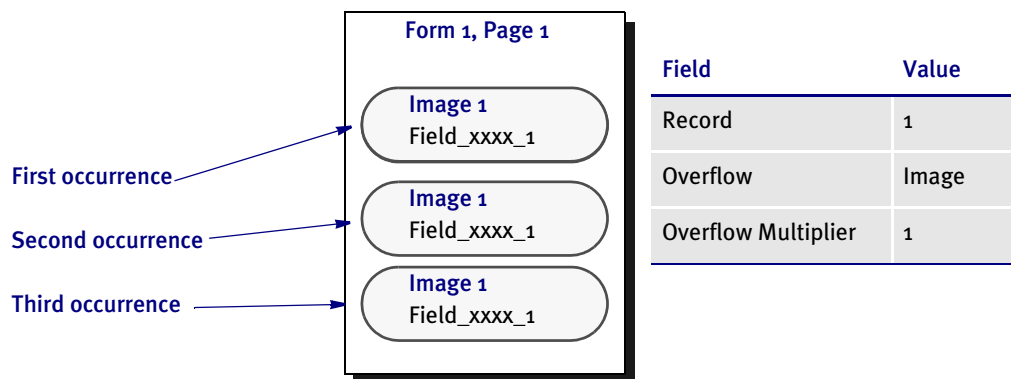
NOTE: If you decide to switch to the new method, remember to remove the old overflow rules from your AFGJOB.JDT file if you want to see an improvement in performance.

Overflow Examples

Here are some examples of how you can use this new overflow method.

Image-level overflow

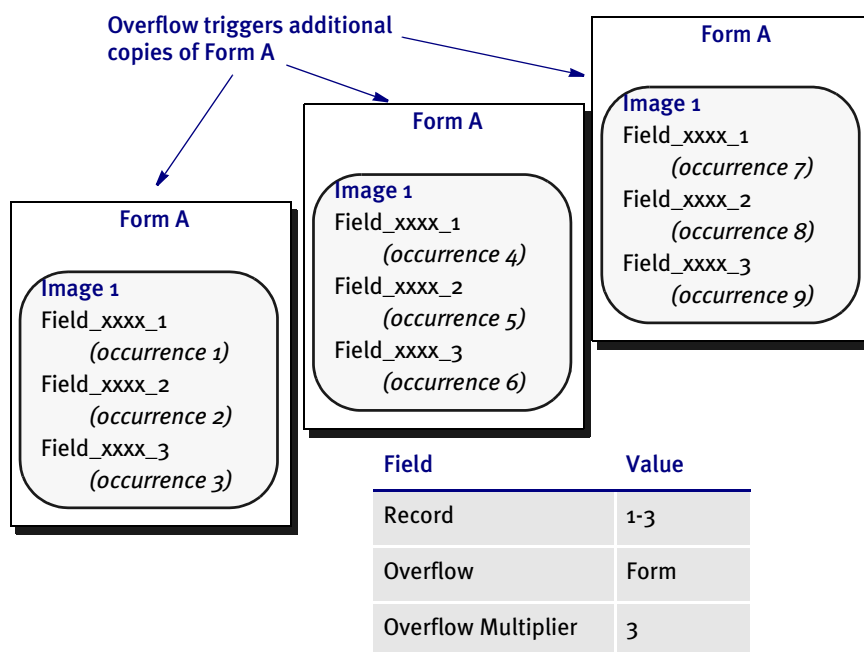
In this example of image-level overflow, Field_xxx_1 repeats on Image_1.



If you have additional occurrences, the system would create Form 1, Page 2 and so on until all the occurrences are handled.

Form-level overflow

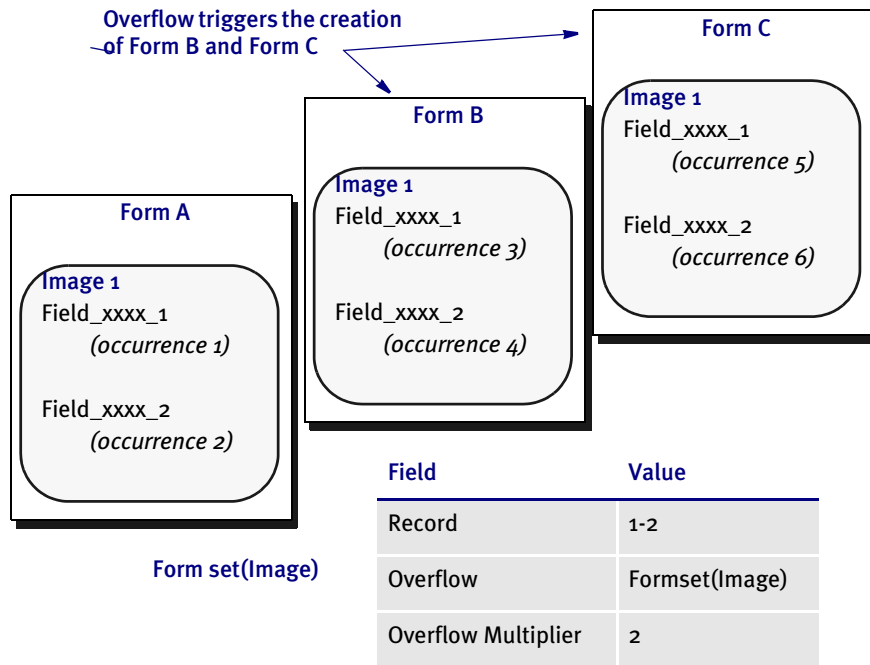
In this example of form-level overflow, the fields that comprise Image 1 (Field_xxx_1, Field_xxx_2, and Field_xxx_3) repeat three times, triggering three copies of Form A.



If you have additional occurrences, the system would create additional copies of Form A until all the occurrences are handled.

**Form set (Image)-level
overflow**

In this example of form set(image)-level overflow, Image_1, which contains Field_xxx_1 and Field_xxx_2, repeats three times, triggering Forms A, B, and C.



If you have additional occurrences, you must set up forms to handle those additional occurrences.

New User Functions

In addition, version 11.2 includes these new user functions:

Function	Description
GetImage	Use this function to return the image's sequence number of like named images within the current form. For compatibility with the field rule record functionality, all counts are zero (0) based.
GetForm	Use this function to return the form's sequence number of like named forms within the form set.
GetFormSetImage	Use this function to return the image's sequence number of like named images within the form set.

If you are using XML extract files, you can use these new user functions to set up incrementers at the form and image level. Otherwise, Studio handles the use of these functions for you when you enter overflow information on the properties window.

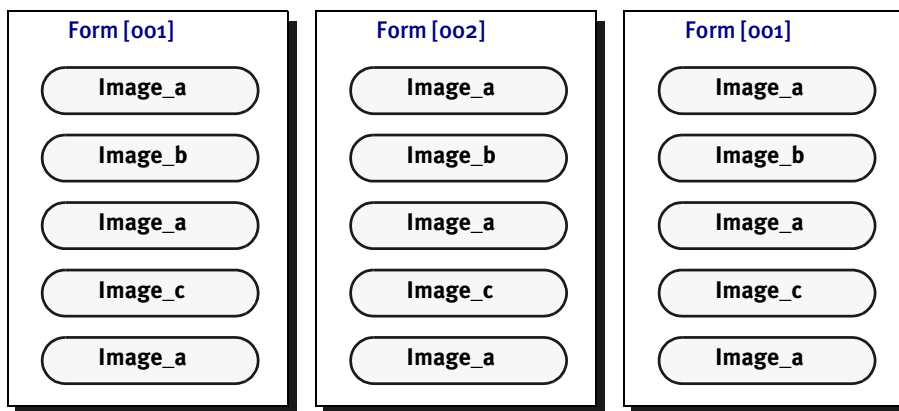
NOTE: These functions correspond to the options you can choose for the Overflow field: Image, Form, Formset(Image).

You can assign these new user functions to fields. There are two additional options when assigning them to fields:

Option	Description
None	Choose None to indicate the field does not use overflow.
Default	Choose Default to use overflow assigned at the image level.

In the Form manager, you can assign overflow at the image level. The three user functions and the None option are available to the image. Any function set for an image will be used on all overflow fields that are set to Default.

Keep in mind that the value for @GetImage, @GetForm, and @GetFormSet, is not the image or form sequence number, but the image or form sequence number for images or forms with the same name. For example, if you have a form set with these forms:



These values are returned when using the @GetImage (“Image”), @GetFormsetImage (“Formset(Image)”) and @GetForm:

Form	Image	GetImage	GetFormsetImage	GetForm
001				
	a	1	1	1
	b	1	1	1
	a	2	2	1
	c	1	1	1
	a	3	3	1
002				
	a	1	4	1
	b	1	2	1
	a	2	5	1
	c	1	2	1
	a	3	6	1
001				
	a	1	7	2
	b	1	3	2
	a	2	8	2
	c	1	3	2
	a	3	9	2

The RunTriggers rule evaluates the form trigger, if the form is triggered. The rule then triggers each of the image triggers before it evaluates the next form trigger. This lets you do regressive form triggers from the image level.

These examples show the rule data entry for an XML file:

```
!/status/summary/staff[**@GETIMAGE**]/inProcess
```

This returns the *inProcess* data element for the X occurrence of the staff section where X is equal to the image’s sequence number in the form (zero based) plus the field rule record count.

```
!/status/summary/staff[**@GETFORMSETIMAGE**]/assigned
```

This returns the *assigned* data element for the X occurrence of the staff section where X is equal to the image’s sequence number in the form set (zero based) plus the field rule record count.


```
!/status/summary/staff[**@GETFORM**]/onHold
```

This returns the *onHold* data element for the X occurrence of the staff section where X is equal to the form's sequence number (zero based) plus the field rule record count.

```
!/status/summary[**@GETFORM**]/staff[**@GETIMAGE**]/closed
```

This returns the *closed* data element for the X occurrence of the summary section where X is equal to the form's sequence number (zero based) plus the field rule record count and the Y occurrence of the staff section where Y is equal to the image is equal to the image's sequence number in the form (zero based) plus the field rule record count.

Overflow Variables

With Documaker Studio, overflow variables that require only a single increment per use are now automatically created for you. You no longer have to declare an overflow variable for your form sets in the AFGJOB.JDT file.

For example, you can now omit these lines from your AFGJOB.JDT file:

```
;SetOvFlwSym;1;DATI0OVF,QDATI0,1;
;SetOvFlwSym;1;MNUM1OVF,QMVNUM,4;
;SetOvFlwSym;1;AGDTDATAOVF,QAGDTD,1;
;SetOvFlwSym;1;CGDECBDOVF,QCGDCB,1;
```

Each of these lines declares a simple overflow variable with a single increment step. When using Studio, these variables are automatically created if you omit them from the AFGJOB.JDT file.

Keep in mind that if you are using the legacy way of handling overflow for any situation, you must still include the SetOvFlwSym rule for those situations.

If *all* of your overflow situations can be handled using the new overflow methodology, also remove the InitOvFlw and ResetOvFlw rules from the AFGJOB.JDT file. These rules are only necessary if you have an overflow situation that uses the legacy way of handling overflow. For instance, even if you are only using the legacy way in one scenario, you must still include the SetOvFlwSym, InitOvFlw, and ResetOvFlw rules in your AFGJOB.JDT file.

NOTE: You can also omit the IncOvSym image-level rule if you are using the new overflow methodology described in this feature.

1868
RPS

AUTOMATICALLY RE-PAGINATING IMAGES

Documaker Workstation now better handles pages comprised of multiple floating images by automatically re-paginating the form set when an image changes size.

NOTE: This applies to both Documaker Workstation and the WIP Edit plug-in. Documaker Server waits until the end of its processing to handle pagination.

Previously, Documaker Workstation did not re-paginate a form when a page consisted of multiple images and one of those images changed size. Now, if a text area causes an image to grow or shrink — due to embedded fields changing or because it was a multi-line text field — the form that contains the image automatically re-paginates.

If you designed the form using Documaker Studio, the positioning information you established is reapplied to the form each time changes in a text area cause the dimensions of an image to change. You can, however, use the AutoPagination option to disable automatic re-pagination:

```
< Control >  
    AutoPagination = Yes
```

Option	Description
AutoPagination	Enter No if you do not want the system to automatically re-paginate when image dimensions change. The default is Yes.

Keep in mind...

- The system will honor positioning information designed into the form via Documaker Studio. Positioning information is stored via SetOrigin rules.
- If a form consists of multiple images on a page, but those images comprise more space than defined for the page size, the system automatically paginates that page and moves the images that did not fit to a new page.
- If an image grows to push another image such that its positioning rule causes it to encroach on a defined footer or the bottom of the page, that image is moved to the next page and the entire form will have the SetOrigin rules reapplied.
- When designing a form, avoid having a footer image that uses a relative position. This ultimately means there can only be one image on the page that is not a header or footer. Footers, typically should be placed using a rule that makes sure it has a relationship to the bottom of the page.
- When images shrink (due to text area shrinking) an image from the next page may be brought back to the current page. In other words, images can not only flow to the next page, but they can come back when space allows.
- Remember that the positioning (SetOrigin) defined in the form is applied. So although there may appear to be a space large enough to hold an image, you also have to account for any additional adjustments applied by the SetOrigin rules.

For example, suppose you have an image that is 2 inches in height, but the SetOrigin rule for that image specifies a relative placement 1/2 inch from the previous image. In this case, 2 1/2 inches of space is required for that image to fit on the page. If there is less than 2 1/2 inches remaining before encountering a footer image or the bottom of the page layout, then that image will move to the next page.

- Images can only flow to and from pages that were created during pagination. If a page was specifically designed into the form via Documaker Studio, then no images will move onto that page from a prior page. Images can only move to or from pages that were created by overflowing their defined page.

1869
RPS

ADDING SECURITY AT THE RESOURCE LEVEL

You can now secure resources in Library Manager so only the users you authorize can check them out. For instance, this lets you control who can change a signature logo (LOG) file.

Studio provides broad categories for security, such as access by file type (BDF, GRP, FOR, FAP, and so on). For instance you can give a user ID full, access, no access, or view-only access to all forms and images. You may, however, have resources that need specific restriction, such as a need to restrict access to LOG files to specific individuals, or groups if you are using the LDAP model.

This enhancement lets you restrict access to individual library resources within a given type. To expand on the first example, suppose you have logos (LOG) in your library that represent signatures. You might secure those resources by restricting them to a particular user or group of users when using Lightweight Directory Access Protocol (LDAP) security model. That way, even though other users could access other LOG files, only specific individuals would have the ability to modify the LOG files that contain signatures.

NOTE: If you are not using the LDAP model, an ID represents a single user. In the LDAP model, the ID could represent a single user or a group of users.

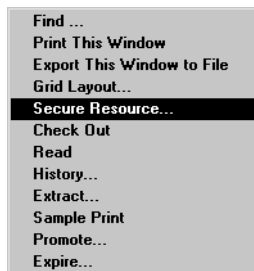
Here is another example within the LDAP security model. Suppose you have two groups of developers sharing the same library. You want them to share the library because all the business runs from a central location. You may, however, want to make sure one group is not accidentally checking out or changing forms that belong to the other group and vice-versa.

For instance, group one might be distinguished by what line of business that they work on, such as Life and Property in an insurance scenario. To meet this need, you could secure the appropriate resources using an ID that represents the *group* to which individual users belong. Group affiliation is a feature of LDAP security Studio can use.

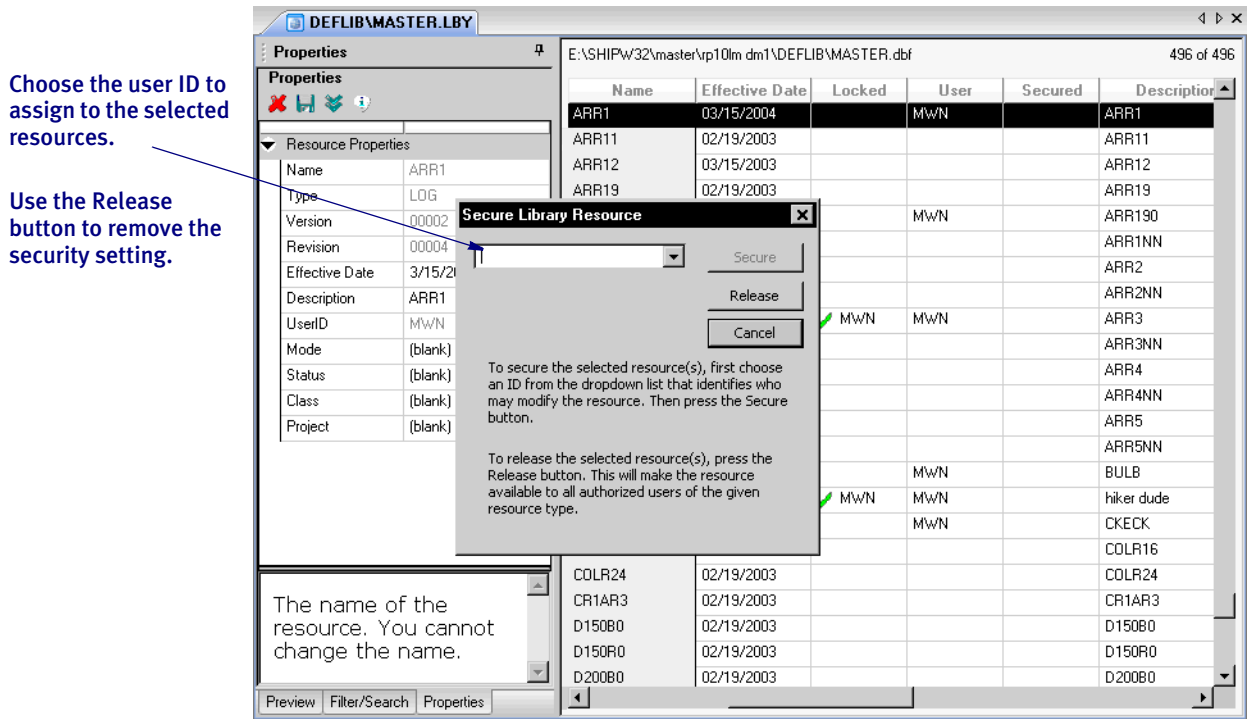
So, when a user logs into a system that belongs to specific group, that user can check out the resources for that group. But if the user does not belong to the group identified on the secured resource, the user cannot check out the resource.

To secure resources, open Library manager and highlight the resources you want to secure. Then right click and choose the Secure Resource option.

After you highlight the resources you want to secure, right click and choose this option.



The Secure Library Resource window appears:



Choose the user ID you want to assign to the selected resources.

1872
RPS

STORING THE WIP INDEX AND WIP DATA IN DATABASE TABLES

Documaker software now lets you store WIP data in a database to enhance security and the commit and rollback process. This also enhances performance when using multiple keys to access transactions in intra- and internet applications. To implement this enhancement, changes were made that affect Documaker Server, Documaker Workstation, Shared Objects, and the WIP Edit plug-in.

NOTE: The WIPDATA.DFD file is a new file.

Docucorp recommends that you only use uppercase for table and column names when storing information in a database. For instance, avoid CustomerName, Customername, or customername and instead use CUSTOMERNAME.

Database management systems (DBMS) vary in how they handle case issues so it is best to standardize on uppercase. With version 11.2, all column names must be in uppercase.

The standard WIP system uses an xBase index and a flat file (NAFILE, POLFILE) format. For WIP transactions Documaker uses a default WIP.DFD when placing transactions into the default xBase system.

When you change database systems, to implement this new feature enhancement you will need a physical WIP.DFD and WIPDATA.DFD file.

The files will need to be modified to add a UNIQUE_ID field.

You can install our sample RPEX1 master resource library (MRL) which includes utilities to create WIP.DFD and WIPDATA.DFD files for MySQL, Microsoft SQL (MS SQL), DB2, and Oracle.

Similarly, Docucorp includes the WIP.DFD and WIPDATA.DFD files for each database type in this document. You can copy and paste to a text file, call them WIP.DFD and WIPDATA.DFD and place them into your MRL's DefLib directory.

You also need to establish connection with your database and set up database files for WIP to talk to.

You can modify your INI files to automatically create the tables and fields for you.

Or, you can use the DFD2DDL utility (DFD2DDLW.EXE) to read the WIP.DFD and WIPDATA.DFD files and create DDL files for your database administrator to use in setting up the tables.

There are INI file changes you must make to implement this feature. These changes are discussed in detail below. You will also find instructions for running a DAL script that can update the INI files for you. A listing of the DAL script is included in this document. The actual DAL script (INI4WIP.DAL) is included in the sample RPEX1 MRL in the rpex1\deflib directory.

You can now store WIP indexes and WIP data in the following types of databases using the following ODBC drivers:

Databases	DB2 version 7.2.2. on AIX Oracle version 8.1.7 on Solaris Microsoft SQL version 8.0 MySQL
ODBC drivers	IBM ODBC driver version 7.0.00.65 Oracle ODBC driver version 8.01.07

Using this Feature

To use this feature, your master resource library (MRL) should be in the following structure:

- The MRL is in: \fap\mstrres\xxxxx (where xxxxx is an MRL name, such as RPEX1)
- The FSIUSER.INI and FSISYS.INI are in: \fap\mstrres\xxxxx
- The WIP.DFD and WIPDATA.DFD files are in: \fap\mstrres\xxxxx\deflib
- The DLL directory is: \fap\dll

You must also have these prerequisites:

- DB2, SQL, or Oracle supported database has been installed.
- A database client with a native or ODBC driver has been configured and tested.

- Documaker Server or Documaker Workstation version 11.1 patch 22 or higher has been installed.

NOTE: Version 11.1, patch 22 includes version 11.2, feature 1872.

Setting Up Your Database

Here is an overview of the steps you will perform. These steps are explained in detail on the following pages.

- (DB2 only) Issue a DB2 Bind.
- Create a WIP.DFD file in your \fap\mstrres\xxxxx\deflib\ directory and make a backup.
- Create the WIPDATA.DFD file in your \fap\mstrres\xxxxx\deflib\ directory and make a backup.
- Modify your INI files to connect Documaker with the database.
- Test using Documaker Workstation or Documaker Server.

Here are the detailed steps:

- 1 Issue a DB2 bind. This step is only necessary if you are using DB2.

Open a DB2 Command window and type the following to bind the database:

```
db2 connect to DATABASENAME user USERNAME using PASSWORD
```

Where *DATABASENAME* is the database name for the WIP index files and *USERNAME* and *PASSWORD* represent your user name and password.

```
db2 bind c:\fap\dll\db2lib.bnd collection 'DB2DB'
```

Decide upon a collection name, such as *DB2DB*. In the following steps you will update your INI file with this collection name:

```
< DBHandler:DB2 >  
CurrentPackageSet= DB2DB)
```

- 2 Create a WIP.DFD file in your \fap\mstrres\xxxxx\deflib\ directory and make a backup.

- If you already have a WIP.DFD file, create a backup copy. If you do not have a WIP.DFD file, see the example below. You can copy the example into a text file called WIP.DFD and use this as your default file.

or

- At a command prompt, using the sample RPEX1 MRL resources, create a WIP.DFD file by executing this command from the \fap\mstrres\rpx1 directory:

```
dalrw32.exe /x=.\deflib\WIP.DAL
```

Example WIP.DAL file

The WIP.DAL file is located in the \rpx1\deflib directory. Here is an example of the WIP.DAL file:

```

***Beginning of WIP.DAL file***
* DAL script to add the UNIQUE_ID to the WIP.DFD file required for
* a database
rc = LoadINIFile("FSI", ".\deflib\wip.dfd")
*msg( "Return code = " & rc )
if (rc = 0)
    msg("wip.dfd was not found!")
    return(0)
end

PutINIString("FSI", "FIELDS", "FIELDNAME", "UNIQUE_ID")
PutINIString("FSI", "FIELD:UNIQUE_ID", "INT_TYPE", "CHAR_ARRAY")
PutINIString("FSI", "FIELD:UNIQUE_ID", "EXT_TYPE", "CHAR_ARRAY")
PutINIString("FSI", "FIELD:UNIQUE_ID", "INT_LENGTH", "32")
PutINIString("FSI", "FIELD:UNIQUE_ID", "EXT_LENGTH", "32")
PutINIString("FSI", "FIELD:UNIQUE_ID", "REQUIRED", "N")
PutINIString("FSI", "FIELD:UNIQUE_ID", "KEY", "N")
PutINIString("FSI", "FIELD:UNIQUE_ID", "KEY", "N")
PutINIString("FSI", "KEY:UNIQUE_ID", "EXPRESSION", "UNIQUE_ID")
PutINIString("FSI", "KEY:UNIQUE_ID", "FIELDLIST", "UNIQUE_ID")
SaveINIFile("FSI", ".\deflib\wip.dfd")
***END of WIP.DAL file***

```

Example WIP.DFD file

After you run the WIP.DAL script or after you manually create the WIP.DFD file, your WIP.DFD file should look identical to the following:

```

***Beginning of WIP.DFD file***
< FIELDS >
; UNIQUE_ID field is not part of the standard WIP.DFD, must add.
FIELDNAME = UNIQUE_ID
FIELDNAME = KEY1
FIELDNAME = KEY2
FIELDNAME = KEYID
FIELDNAME = RECTYPE
FIELDNAME = CREATETIME
FIELDNAME = ORIGUSER
FIELDNAME = CURRUSER
FIELDNAME = MODIFYTIME
FIELDNAME = FORMSETID
FIELDNAME = TRANCODE
FIELDNAME = STATUSCODE
FIELDNAME = FROMUSER
FIELDNAME = FROMTIME
FIELDNAME = TOUSER
FIELDNAME = TOTIME
FIELDNAME = DESC
FIELDNAME = INUSE
FIELDNAME = ARCKEY
FIELDNAME = APPDATA
FIELDNAME = RECNUM

< FIELD:UNIQUE_ID >
; This section is not part of the standard WIP.DFD, must add
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 32
EXT_TYPE = CHAR_ARRAY

```

```
EXT_LENGTH = 32
KEY = N
REQUIRED = N
< FIELD:KEY1 >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 30
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 30
KEY = N
REQUIRED = N
< FIELD:KEY2 >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 30
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 30
KEY = N
REQUIRED = N
< FIELD:KEYID >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 20
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 20
KEY = N
REQUIRED = N
< FIELD:RECTYPE >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 3
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 3
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:CREATETIME >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:ORIGUSER >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:CURREUSER >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
```



```
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:MODIFYTIME >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:FORMSETID >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 32
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 32
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:TRANCODE >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 2
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 2
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:STATUSCODE >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 2
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 2
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:FROMUSER >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:FROMTIME >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
```

```
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:TOUSER >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:TOTIME >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 8
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 8
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:DESC >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 30
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 30
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:INUSE >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 1
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 1
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:ARCKEY >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 18
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 18
INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:RECNUM >
EXT_TYPE = NOT_PRESENT
EXT_LENGTH = 0
EXT_PRECISION = 0
INT_TYPE = LONG
INT_LENGTH = 4
```

```

INT_PRECISION = 0
KEY = N
REQUIRED = N
< FIELD:APPDATA >
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 150
EXT_PRECISION = 0
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 150
INT_PRECISION = 0
KEY = No
REQUIRED = No
< KEYS >
KEYNAME = UNIQUE_ID
KEYNAME = DOCTAG
KEYNAME = KEY2TAG
KEYNAME = KEYIDTAG
KEYNAME = USERTAG
< KEY:UNIQUE_ID >
EXPRESSION = UNIQUE_ID
FIELDLIST = UNIQUE_ID
< KEY:DOCTAG >
EXPRESSION = KEY1+KEY2+KEYID+RECTYPE
FIELDLIST = KEY1,KEY2,KEYID,RECTYPE
< KEY:KEY2TAG >
EXPRESSION = KEY2
FIELDLIST = KEY2
< KEY:KEYIDTAG >
EXPRESSION = KEYID
FIELDLIST = KEYID
< KEY:USERTAG >
EXPRESSION = CURRUSER

FIELDLIST = CURRUSER
***End of WIP.DFD file***

```

If you are using a custom WIP.DFD file, you must modify the WIP.DFD file for database support. Make sure no field listed is missing otherwise unpredictable results will occur.

For DB2, we recommend you change all Keys and Required options to No. While you can set these to Yes, you have to then be sure there is data for that field to avoid errors. Here is an example:

```

Key           = No
Required      = No

```

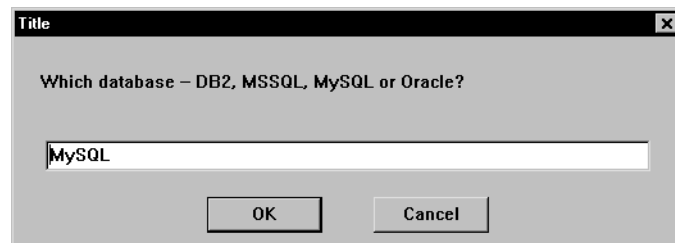
- 3 Create the WIPDATA.DFD file in your \fap\mstrres\xxxxx\deflib\ directory and make a backup.
 - If you already have a WIPDATA.DFD file, create a backup copy. If you do not have a WIPDATA.DFD file, see the following example. You can copy the example into a text file called WIPDATA.DFD and use this as your default file.

or

- At a command prompt, using the sample RPEX1 MRL resources, you can create a WIPDATA.DFD file by entering this command from the \fap\mstres\rpex1 directory:

```
dalrw32.exe /x=.\deflib\WIPDATA.DAL
```

The following prompt appears:



Enter the appropriate database type and click Ok.

Example WIPDATA.DAL file

The WIPDATA.DAL file is located in the rpex\deflib directory. If you do not have a WIPDATA.DAL file, here is an example:

```
***Beginning of WIPDATA.DAL file***
* DAL script to produce the wipdata.dfd for different database
rc = LoadINIFile("FSI", ".\deflib\wipdata1.dfd")
*msg( "Return code = " & rc )
if (rc = 0)
    msg("wipdata1.dfd was not found!")
    return(0)
end

myDB = INPUT("Which database -- DB2, MSSQL, MySQL or Oracle? ", , ,
"MySQL" )
if (myDB = "Oracle")
PutINIString("FSI", "FIELD:CARDDATA", "INT_LENGTH", "1950")
PutINIString("FSI", "FIELD:CARDDATA", "EXT_LENGTH", "1950")
end

if ((myDB = "MSSQL") OR (myDB = "MySQL"))
PutINIString("FSI", "FIELD:CARDDATA", "INT_TYPE", "BLOB")
PutINIString("FSI", "FIELD:CARDDATA", "EXT_TYPE", "BLOB")
PutINIString("FSI", "FIELD:CARDDATA", "INT_LENGTH", "8")
PutINIString("FSI", "FIELD:CARDDATA", "EXT_LENGTH", "8")
end

SaveINIFile("FSI", ".\deflib\wipdata.dfd")
***END of WIPDATA.DAL file***
```

Example WIPDATA.DFD file

After you run the WIPDATA.DAL script or after you manually create the WIPDATA.DFD file, your WIPDATA.DFD file should be identical to this example:

```
***Beginning of WIPDATA.DFD file***
< FIELDS >
FIELDNAME = FORMSETID
FIELDNAME = SEQ_NUM
FIELDNAME = CONT_FLAG
FIELDNAME = TOTAL_SIZE
FIELDNAME = CARDDATA
```

```

< FIELD:FORMSETID >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 32
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 32
KEY = N
REQUIRED = N
< FIELD:SEQ_NUM >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 5
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 5
KEY = N
REQUIRED = N
< FIELD:CONT_FLAG >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 1
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 1
KEY = N
REQUIRED = N
< FIELD:TOTAL_SIZE >
INT_Type = LONG
INT_Length = 4
EXT_Type = LONG
EXT_Length = 4
Key = No
Required = No

< FIELD:CARDDATA >
INT_Type = VARCHAR
INT_Length = 3950
EXT_Type = VARCHAR
EXT_Length = 3950
Key = No
Required = No

< Keys >
KeyName = FORMSETID
KeyName = SEQ_NUM
< KEY:FORMSETID >
Expression = FORMSETID
FieldList = FORMSETID
< KEY:SEQ_NUM >
Expression = SEQ_NUM
FieldList = SEQ_NUM
***End of WIPDATA.DFD file***

```

Changing the CARDDATA field

This is necessary if you manually need to change the WIPDATA.DFD file. This field is set for you if you use the DAL script and select the appropriate database type.

If you are using MS SQL and MySQL, change the CARDDATA field as shown here:

```

< Field:CARDDATA >
    INT_Type    = BLOB
    INT_Length  = 8

```

```
EXT_Type    = BLOB
EXT_Length  = 8
Key         = No
Required    = No
```

If you are using an Oracle native driver, change the CARDATA field as shown here:

```
< Field:CARDATA >
  INT_Type    = VARCHAR
  INT_Length  = 1950
  EXT_Type    = VARCHAR
  EXT_Length  = 1950
  Key         = No
  Required    = No
```

If you are using a DB2 native or ODBC driver, change the CARDATA field as shown here:

```
< Field:CARDATA >
  INT_Type    = VARCHAR
  INT_Length  = 3950
  EXT_Type    = VARCHAR
  EXT_Length  = 3950
  Key         = No
  Required    = No
```

Copy the WIP.DFD and WIPDATA.DFD files into the MRL's DefLib directory, such as \fap\mstres\rpex1\deflib\wip.dfd.

4 Modify your INI files to connect Documaker with the database.

You can automatically update your (FSIUSER or CONFIG) INI file by executing this command from the \fap\mstres\rpex1 directory:

```
dalrw32.exe /x=. \deflib\INI4WIP.DAL
```

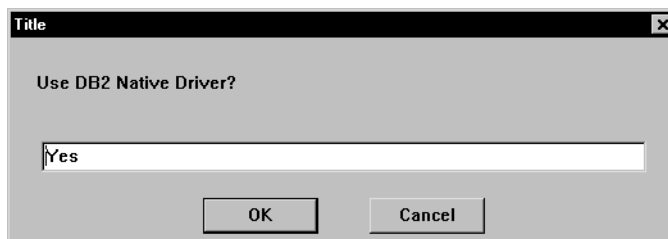
Make sure you run this DAL script in the directory where your INI file resides. The following prompt appears:



Enter the appropriate INI file name and click Ok. The following prompt appears:



Enter the table name and click Ok. For best results, use the default table name (WIPData). The following prompt appears:



Click Ok if you are using the DB2 Native driver to accept the default, otherwise enter **No** and click Ok to continue.

Depending on your database, you may be prompted for a database name, user ID, password, and other information.

Example INI4WIP.DAL file

If you do not have an INI4WIP.DAL file, here is an example of the script you should use:

```
***Beginning of INI.DAL file***
* DAL script to update the fsiuser.ini for database
infile=INPUT("Enter the INI file: ",,,"fsiuser.ini")
rc = LoadINIFile("FSI", infile)
*msg( "Return code = " & rc )
if (rc = 0)
    msg("fsiuser.ini was not found!")
    return(0)
end

* common to all
myWipData = INPUT("Please enter the table name for WipData: ", , ,
"WipData")
PutINIString("FSI", "WipData", "WipData", myWipData)
PutINIString("FSI", "WipData", "DatabaseWIP", "TRUE")
PutINIString("FSI", "WipData", "WipDataDFD", ".\deflib\wipdata.dfd")
PutINIString("FSI", "WipData", "WipDFDFile", ".\deflib\wip.dfd")

ans = INPUT("Use DB2 Native Driver? ",,,"Yes")
if (ans = "Yes")
    PutINIString("FSI", "DBHandler:DB2", ";BindFile", "")
    PutINIString("FSI", "DBHandler:DB2", ";CurrentPackageSet",
"collection_name");
    PutINIString("FSI", "DBHandler:DB2", "CommitEvery", "0")
    PutINIString("FSI", "DBHandler:DB2", "Connect", "Yes")
    PutINIString("FSI", "DBHandler:DB2", "CreateIndex", "No")
    PutINIString("FSI", "DBHandler:DB2", "CreateTable", "No")
    mydb = INPUT("Name of the database? ",,,)
    * check for not empty

    PutINIString("FSI", "DBHandler:DB2", "Database", mydb)
    myuserid = INPUT("userid? ",,,)
    PutINIString("FSI", "DBHandler:DB2", "UserID", myuserid )
    mypasswd = INPUT("password? ",,,)
    PutINIString("FSI", "DBHandler:DB2", "Passwd", mypasswd )
    PutINIString("FSI", "DBTable:WipData", "DBHandler", "DB2")
    PutINIString("FSI", "DBTable:WipData", "UniqueTag", "FORMSETID")
```

```

        SaveINIFile("FSI", "db2user.ini")
        return('db2')
    end

ans = INPUT("Use Oracle Native Driver? ",,,"Yes")
if (ans = "Yes")
    PutINIString("FSI", "DBHandler:ORA", "CreateIndex", "No")
    PutINIString("FSI", "DBHandler:ORA", "CreateTable", "No")
    mydb = INPUT("Name of the database? ",,,)
    *   check for not empty

    PutINIString("FSI", "DBHandler:ORA", "Database", mydb)
    myuserid = INPUT("userid? ",,,)
    PutINIString("FSI", "DBHandler:ORA", "UserID", myuserid )
    mypasswd = INPUT("password? ",,,)
    PutINIString("FSI", "DBHandler:ORA", "Passwd", mypasswd )
    PutINIString("FSI", "DBTable:WipData", "DBHandler", "ORA")
    PutINIString("FSI", "DBTable:WipData", "UniqueTag", "FORMSETID")
    PutINIString("FSI", "ORA_FileConvert", "Wipdata", "WIPDATA")
    SaveINIFile("FSI", "orauser.ini")
    return('oracle')
end

PutINIString("FSI", "DBHandler:ODBC", "AlwaysSQLPrepare", "Yes")
myServer = INPUT("Please enter the Data Source Name : ", , , )
*verify it is not blank
PutINIString("FSI", "DBHandler:ODBC", "Server", myServer)
PutINIString("FSI", "DBHandler:ODBC", "CreateTable", "No")
PutINIString("FSI", "DBHandler:ODBC", "CreateIndex", "No")
myUserID = INPUT("Please enter the UserID: ", , , )
PutINIString("FSI", "DBHandler:ODBC", "UserID", myUserID)
myPasswd = INPUT("Please enter the password: ", , , )
PutINIString("FSI", "DBHandler:ODBC", "Passwd", myPasswd)

PutINIString("FSI", "DBTable:Wip", "DBHandler", "ODBC")
PutINIString("FSI", "DBTable:Wip", "UniqueTag", "FORMSETID")

PutINIString("FSI", "DBTable:WipData", "DBHandler", "ODBC")
PutINIString("FSI", "DBTable:WipData", "UniqueTag", "FORMSETID")

PutINIString("FSI", "ODBC_FieldConvert", "DESC", "DESCRIPTION")

SaveINIFile("FSI", "odbcuser.ini")
return('odbc')
***END of INI.DAL file***

```

Manually updating INI files

To update your INI files manually, here are the settings you need to add. Add this control group regardless of the database type:

```

< WIPData >
    WIPData      = WIPData
    DatabaseWIP= Yes
    WIPDataDFD = .\deflib\wipdata.dfd
    File        = WIP
    Path        = .\wip
    WIPDFDFile = .\deflib\wip.dfd

```


Add these options if you are using the DB2 (native driver):

```
< DBTable:WIP >
    DBHandler    = DB2
    UniqueTag    = DocTag
< DBTable:WIPData >
    DBHandler    = DB2
    UniqueTag    = FormSetID
< DBHandler:DB2 >
;   BindFile     = d:\fap\dl1110\db2lib.bnd
    CommitEvery  = 0
    Connect      = Yes
    CreateIndex  = No
    CreateTable  = No
    Database     = ARCDDB
    Debug        = No
    PassWd       =
    UserID       =
    CurrentPackageSet= REL112
    Class        = DB2
< DB2_FileConvert >
    WIP          = WIP
```

Add these options if you are using the SQL (ODBC) driver:

```
< DBTable:WIP >
    DBHandler    = ODBC
    UniqueTag    = DocTag
< DBTable:WIPData >
    DBHandler    = ODBC
    UniqueTag    = FormSetID
< DBHandler:ODBC >
    CommitEvery  = 0
    Connect      = Yes
    CreateIndex  = No
    CreateTable  = Yes
    Server       = WIPDB
    Debug        = No
    PassWd       =
    UserID       =
    AlwaysSQLPrepare = Yes
<ODBC_FileConvert>
    WIP          = WIP
< ODBC_FieldConvert >
    Desc         = DESCRIPTION
```

Add these options if you are using the Oracle (ODBC) driver:

```
< DBTable:WIP >
    DBHandler    = ODBC
    UniqueTag    = DocTag
< DBTable:WIPData >
    DBHandler    = ODBC
    UniqueTag    = FormSetID
< DBHandler:ODBC >
    CommitEvery  = 0
    Connect      = Yes
```

```

        CreateIndex      = No
        CreateTable      = No
        Server            = WIPDB
        Debug             = No
        PassWd            =
        UserID            =
        AlwaysSQLPrepare = No
<ODBC_FileConvert>
    WIP                  = WIP
<ODBC_FieldConvert >
    Desc                 = DESCRIPTION

```

Add these options if you are using the Oracle (native) driver:

```

< DBTable:WIP >
    DBHandler           = ORA
    UniqueTag            = DocTag
< DBTable:WIPData >
    DBHandler           = ORA
    UniqueTag            = FormSetID
< DBHandler:ORA >
    Connect              = Yes
    CreateIndex          = No
    CreateTable          = No
    Database             = WIPDB
    Debug                = No
    PassWd               =
    UserID               =
    Class                = ORA
< ORA_FileConvert>
    WIP                  = WIP
< ORA_FieldConvert >
    Desc                 = DESCRIPTION

```

This table describes how you use the DBHandler and UniqueID fields in the DBTable:WIPData control group:

Option	Description
DBHandler	Tells the system that WIPDATA is a table in an DB2-compliant database used to store the WIP data.
UniqueTag	Use this option to assign a unique key. For example, you could enter FormSetID or DocTag, as long as it is unique.

If the name you assign to the WIP table is longer than eight characters, add the following control group and option for the name conversion to take place:

```

< DB2_FileConvert >
    WIP = FAPQA.WIP

```

If a DBA needs to create the WIP index table with all the fields defined in the WIP.DFD file and the WIPDATA table with all the fields defined in the WIPDATA.DFD, you can set the CreateTable option to No. Otherwise, set it to Yes to tell the system to create the table.

Oracle and DB2 database require that you set CreateTable to No. A DDL is required to be created and run on an Oracle and DB2 database by a DBA. The following steps have to be executed for those databases.

If you set the CreateTable option to No, you must create a DDL from the DFDs. From a command prompt, go to your xxxx\deflib directory, such as rpex1\deflib. Then execute one of these commands:

To **Enter this command**

Produce the DDL	dfd2ddl.exe /i=wip.dfd /o=wipdb.sql
Append to the DDL	dfd2ddl /i=wipdata.dfd /o=wipdb.sql

The DBA can then execute the DDL to create WIP and WIPData tables.

Specify the same output DDL file to append the SQL statements into one file, such as WIPDB.SQL. If the database is not MS SQL, be sure to include the /D parameter to specify the database to produce the correct DDL.

NOTE: The Desc field in the WIP.DFD file is reserved for MS SQL and Oracle. Therefore, the DBA must change it to a different name, such as Description, in the DDL. Use these options to map the Desc field in the WIP.DFD file to the Description field in the WIP table:

```
< ODBC_FieldConvert >
    Desc = DESCRIPTION
< ORA_FieldConvert >
    Desc = DESCRIPTION
```

5 Test the result.

To test **Follow these steps**

Documaker Workstation	Run Documaker Workstation (AFEMAIN.EXE) to create and save a test transaction to WIP. Then complete it to delete the test transaction from the WIP List.
Documaker Server	<p>To store WIP data in a database, run this command for multi-step execution:</p> <pre>gentnw32.exe, gendaw32.exe and genwpw32.exe</pre> <p>For single-step mode, use this command:</p> <pre>gendaw32.exe /jdt=wipjob.jdt</pre> <p>Then, to merge WIP data onto your form sets, run this command:</p> <pre>gendaw32.exe /jdt=afgjob.wip</pre>

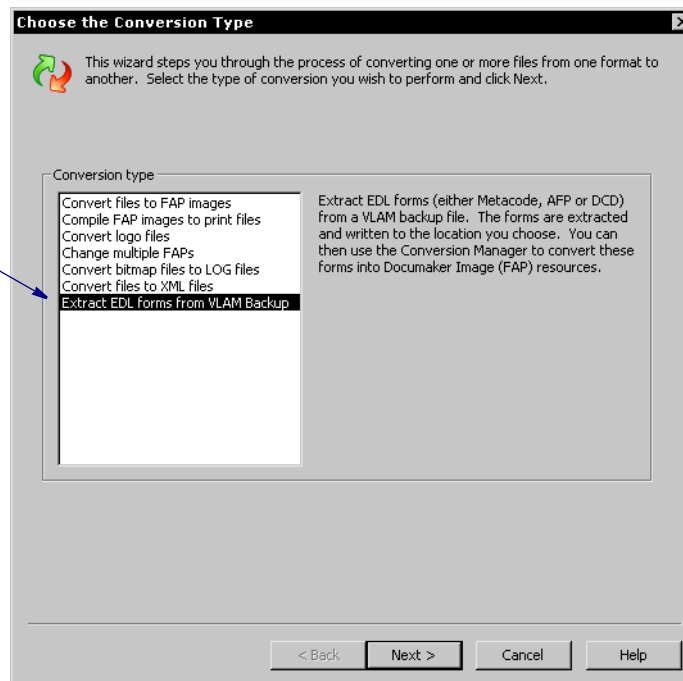
1888
RPS

EXTRACTING EDL FORMS FROM A VLAM FILE

You can now use the Conversion manager to extract the Electronic Document Library (EDL) forms from a VLAM backup file and write those EDL members to disk. Each EDL form is written to disk and given its actual Documerge form name instead of the 8-byte *chain* name that Docucorp (FP) Control Panel would give it.

To extract EDL forms, open Conversion manager. The Choose the Conversion Type window appears:

Click here to extract EDL members from a VLAM file



Choose the option to extract EDL forms and click Next. The Select VLAM Backup File and Forms to Extract window appears:

Enter the name of the VLAM backup file or click here to select browse for the file.

Studio then display a list of the forms in the VLAM file here

Use these buttons to indicate which forms you want Studio to extract. For instance, click here to move all forms into the extract list.

First choose the VLAM backup file you want to extract forms from. Studio detects the forms in the VLAM file and displays them in the left column. Highlight the forms you want to extract and use the arrow buttons to move them to the right column.

Click Next when you are finished. The Extraction Information window appears:

The available print types are AFP, DCD, and META. The first available print type appears as the default.

Click here if you also want to create FAP files.

Name	Description	Effective Date
LM.LIFE.BANNER	INSTALLATION TEST BANNER	1/1/1900
LM.LIFE.BANNER.1403	INSTALL TEST 1403	1/1/1900
LM.LIFE.BANNER.3800I	INSTALL TEST 3800I	1/1/1900
LM.LIFE.CONTRACT	INSTALLATION TEST CONTRACT	1/1/1900
LM.LIFE.CONTRACT.1403	INSTALL TEST 1403	1/1/1900
LM.LIFE.CONTRACT.3800I	INSTALL TEST 3800I	1/1/1900
LM.LIFE.CONTRACT.IMI	INSTALL TEST 9700	1/1/1900
LM.LIFE.DEC	INSTALLATION TEST DEC	1/1/1900
LM.LIFE.DEC.1403	INSTALL TEST 1403	1/1/1900

You can choose from these print types: AFP, DCD, and META. Once you select the print type, Studio updates the information in the grid.

If you only want to extract the EDL members to disk, click Next and go to the Finish window. The Finish window shows you the path where the EDL members will be written. You can change this path if you want. If the path you enter does not exist, Studio creates it for you.

If you want to extract the EDL members to disk and also convert these EDL members (normalized Metacode or normalized AFP files) into FAP files, check the Proceed to Print Stream to Image (FAP) Conversion box and click Next. You can then follow the pages of the wizard to convert the EDL members into FAP files and check the FAP files into the library.

NOTE: For more information about converting the print streams into images, see feature 2165.

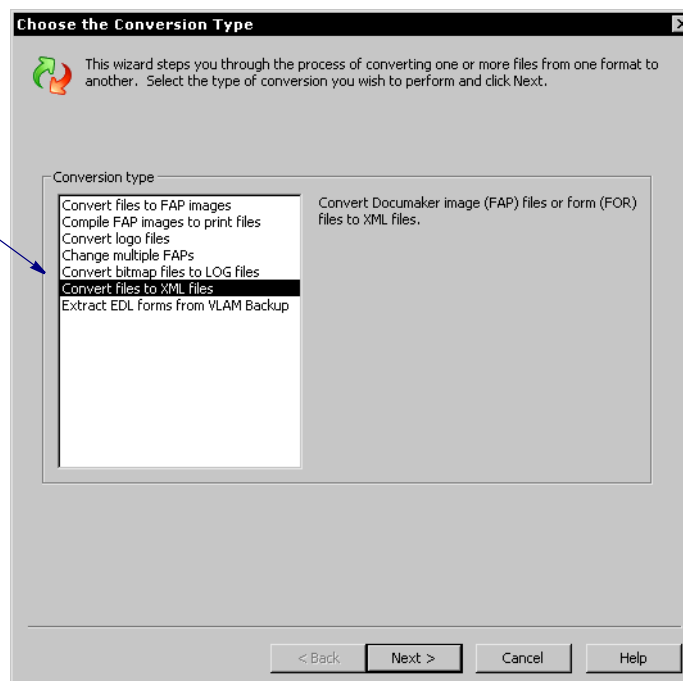
1891
RPS

CONVERTING FAP AND FOR FILES INTO XML FILES

You can now convert FAP files and FOR files into XML files which include field and field attribute information. You can use the output and the XMLDIFF utility to compare the fields, for instance, on a form in your production library versus a form in your development library.

To convert a FAP or FOR file, open Conversion manager. The Choose Conversion Type window appears.

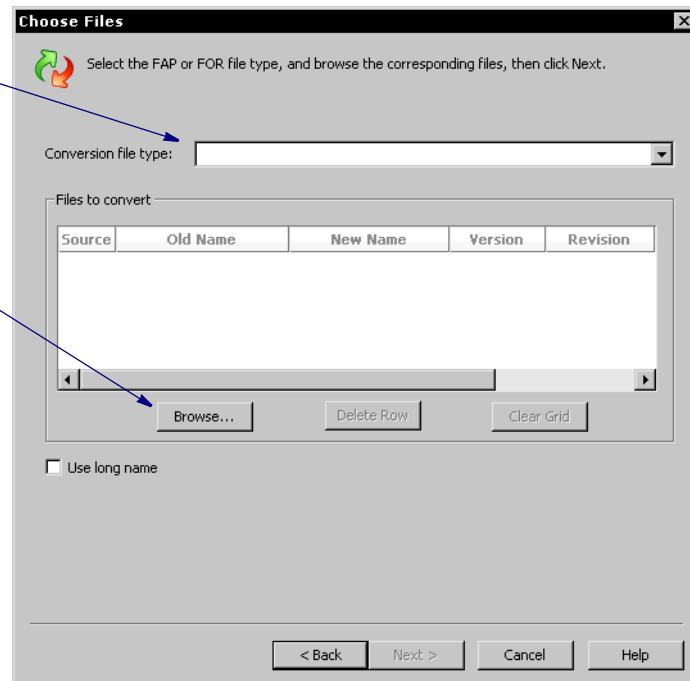
Choose the option to convert to XML files.



Select the option to convert to XML files. The Choose Files window appears.

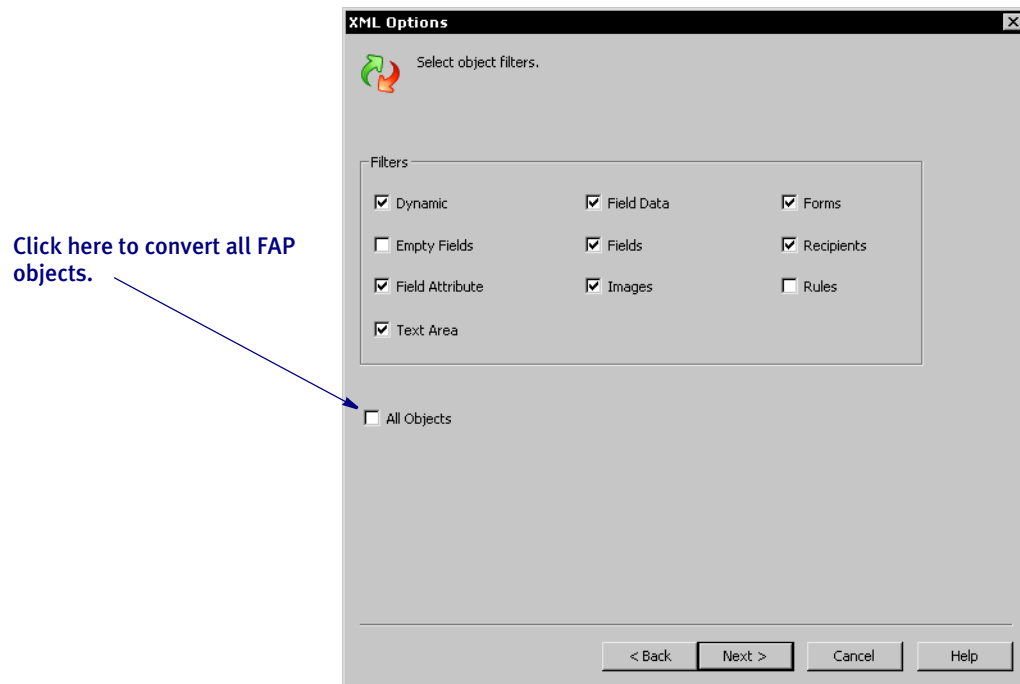
Choose the type of file here.

Click Browse to select the files you want to convert.

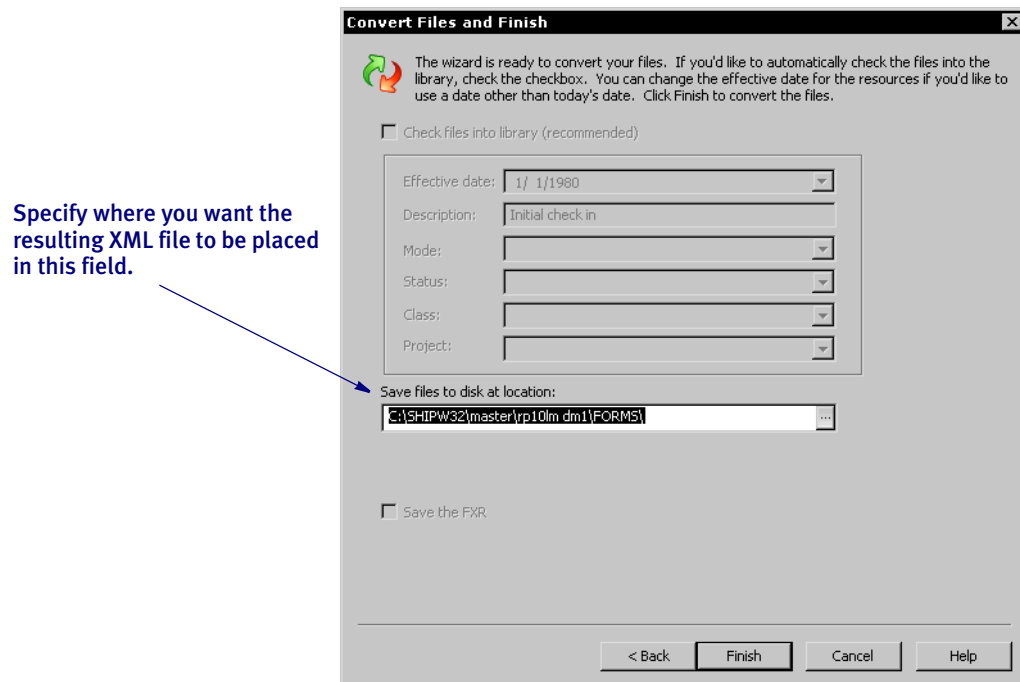


First select the type of file, either FAP or FOR. Then click Browse to choose the files you want to convert and click Next. You can hold down the **SHIFT** key and select multiple FOR or FAP files. The XML Options window appears.

NOTE: You cannot convert specific versions/revisions of the form. Studio only converts the latest version/revision.



Select the types of items you want included in the XML file. Click Next when you are finished. The Convert Files and Finish window appears.



Click Finish to start the conversion.

1892
RPS

USING THE NEW IMAGE USAGE REPORT

You can use the new Image Usage report in Documaker Studio to list all of the images associated with the Business Definition (BDF) file you specify.

Studio starts with the Business Definition file to learn about the groups and forms in use. From this, the final report lists all images in use and those images that are not directly referenced on any form.

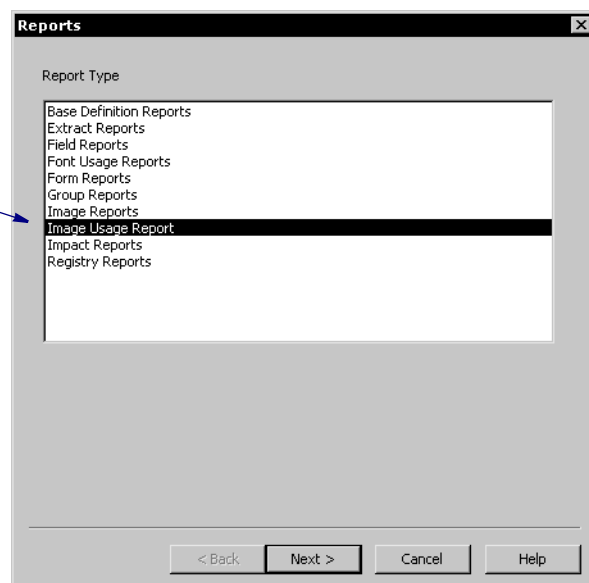
The BDF file functions more like a tree trunk, with groups as the main branches, forms as the smaller branches, and images as the leaves on a tree. So images are not explicitly defined in the BDF file, but it contains the information necessary for Studio to determine which groups, forms, and images belong to a business definition.

This report is divided into these sections:

- Referenced images - These are library images (FAPs) referenced on forms (FOR) derived from the groups (GRP) defined in the business definition (BDF).
- Images not referenced by forms - These are library images that were not found by following the descendants of the business definition (BDF) you selected.
- Referenced images but not in library - These are image names that were referenced on a form (FOR), but were not in the library.

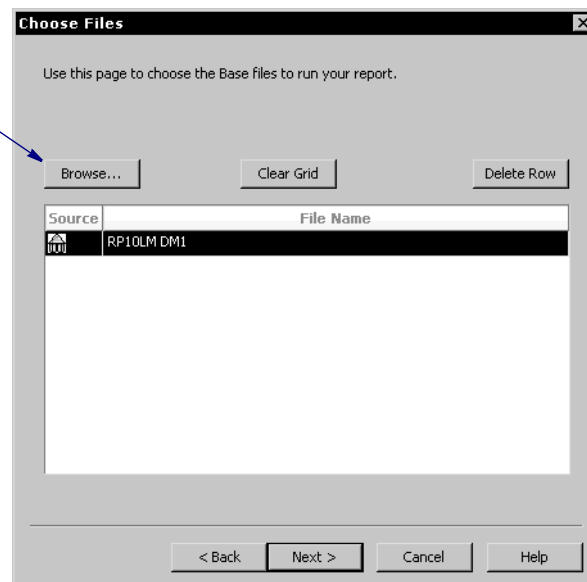
To create this report, open the Report manager and select the option to generate the Image Usage report.

Highlight the Image Usage report and click Next.



The Choose Files window appears:

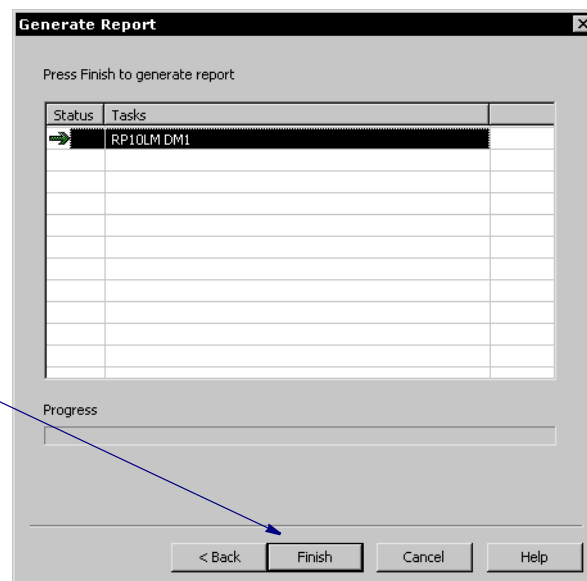
You can click Browse to search for the BDF file you want.



Here you select the Business Definition (BDF) file from which to derive the report. Use the Browse button to choose the BDF file from the library. You may only have one BDF file. When ready, click Next.

The Generate Report window appears:

Click Finish.



Click Finish to generate the report.

Here is an example of the first section of the report:

Use these buttons to send the report to the printer or to close the report.
You can click Wizard to close this report and go back to the Report manager and generate another report.

Print	Print All	<	<<	>>	>	Wizard	Close
-------	-----------	---	----	----	---	--------	-------

RP10LM DM1		
Referenced Images		
Image	Number of Pages	Number of References
102smfnt	1	1
ARRWDMY	1	1
BASE1	1	1
BASE2	1	1
BASE3	1	1
BASE4	1	1
BASE5	1	1
BASE6	1	1
BASE7	1	1
BASE8	1	1
CUST1	1	1
CUST2	1	1
CUST3	1	1
CUST4	1	1
CUST5	1	1
CUST6	1	1

This section lists the images in alphabetical order, notes the number of pages in each image, and the number of references to that image. For the number of pages, any image that is not a multi-page image will be shown as one (1) page — even if the image is less than a page in size.

Scroll down to the next section:

Print

Print All

<

<<

>>

>

Wizard

Close

qtrans2	1	1
qtrtal	1	1

Images Not Referenced By Forms

Image
102SMFNT
APPENDS
BCKGRND
BILL OF SALE
BILLCGAR
BILLCSWR
BILLCWT3
BILLCWTR
BILLHDRP
BILLLOAN
BILLMSG1
BILLMSGR
BILLMSSL
BILLQUES
BILLRGAR
BILLRSWR
BILLRWT3
RP10LMDM1

This section lists the images which are not referenced by any forms. These could be orphaned images or simply images you have not yet combined into a form.

It is possible that these images are referenced by a Paragraph Assembly routine or by DAL scripts. In addition, you may have images in your library that are not used on current documents, but are referenced by archived documents.

NOTE: Be cautious about removing images based on this report. This report focuses on the business definition you selected. It does not tell you if the image is used in another business definition.

Scroll down to the next section:

Images Referenced But Not in Library	
Image	Form
qimage3	qimage3

The last section of the report lists the images which are referenced by a form, but are not in the library. On the left you see the name of the image. On the right is the form that references that image.

1895
RPS

GENERATING THE USPS INTELLIGENT MAIL BARCODE (4-STATE CUSTOMER BAR CODE)

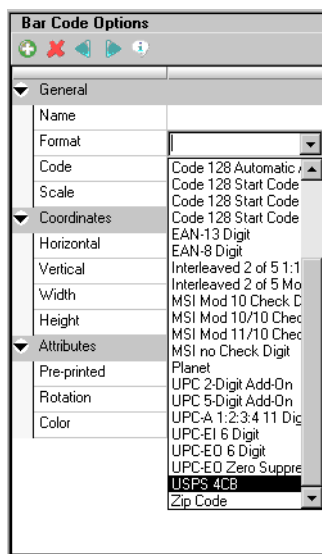
You can now generate the Intelligent Mail[®] bar code, formerly referred to as the *4-State Customer bar code* on your forms. The Intelligent Mail[®] bar code is a height-modulated bar code using four distinct vertical bar types. It encodes a 20, 25, 29, or 31-digit string into 65 vertical bars, each representing one of four possible states: full bar, ascender, tracker, and descender. Intelligent Mail bar code expands the ability to track individual mail pieces and provide its customers with greater visibility in the mail stream.

Type	Field	Digits
Tracking Code	Bar Code ID	2 (The 2nd digit must 0-4)
	Service Type ID	3
	Customer ID	6
	Sequence Number	9
Routing Code	Delivery Point ZIP Code	0, 5, 9, or 11
Total		31 maximum

Here is an example of the 4CB bar code:



In Studio or Image Editor, you would select the USPS 4CB option in the Format field. Here is an example from Studio:



1901
RPS

USING THE NEW UNDO/REDO OPTION

When working in Form and Group managers in Documaker Studio, you can now cancel or repeat the last action you performed using the Edit, Undo option or the Edit, Redo option. You can also press CTRL+Z to undo an action or CTRL+Y to repeat an action.

Studio tracks up to 25 actions and you can continue to choose Undo/Redo, or press CTRL+Z/CTRL+Y, until you get to the change you want to undo or repeat. If a change affects multiple items once, Studio considers it a single change.

NOTE: The Undo/Redo option is also available in the Image and Logo managers.

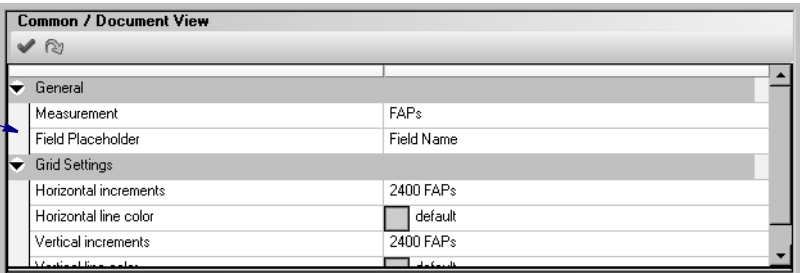
1906
RPS

VIEWING FIELD INFORMATION

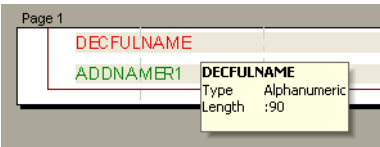
By default, Studio now shows you the name of the field instead of a series of Xs when you are designing forms.

To turn off the display of the field’s name and instead see Xs in the field, choose Options from the View menu. Then choose the Common, Document View option. Use the Field Placeholder field to turn on or off the display of the field’s name.

Customize the display here

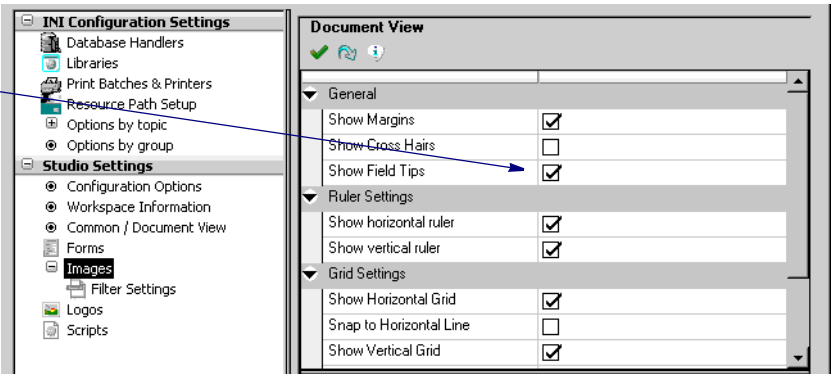


Regardless of how you set the Field Placeholder field, if you pause the cursor over a field, Studio can show you the name of the field, its type, and its length, as shown here:



These information boxes are called field tips and you can turn them on or off for the Form and Image managers using the Settings option. This example shows the setting for the Image manager:

Click here to turn on or off the display of field tips.



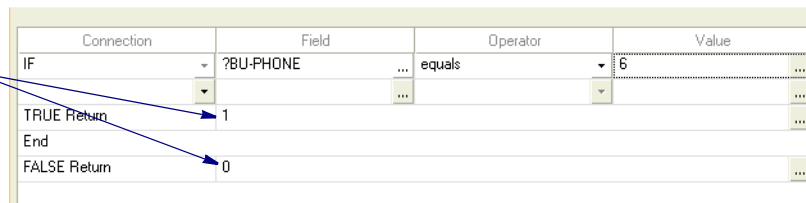
1911
RPS

SPECIFYING RETURN VALUES

The DAL wizard in Studio now lets you specify return values based on conditions you establish. The default return value for a TRUE result is one (1) and zero (0) for a FALSE result, but you can specify different values.

Enter the value you want returned if the condition is evaluated to be true or false. You can also select an XDD member you want the system to look up.

Enter the return values you want to use here.



The screenshot shows a table with four columns: Connection, Field, Operator, and Value. The first row is an IF condition: Connection is a dropdown, Field is '?BU-PHONE', Operator is 'equals', and Value is '6'. Below the table, there are three rows for return values: 'TRUE Return' with a value of '1', 'End', and 'FALSE Return' with a value of '0'. A blue arrow points from the text 'Enter the return values you want to use here.' to the 'TRUE Return' and 'FALSE Return' rows.

Connection	Field	Operator	Value
IF	?BU-PHONE	equals	6
TRUE Return			1
End			
FALSE Return			0

Keep in mind...

- The return value you specify can only be a constant or an XDD member. It cannot be a function or a variable.
- If you edit a script using the script editor, the wizard may not parse the script correctly.
- If you edit the *TRIGTOK line, the wizard may not parse the script correctly.

1916
RPS

USING THE NEW RESETIMAGEDIMENSION RULE OPTIONS

Two new options are available for the ResetImageDimension rule. These options enable the system to work as it did in prior releases.

For instance, in legacy implementations, where only DDT files were loaded and not FAP files, fields are the only objects on an image the ResetImageDimension rule recognizes. That meant image bottoms were most likely being assigned at the last mapped field. In subsequent releases, because of new features and the new Studio model of development, FAP files are loaded during batch runs. Therefore, to get behavior similar to what you had in the past, you must either change the ResetImageDimension rules to use the new LastField parameter, or use the new RID_LastMapField INI option to change the behavior of the MinHeight parameter.

- The new LastField parameter is similar to the MinHeight parameter except only the fields with data are examined when the system looks for the lowest point on the image. Use this parameter on images that have only fields or no objects lower than the last few fields you expect to map.

When you use this parameter, the bottom of the image is moved to a position below the lowest mapped field on the image. This is what you want in situations where the next image should be placed immediately below where the last field was mapped. For instance, assume you have a small image used for addresses. It may contain up to eight lines, but depending upon the address only two or three lines might be used and you would like to set the bottom of the image below where the last field was mapped. Here is an example:

```
<Image Rules>  
;ResetImageDimension;LastField;
```

NOTE: Depending upon your print or display method, changing the bottom of the image with this parameter could mean that any objects below this point will not be visible and may not print. Or it could mean those objects will simply overprint the next image in sequence on the same page.

- You can use the new RID_LastMapField INI option to change the MinHeight parameter to work like the new LastField parameter described above.

```
< RunMode >  
RID_LastMapField = Yes
```

(RID is an abbreviation for ResetImageDimension.)

The default is No. Enter Yes if you want to modify the behavior of the MinHeight parameter on all images that would use the ResetImageDimension rule.

NOTE: Depending upon your print or display method, changing the bottom of the image with this option could mean that any objects below this point will not be visible and may not print. Or it could mean those objects will simply overprint the next image in sequence on the same page.

Also note that the term *last field* refers to the *lowest* field mapped on an image and not the physical sequence in which the fields are mapped. The lowest field is the one that is the greatest distance from the top of the image.

1918
RPS

RETAINING THE DUPLEX VALUE WHEN CONVERTING NORMALIZED METACODE FILES

The system now retains a normalized Metacode file's DJDE Duplex value as it is converted from a normalized Metacode file to a FAP. This duplex value (either Yes or No) is stored in the FAP file and, when the FAP file is normalized back to a Metacode file, it is written back out as it was read in.

Before this change, the normalized Metacode file was always written to contain a No value for Duplex.

1932
RPS

CONTROLLING ACCESS TO THE FIELD DICTIONARY MANAGER

You can now control access to the Field Dictionary Manager in Documaker Studio. The system administrator can, for each user, assign the following access rights:

- Full access
- View only access

- No access

You must have Full Access to change and update information via the Field Dictionary Manager. These settings also determine whether the user can update the field database using the Image Manager or imports that store fields into the FDB.

This does not prevent a user from creating a variable field on an image, nor does it prevent a user from dragging and dropping a field from the FDB (common field list). This only affects adding or updating field information in the field database.

1937
RPS

USING THE NEW TOOLBAR BUTTONS

Within the context of the new user interface enhancements, many of the toolbar buttons have been changed to use icons that are more intuitive and similar to those found in other commonly-used software products. These changes help make Studio easier to use.

In addition, the settings offers a larger view of the toolbar for those who like larger buttons. You can also include text on the buttons. Here are some examples:

Large icons



Large icons with text



Small icons



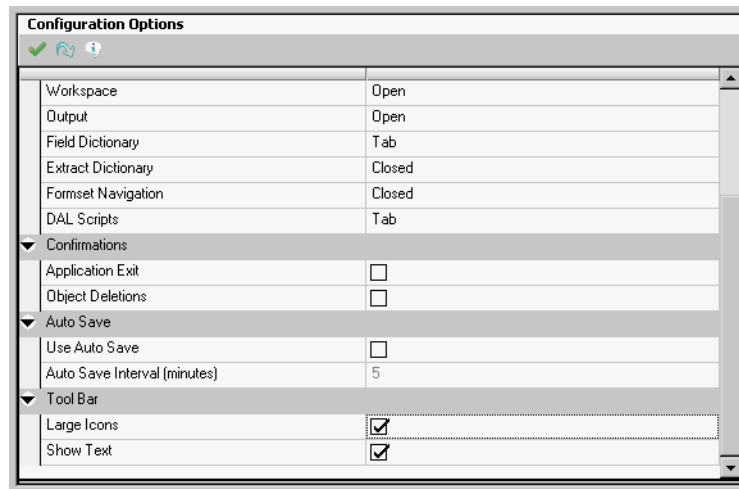
Small icons with text



NOTE: If you place your mouse on a button, tool tips appear to identify the button.

To choose between large and small icons and determine whether the name of the tool appears on the toolbar buttons, choose the Manage, Settings option and then click on Configuration Options:

Use these options to choose large or small icons and to display or omit the name of the button.



NOTE: You can also get to the Configuration Options from the Image manager by choosing the View, Options menu option.

1938
RPS

DOCKING AND AUTOMATICALLY HIDING WINDOWS

Many of the supporting windows associated with the Workspace view or an individual document's view can now be docked, tabbed, or automatically hidden when not in use. You choose the arrangement that works best for you.

You can drag and drop to combine windows which will create a new tab in a single position. Or you can *tear off* any of the individual tabs to create new windows that can dock beside another docked window or on a different side of your workspace view.

You can also combine certain windows in a *split mode*, where they occupy a similar space. This lets you see portions of each window at a same time. When you use the split view, you can size each section to suit your needs. And even when sized smaller, you are still able to scroll both vertically and horizontally to access to any part of the window you need. In some cases, you can even close certain docking windows that you might not use or only use sparingly.

The docking windows each have a small icon that looks like a push-pin near the top corner. Here are some examples:

This indicates the window is pinned to the screen.

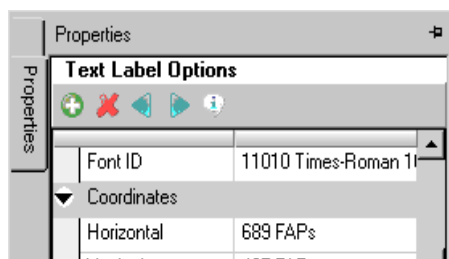


This indicates that auto-hide is on for this window.



The push-pin operates much like it would if you were pinning a piece of paper to a bulletin board. When the push-pin is upright the window is *pinned* and docked in the position you have chosen. If you click on the push-pin, it will turn sideways and indicate the window is no longer pinned in position. This activates the auto-hide feature and the window will automatically hide and appear as a tab on the edge of the document view when your mouse is in another part of the document.

Move your cursor over this tab and the Properties window is reactivated.

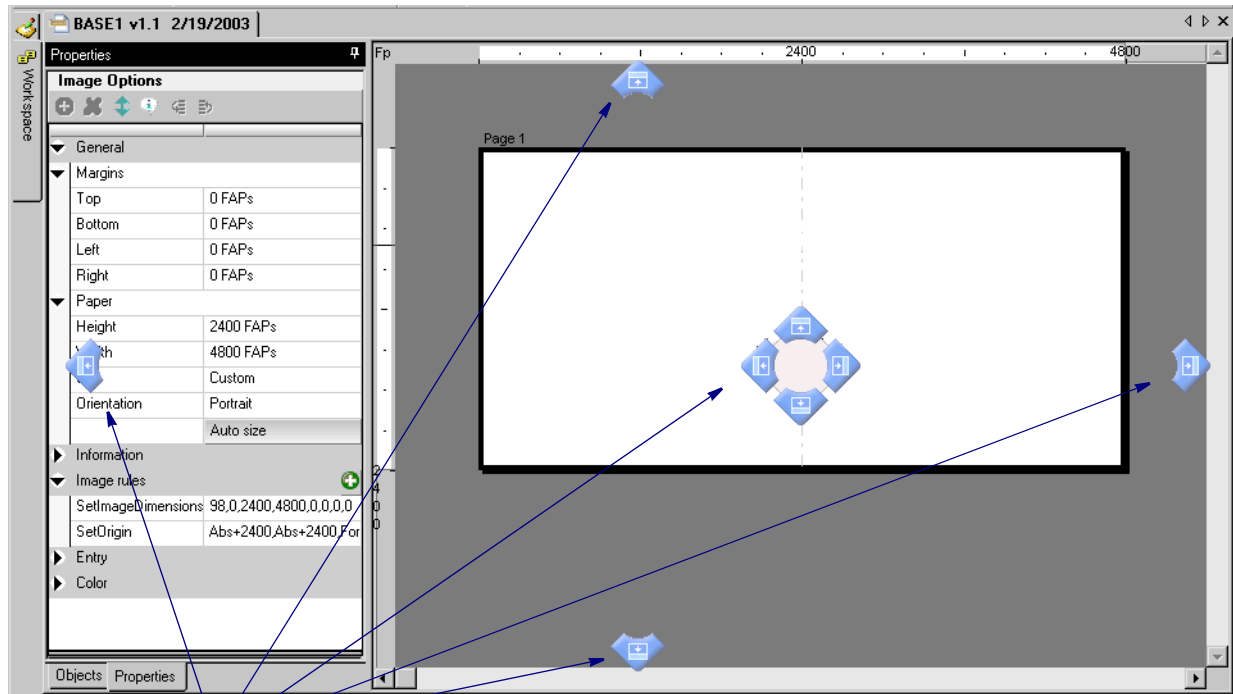


To reactivate the view window, move your cursor over the tab or click on it. If you click on an auto-hide tab the window remains open until you click another part of the form. Otherwise, the window automatically hides again after you move the cursor away.

Depending upon whether you are dropping the window on top of another, moving the window slightly one way or the other will show you the different options you have for viewing the window in that position. When you release the mouse, the window appears in the position shown to you.

If you close a window for your document view, you can reactivate it via the Window menu. Simply choose the appropriate window name and it reappears.

You can also use the docking helpers as you drag a dockable window to a new position in the window:



The docking helpers appear when you click on a dockable window and begin to drag it to a new position in the window.

If you move the cursor to one of the docking helpers and click, the window takes the position indicated. Sometimes this may be helpful in positioning windows.

1950
RPS

SETTING THE DEFAULT EFFECTIVE DATE FOR NEW RESOURCES

Now you can set a default effective date for new resources being checked into a workspace. Previously, workspace creation and import would default the date to 1-1-1980, while the check-in window for a new resource would default to the current date.

Studio now remembers the last date used for a new resource or workspace creation and uses that date as the default for the next new resource. You can, of course, override the date.

NOTE: If no default value is found for the workspace, Studio uses Jan 1, 1980.

The default date is stored as a registry item by workspace name so if you have more than one workspace, each can have a different default. The date string is stored in YYYYMMDD format.

When a workspace is created, the default library date is stored in the registry. When new resources are checked into the workspace, the library effective date field defaults to the last value used for a new resource.

Studio does not update the registry for existing library items that you give a new version effective date. To modify the default date for a workspace, change the effective date on a new library resource.

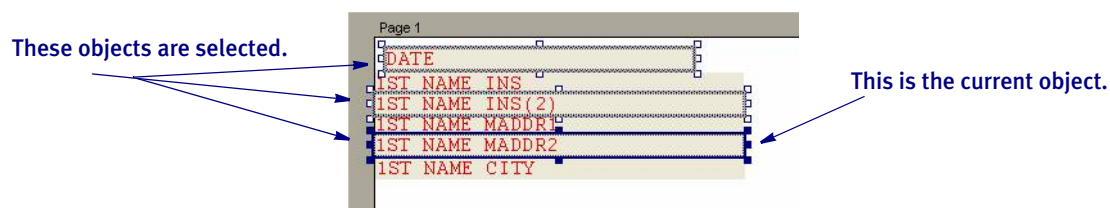
Also, other locations that query for a default effective date will use this same value. This includes the various import and conversion features where resources are checked into the library.

1966
RPS

HIGHLIGHTING SELECTED OBJECTS

The way Studio displays selected objects, especially images, has been enhanced to make it easier to tell what you have selected and which one is the current object.

Here is an example which shows three variable fields selected, one of which is the current object:



1971
RPS

USING THE FORM MANAGER NAVIGATION TREE

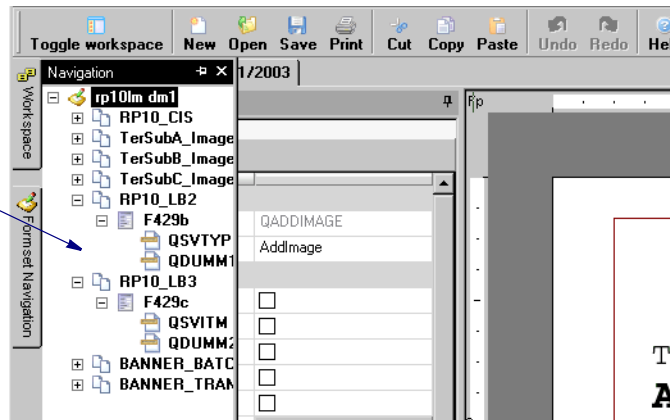
To make it easier to open and also see the relationships between the various content within the library, Studio now lets you display a navigation tree by selecting the View, Navigation option.

The navigation tree lets you see the file relationships from the Business Definition (BDF) file down to the images (FAP files) that comprise your forms. The information is presented in a hierarchical tree fashion.

NOTE: You may prefer to select documents to edit using this tree view instead of using the simple alphabetical listing provided in a library selection view.

Here is an example:

You can click any of the images shown here to quickly open that image.



By default, the Navigation tree is in auto-hide mode. You can pin it to the window if you prefer to always have it open.

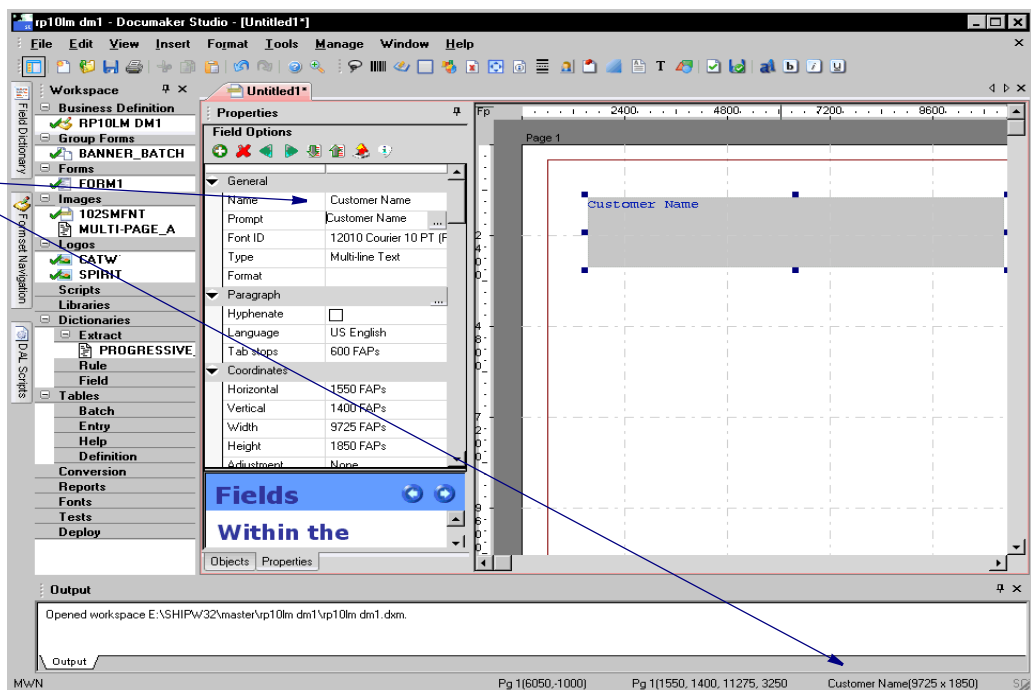
NOTE: In addition to editing an object from the Navigation window, you can preview or open each object for view-only access.

1972
RPS

SHOWING THE OBJECT NAME ON THE STATUS BAR

When you are working with an object on an image, such as a variable field, Studio now displays the object's name on the status bar, along with the object's coordinates.

The field name appears under Field Options and on the status line



In this example, the variable field's name also appears under Field Options, but as you scroll through the Field Options, the field's name can scroll out of view. With this enhancement, you can always see the field name.

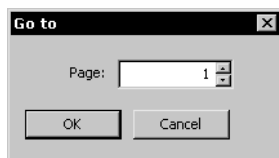
NOTE: If the field name is too long to fit on the status bar, it will be truncated.

1981
RPS

EASIER NAVIGATION IN MULTI-PAGE FAP FILES

Now you have two new ways to navigate in multi-page FAP files.

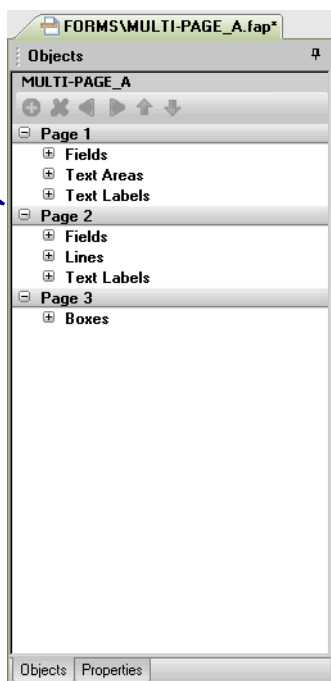
- You can use the Edit, Go to Page option or right click while in the image and select Go to Page. The Go To window appears:



Enter or select the page you want to jump to and click Ok.

- Click on the page number in the object tree to jump to the page.

Click here to go to page 2.



1990
RPS

ADDING CODE PAGE AND CHARACTER SET NAMES TO THE FXR

The system now adds character set and code page names to the font cross-reference (FXR) file when you import AFP fonts. This means the FXR entries for the imported AFP fonts will now include the AFP code page and character set names, when before only the AFP coded font would be included.

NOTE: This is available in both Docucreate and Documaker Studio and only applies when you are importing AFP fonts.

Keep in mind the additional information will appear in any new FXR file you create. To add this information to an existing FXR file, you will need to re-create the FXR file. See also Feature 2064 for additional information.

1995
RPS

USING THE CHAR AND CHARV DAL FUNCTIONS

Version 11.2 includes these new DAL functions which you can use to convert integers into characters and vice versa.

Char

Use this function to convert an integer into a single character.

Syntax

Char (Integer)

Parameter	Description	Defaults to...
Integer	An integer value that ranges zero (0) to 255.	No default.

Example

Here is an example:

```
what_char = Char (64)
```

The variable, *what_char*, is set to the character: '@'.

CharV

Use this function to convert a single character into an integer value.

Syntax

CharV (String)

Parameter	Description	Defaults to...
String	A character string. If the string contains more than one character, only the first character is converted. The remaining characters are ignored.	No default.

Example In this example, assume the variable, `char_to_convert`, contains the single character: “@”.

```
#_the_integer = CharV(char_to_convert)
```

The integer variable, `#_the_integer`, is set the value: 64.

In this example, assume the variable, `the_string`, contains the characters: “@()”.

```
#_the_integer = CharV(the_string)
```

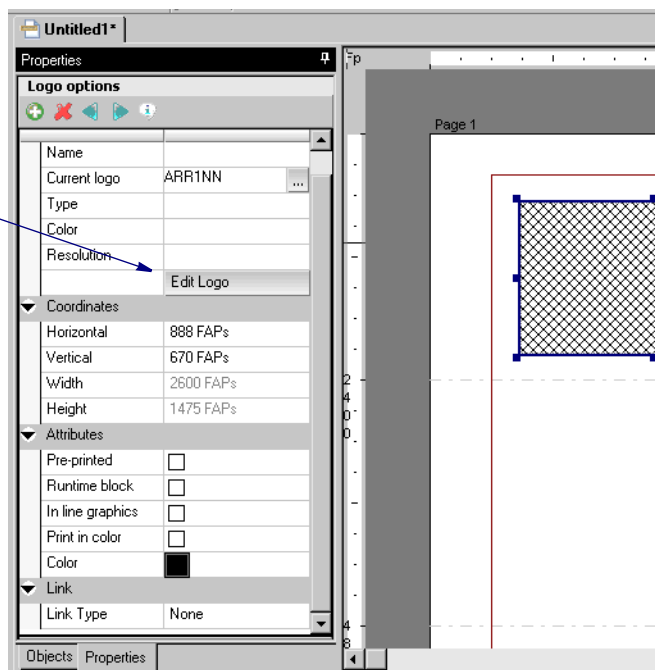
The integer variable, `#_the_integer`, is set the value: 64. The remaining characters are ignored.

1997
RPS

EDITING LOG FILES FROM WITHIN IMAGE MANAGER

While viewing the properties of a LOG file in Image manager, you can now click the Edit Logo button and edit the LOG file. When you click Edit Logo, Studio checks the logo out of the library and opens it using Logo manager.

Click here to check out the LOG file from the library and open it in Logo manager



When you close Logo manager, you return to Image manager.

NOTE: Please note that the Edit Logo button only applies to LOG files. Also note that if you make changes to the LOG file those changes may not immediately appear in the FAP file.

1999
RPS

INCLUDING RDP INFORMATION ON GROUP REPORTS

Group reports now include information on the regional date processing (RDP) rules that affect the group.

The RDP rule
information
appears here

The screenshot shows a software window titled "Group Reports" with a menu bar containing "Print", "Print All", navigation buttons, "Wizard", and "Close". The main content area is titled "COMPANY_LINE OF BUSINESS" and displays two sections: "dalapp1" and "dalappnd". Each section contains a "Triggers" table and an "RDP Rules" table. An arrow points from the text "The RDP rule information appears here" to the "RDP Rules" table for the "dalapp1" group.

Form Name	Form Description	Dec Options	Pre-selected	Filename
dalapp1	dalapp1	None	Yes	dalapp1
dalappnd	dalappnd	None	Yes	dalappnd

dalapp1

Triggers

Type	Detail
trigger test	AGENT(1),HOME OFFICE(1),INSURED(1)

RDP Rules

Name	Date Search Token	Date	State Search Token
Date search	Date search	01/01/1980 - 06/13/2006	Regin Search,Regin Search,Regin Search,Regin Search
Date search	Date search	01/11/1980 - 01/31/1980	Regin Search

dalappnd

Triggers

Type	Detail
test1	AGENT(1),HOME OFFICE(1),INSURED(1)
fdsf	AGENT(1),HOME OFFICE(1),INSURED(1)

Studio includes the following information:

- Name
- Date Search Token
- Date
- State Search Token

You can print the Group report from either the Reports option or from the Report tab which is available while you are working with groups.

NOTE: To set up regional date processing, open the appropriate business unit and choose View, RDP.

2002
RPS

USING SUPERScript AND SUBScript IN TEXT LABELS, TEXT

AREAS, AND VARIABLE FIELDS

You can now use superscript and subscript fonts in text labels, text areas, and variable fields. The system will automatically:

- Use a point size that is one-half of the current point size
- Position the superscript text above a mid-point line for the current font
- Position the subscript text below the mid-point line for the current font

NOTE: The superscript or subscript font you choose must be in your FXR file.

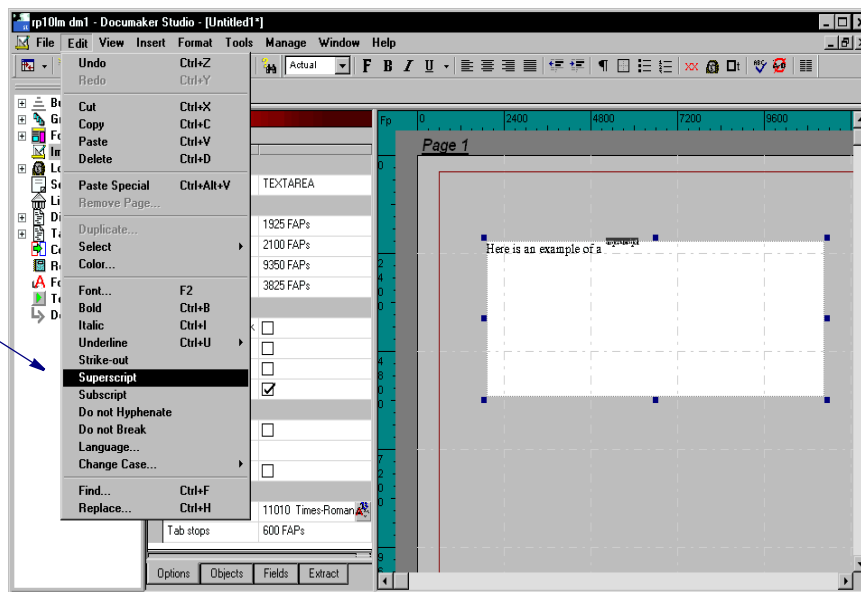
In Text Editor

In Text Editor, select the text you want to superscript or subscript, then choose the appropriate option from the Edit menu. Text Editor tries to choose a smaller font from the FXR using the family name. If a smaller font is not available, the current font is used.

In Studio's Image Manger

In Image manager, select the text you want to superscript or subscript, then choose the appropriate option from the Edit menu. The text is offset from the baseline.

Choose Superscript or
Subscript from the Edit menu



You can also use the Field Options and Text Options to set these styles. The text is offset for display but the grabbers and coordinates reflect the text rectangle without this adjustment. This lets the alignment functions work. The current font is maintained for these objects. There is no attempt to locate a smaller font but the text is offset.

NOTE: The output from the RTF Print Driver may differ from the other drivers. RTF uses subscript and superscript commands to make the text easier to edit. Microsoft Word, however, automatically resizes the text in subscripts or superscripts. All other Documaker print drivers maintain the font specified for the subscript or superscript text.

2003
RPS

DRAGGING AND DROPPING OBJECTS

You can now drag and drop selected objects from one page to another while editing a multi-page FAP in Image manager. You can do this from the object tree and the document design window.

NOTE: Be sure to drag the cursor all the way onto the target page before you drop the object.

2005
RPS

CUSTOMIZING THE LIBRARY MANAGER WINDOW

You can now rearrange the order of the columns (fields) on the Library Manager window. Studio also now remembers the sizes you set the columns. Previously, Studio would reset the columns to their default sizes when you exited.

You can now rearrange
and resize these
columns

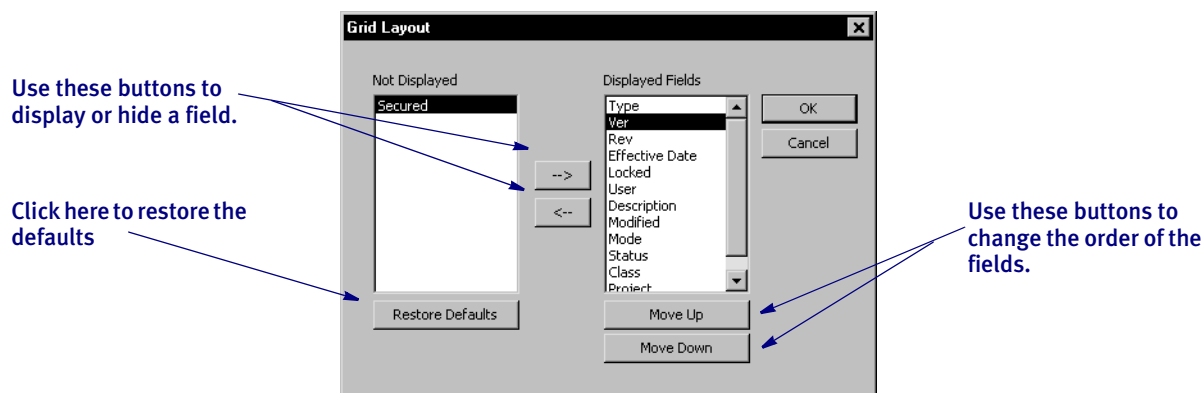
E:\SHIPW32\master\fp10lm dm1\DEFLIB\MASTER.dbf 496 of 496

Name	Type	Ver	Rev	Effective Date	Locked	User	Secured	Description	Modified
RP10LM DM1	BDF	00001	00001	01/01/2003	<input checked="" type="checkbox"/>	MWN	MWN	Initial Check	06/03/2004 12:23:11 P
AFTERW	DAL	00001	00002	03/01/2003		MWN		AFTERW	06/06/2006 05:44:08 P
AFTERW1	DAL	00001	00002	02/19/2003		MWN		AFTERW1	08/27/2004 04:18:35 P
ARCSPLIT	DAL	00001	00002	02/19/2003		MWN		ARCSPLIT	06/06/2006 05:44:18 P
ASCII	DAL	00001	00001	02/19/2003				ASCII	05/27/2004 01:36:20 P
BANNER	DAL	00001	00003	03/15/2003		MWN		banner	08/30/2004 03:35:17 P
BANNER	DAL	00002	00006	03/15/2003		MVF		banner	06/02/2004 10:07:22 A
CAT	DAL	00001	00001	02/19/2003				CAT	05/27/2004 01:36:20 P
COMBINE	DAL	00001	00002	02/19/2003		MWN		COMBINE	08/27/2004 04:18:41 P
EXT2GVM	DAL	00001	00001	02/19/2003				EXT2GVM	05/27/2004 01:36:20 P
GVMEXIST	DAL	00001	00001	02/19/2003				GVMEXIST	05/27/2004 01:36:20 P
HAVE	DAL	00001	00001	02/19/2003		MVF		HAVE	05/27/2004 01:36:20 P
KICKWIP	DAL	00001	00001	02/19/2003				KICKWIP	05/27/2004 01:36:20 P
LM1	DAL	00001	00003	03/15/2003				LM1	06/01/2004 02:43:27 P
LM1	DAL	00002	00003	03/15/2006		MVF		LM1	06/01/2004 02:43:28 P
ONDEMAND	DAL	00001	00001	02/19/2003				ONDEMAND	05/27/2004 01:36:21 P
ONDMDLIB	DAL	00001	00001	02/19/2003				ONDMDLIB	05/27/2004 01:36:21 P
PCL	DAL	00001	00001	02/19/2003				PCL	05/27/2004 01:36:21 P
PREIMAGE	DAL	00001	00001	02/19/2003				PREIMAGE	05/27/2004 01:36:21 P
PRETRAN	DAL	00001	00001	02/19/2003		MVF		PRETRAN	05/27/2004 01:36:21 P
PSTCHKI1	DAL	00001	00001	02/19/2003				PSTCHKI1	05/27/2004 01:36:21 P
PSTCHKI2	DAL	00001	00001	02/19/2003				PSTCHKI2	05/27/2004 01:36:21 P
PSTCHKT1	DAL	00001	00001	02/19/2003				PSTCHKT1	05/27/2004 01:36:21 P

To change the order of the columns, right-click within the grid to display the following menu. Then choose the Grid Layout option.



The Grid Layout window appears.



Here you can change the order of any column except the Name field, which is always shown in the first column. When you move a field from the hidden to the displayed list, Studio adds it to the bottom of the list.

You can also hide fields you do not want to see. Hidden fields are not removed from the grid and you can restore them using the Restore Defaults button on the Grid Layout window.

When you click Ok, the grid reappears with your new settings.

2011
RPS

USING THE NEW PROCESSTRIGGERS RULE

Use this form set level (level 2) rule to process the groups (Key1, Key2 combinations) that exist in the form set, as opposed to only a single set of keys specified in the TRNFILE.DAT file.

NOTE: This rule replaces the LoadRcpTbl and RunSetRcpTbl rules. You can replace those two rules with this one even if you are not using multiple lines of business (Key2s) in your document.

Syntax

```
;ProcessTriggers;;
```

There are no parameters for this rule.

Place this rule after the BuildFormList rule in your AFGJOB.JDT file. Insert this rule after any import rule that might be used to create the starter document. For instance, insert the BuildFormList rule first, followed by your import rule, and then include ProcessTriggers to add additional forms or assign recipient counts to the forms included via the import.

This rule does not trigger the Key2 (lines of business), however, if there are multiple lines of business defined at the point where triggering begins, this rule processes the triggers in each group. One way to define multiple groups is via an import file.

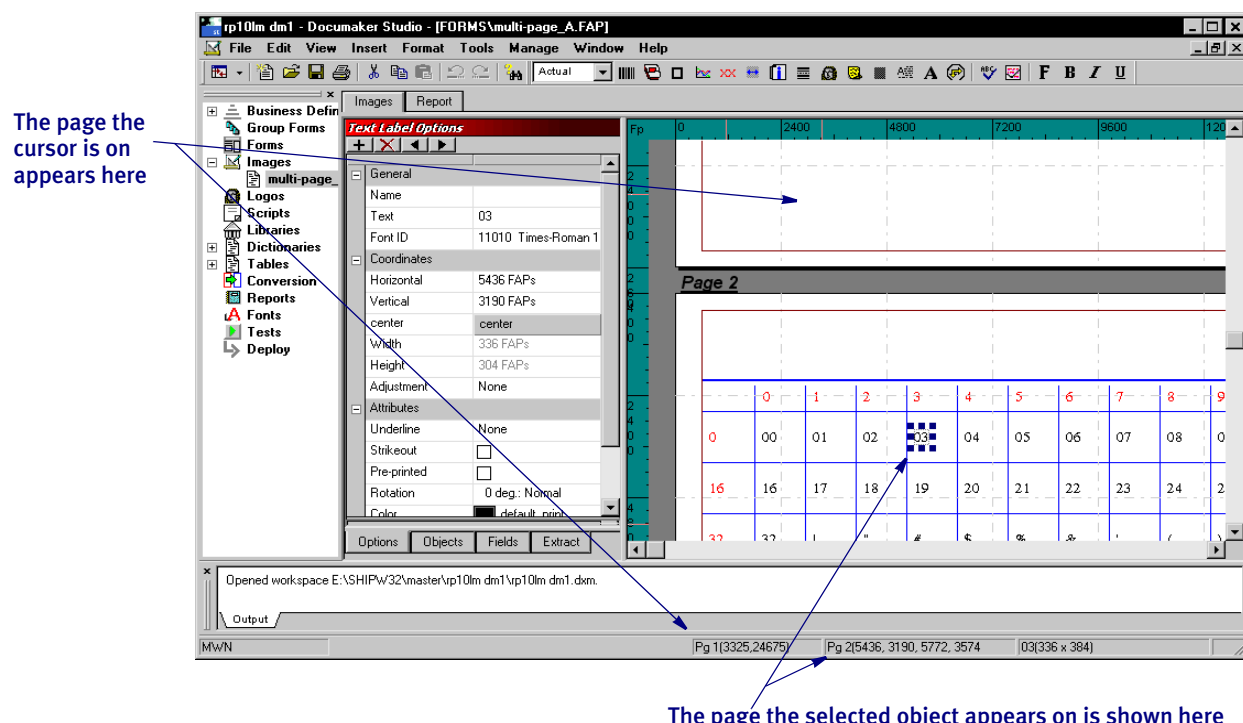
NOTE: The ProcessTriggers rule was added to support multiple lines of business during the triggering process. Normally, the RunSetRcpTbl file only supports the main (Key1+Key2) setting identified with the transaction. The ProcessTriggers rule, however, will process all defined lines of business (each Key1+Key2 combination) at trigger time. This means if you use an import rule to create the transaction, you can have multiple lines of business in batch processing and trigger additional forms and images. You can use the ProcessTriggers rule with both Documaker Studio and the legacy tools.

If you are only using the Studio implementation model, you may want to use the RunTriggers rule described in Feature 1866.

2015
RPS

DISPLAYING PAGE NUMBERS

When working with multi-page images, Studio now displays the page number along with the coordinates on the status bar.



The page information appears regardless of your Zoom setting.

2017
RPS

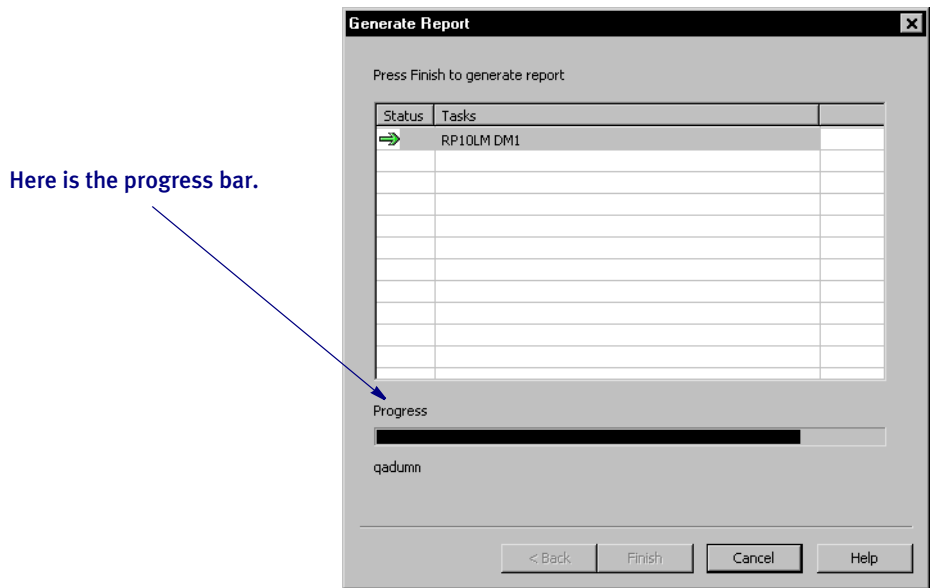
BETTER DISPLAY WHEN RESIZING LINES

Studio now more clearly shows you the end of a line when you are resizing the line in Image manager.

2019
RPS

PROGRESS BARS DURING REPORT GENERATION

Studio now shows a progress bar while generating reports. Most reports are generated almost instantly, but for longer reports this shows you the progress the system is making. Here is an example:

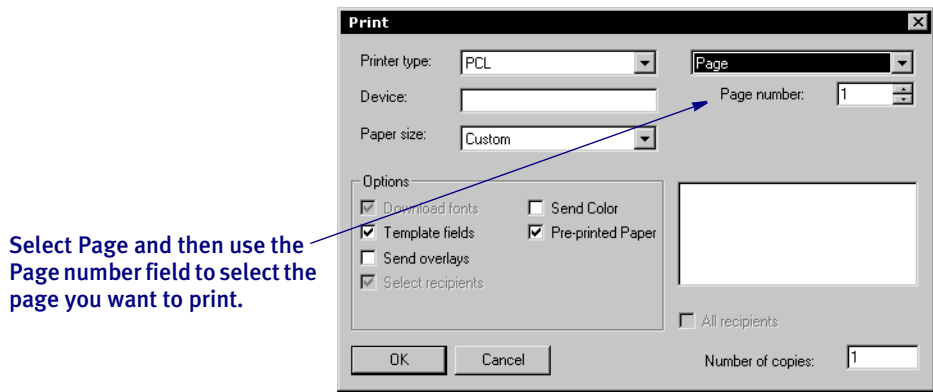


2023
RPS

PRINTING A SPECIFIC PAGE OF A FORM SET

Now you can print a specific page of a multi-page document from any window in which a document is displayed, such as the Image manager.

To print a specific page, open the document in the appropriate view. Right-click in the view and choose the Print option. The Print window appears.



Select the Page option, then select the number of the page you want to print. Click Ok to print that page.

NOTE: You can also choose the File, Print option, then select Document View in the Print Selection window. Then the Print window appears.

2026
RPS

LBYPROC UTILITY ENHANCEMENTS

The LBYPROC utility now supports ****update**** scripts, INI tracing, and has improved return code processing. See the documentation on the LBYPROC utility in the Docutoolbox Reference for more information.

2028
RPS

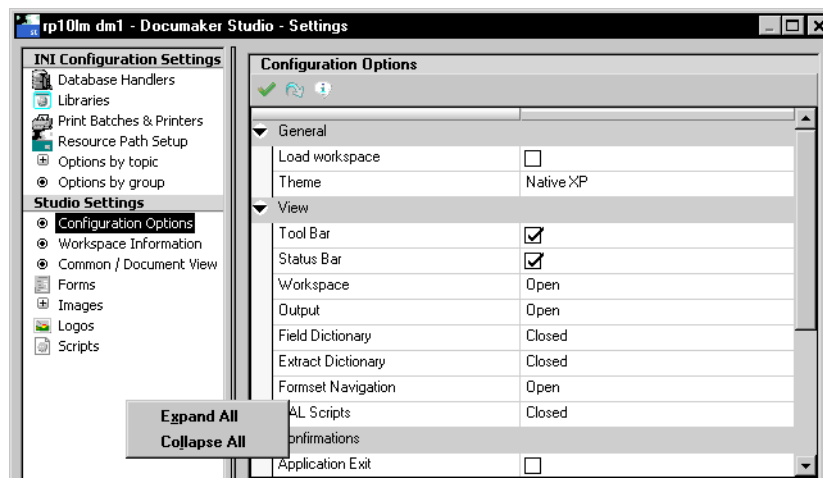
MISCELLANEOUS ENHANCEMENTS TO THE STUDIO INTERFACE

These changes and enhancements were made to Documaker Studio in version 11.2:

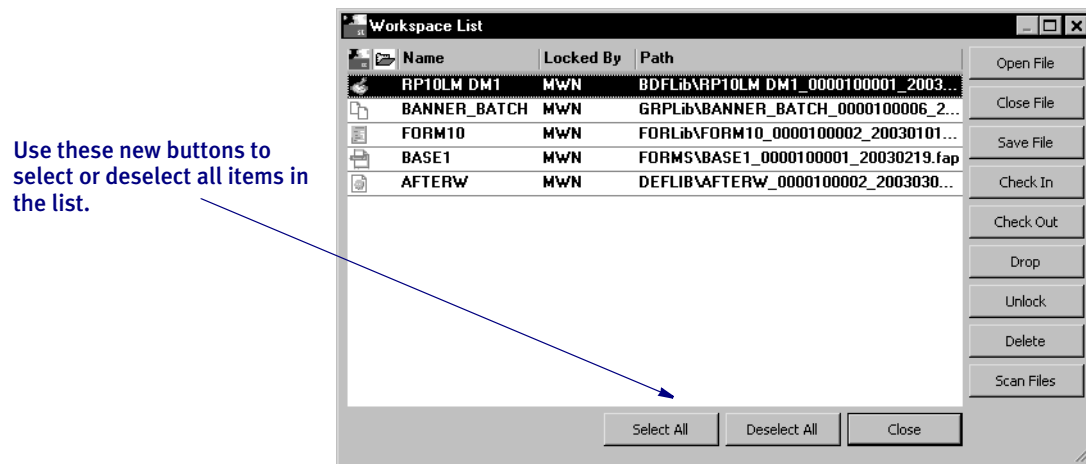
PCR15468 - In the workspace tree, library types browsed from disk now show a different icon from other document types browsed from disk.

PCR 17187 - When printing the Output window, Studio now includes a header and footer on the pages to include the date and page number.

PCR 19003 - You can now right-click in the Settings window (choose Manage, Settings) to display a menu that lets you expand all options or collapse all options.



PCR 19268 - The Workspace List window (choose Manage, Workspace) now includes Select All and Deselect All buttons.



PCR 19881 - To make it easier to restore workspace preferences to the defaults, you can now use this startup parameter to clear the registry settings related to Studio:

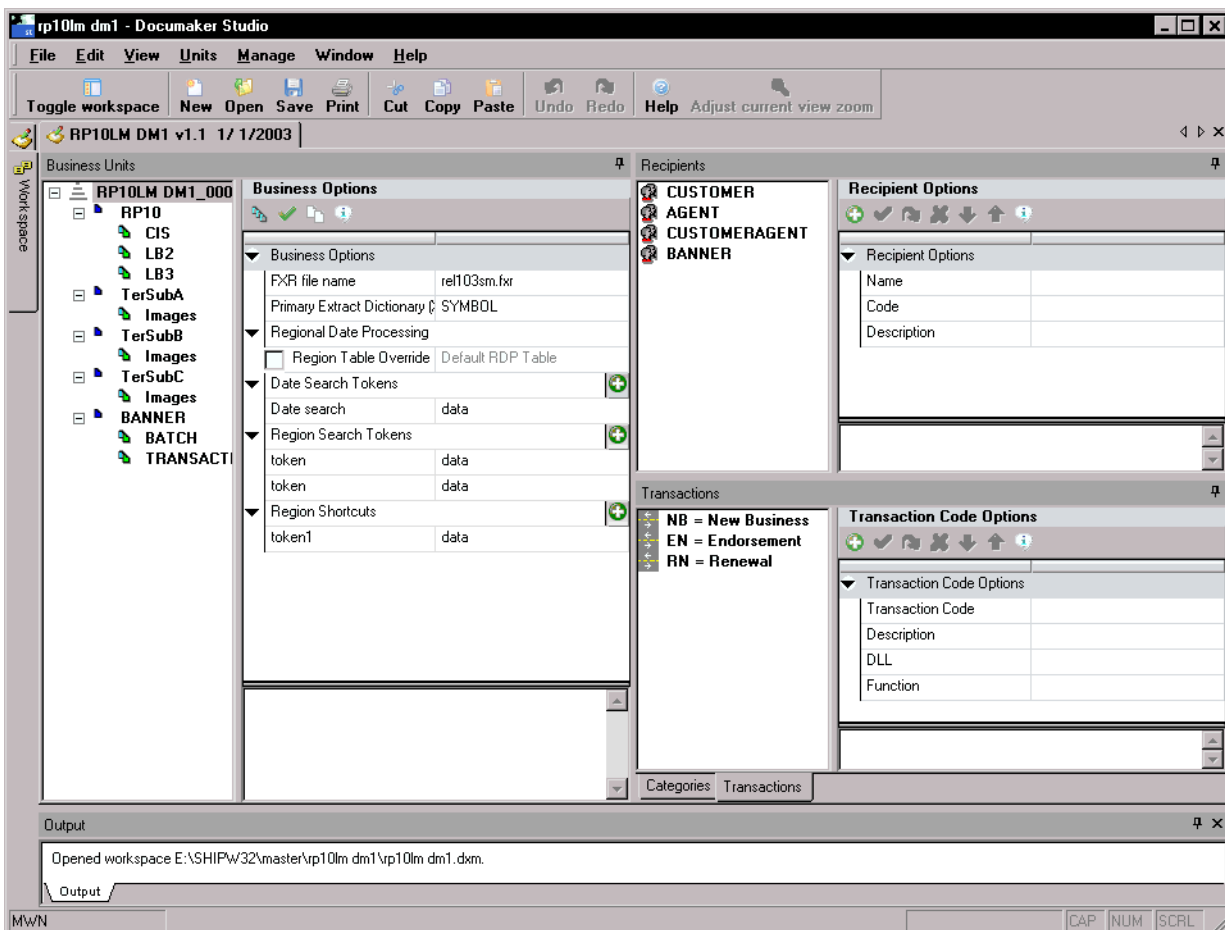
```
DMSTUDIO /CLEAR=ALL
```

If you include this parameter, all preference settings stored in the registry for Studio are removed. Studio will then look just as it did after you initially installed it.

2029
RPS

MISCELLANEOUS BUSINESS DEFINITION MANAGER CHANGES

The Business Definition File (BDF) manager was rearranged to provide a view-at-a-glance look as a default. You can change the way things are organized on the window to suit your needs.



2030
RPS

MISCELLANEOUS ENHANCEMENTS TO STUDIO'S GROUP MANAGER

The following changes and enhancements have been made to Documaker Studio in version 11.2:

- Version 11.2 incorporates the trigger and regional date rule definitions into the main tree. These types of items are no longer presented on a separate view page. When you click on the trigger or RDP rule, the property area now shows the appropriate options for the item selected.

PCR 15951 - Now when you choose a new form by right-clicking in the Group manager, Studio copies the contents of the Name and Description fields into the new form's properties.

PCR 19466 - Triggers are now shown in blue.

2032
RPS

MISCELLANEOUS CHANGES TO STUDIO'S IMAGE MANAGER

The following changes and enhancements have been made to Documaker Studio's Image manager in version 11.2:

PCR 19196 - Two new text area properties have been added in the Defaults section:

These properties have been added.

Text Area Options	
+ X ◀ ▶	
General	
Name	TEXTAREA
Coordinates	
Horizontal	1275 FAPs
Vertical	1525 FAPs
Width	9925 FAPs
Height	8100 FAPs
Sizing	
Can grow and shrink	<input type="checkbox"/>
Can span pages	<input type="checkbox"/>
Must fit on page	<input type="checkbox"/>
Adjust top line	<input checked="" type="checkbox"/>
Attributes	
Hyphenation	<input type="checkbox"/>
Language	
Pre-printed	<input type="checkbox"/>
Default	
Font ID	11010 Times-Roman 10 PT (PCL)
Tab stops	600 FAPs

Options Objects Fields Extract

- The Font ID property identifies the default font ID to use.
- The Tab Stops property defines the default interval for tab stops within this text area.

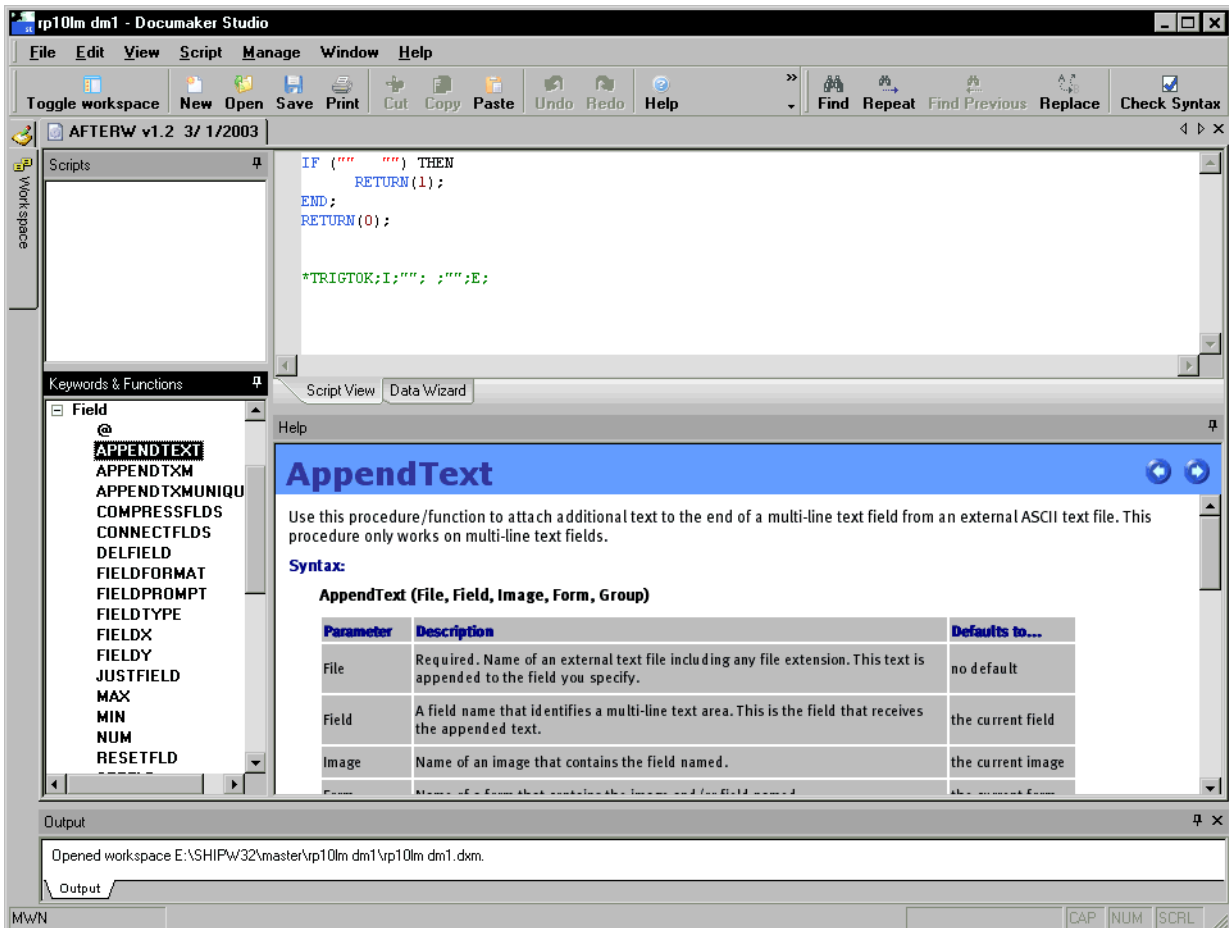
Both let you define default values available when you first create the text area and you can later override either setting as you format the paragraph. Changing the defaults on a text area that already includes paragraph text does not affect that text.

2034
RPS

MISCELLANEOUS CHANGES TO STUDIO'S SCRIPT MANAGER

The following changes and enhancements have been made to Documaker Studio's Script manager in version 11.2:

PCR 19460 - Studio now includes Help information in the Script manager for the individual DAL functions. As you click on the available function names or keywords, you see the Help information for that function or keyword.



PCR 19229 - The header and footer on the Script report are now consistent with reports from other managers.

2035
RPS

MISCELLANEOUS CHANGES TO STUDIO'S LIBRARY MANAGER

The following changes and enhancements have been made to Documaker Studio's Library manager in version 11.2:

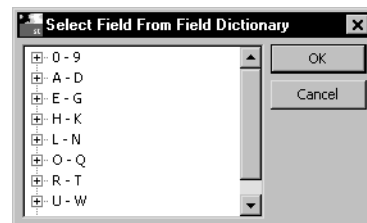
Now you can right-click in Library manager and select Stop Loading to make Studio stop loading library records. This can be useful if you enter a search or filter criteria in error and you do not want to continue the process.

If you select Stop Loading to stop the process, click Filter Now or Search to start over using the current criteria. Keep in mind, you can change criteria while the system is loading items and then click Filter Now or Search to have Studio immediately begin a new search with the new criteria.

NOTE: You do not have to wait for Studio to load all items before you can select an item and begin to work with it.

PCR 16849 - The display changes as you move the horizontal scroll bar left or right in the library grid. Previously, you had to release the mouse button to have the display change to reflect the new position of the scroll bar.

PCR 19198 - Now if you are searching on a field, you can select a field from the field database (FDB) in addition to entering the field's name.



PCR 19836 -When creating a new workspace and choosing ODBC for your library database type, the system now creates the LBYLOG file for you using the same ODBC settings. This saves you from having to create this manually in Settings after the workspace is created.

2036
RPS

MISCELLANEOUS CHANGES TO STUDIO'S DICTIONARY MANAGERS

The following changes and enhancements have been made to Documaker Studio's Dictionary managers in version 11.2:

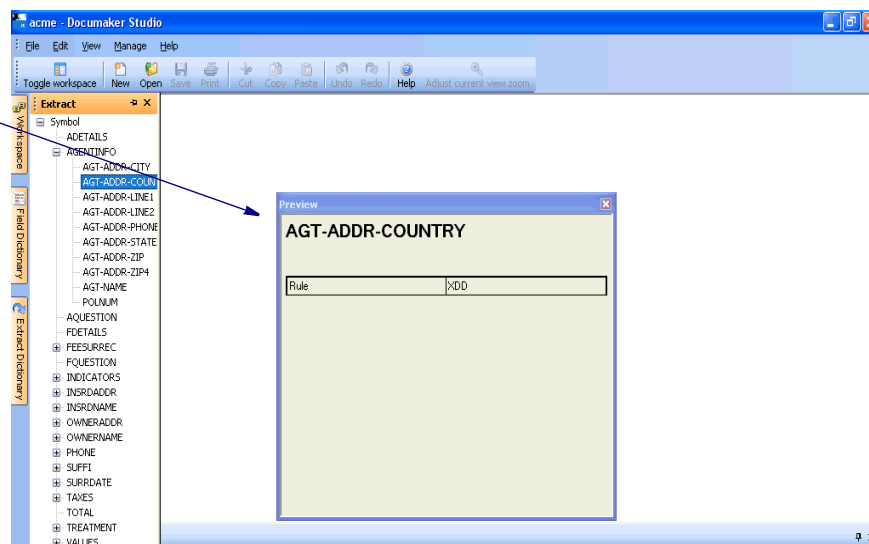
PCR15832 - Extract Dictionary Manager now offers two new menu options when you right-click on a field or record parent in the tree:

- Duplicate
- Duplicate with Children

These options let you duplicate the item or duplicate the item with children. The Duplicate with Children option copies not only the item but also any child (descendant) items that occur below it.

PCR 20119 - Now you can select an item in the Field Dictionary tree and the Extract Dictionary tree, right click and choose the Preview option to see a Preview window. This window shows basic information about the item.

Here is an example of a Preview window from the Extract Dictionary.



2038
RPS

MISCELLANEOUS CHANGES TO STUDIO'S CONVERSION MANAGER

The following changes and enhancements have been made to Documaker Studio's Conversion manager in version 11.2:

PCR 18497 - The following message now appears in the output area if the you click Cancel during a conversion:

```
"--- Conversion Cancelled ---"
```

2039
RPS

MISCELLANEOUS CHANGES TO STUDIO'S REPORTS MANAGER

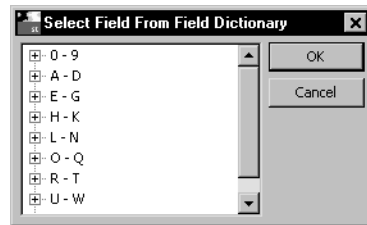
The following changes and enhancements have been made to Documaker Studio's Reports manager in version 11.2:

The FDB report is now cross-referenced with the XDD and vice-versa. The column indicates whether there is a field of like name in the base XDD defined by the workspace.

PCR 15688 - You can now perform a Save As for the various reports in Report manager.

PCR 19299 - You can now look up library member names for the Impact report.

PCR 19198 - Now if you are searching for references to a field for the Impact report, you can select the field from the field database (FDB) in addition to entering the field's name.



2041
RPS

MISCELLANEOUS CHANGES TO STUDIO'S TEST MANAGER

The following changes and enhancements have been made to Documaker Studio's Test manager in version 11.2:

PCR 16054 - The Test manager now automatically selects the Errors window to come forward if an error occurs during the test.

PCR 19391 - The default file extension is now XML when you click Browse for XML extract files.

2042
RPS

MISCELLANEOUS ENHANCEMENTS TO DEPLOYMENT MANAGER

The following changes and enhancements have been made to Documaker Studio in version 11.2:

PCR 19219 - You can use the new Save and Exit button to save the deployment parameters you have set up without actually running the deployment.

PCR 19492 - The system now pauses while Pre- and Post-execution processes are executed.

2045
RPS

NEW DEFAULT FOR THE CHECKNEXTRECIP OPTION

The CheckNextRecip option enables recipient batch records for the same transaction that occur in sequence in a recipient batch file to be queued and printed in one call to the print driver. This enables the transaction to be loaded only once, and then each recipient is processed in turn inside the print driver.

The default for the this option has been changed from Yes to No.

```
< RunMode >  
    CheckNextRecip = No
```


This change will make new implementations easier. The previous default conflicted with some features — like individual mailer pages — and would often have to be changed. The new default minimizes the likelihood you would have to change the option's setting.

For instance, this option must be set to No (now the default) when you are using the GenPrint program with print drivers that must print to a separate file for each recipient, such as the PDF, XML, HTML, and RTF print drivers. This change makes it easier to use those print drivers.

NOTE: You can still set this option to Yes and some environments may see a marginal performance benefit with this option enabled. If you prefer to have this option enabled and have all recipients printed together, be sure you add both the option and the Yes setting into your INI file.

```
< RunMode >
    CheckNextRecip = Yes
```

Otherwise, the system's new default (No) will govern the GenPrint program's behavior.

2061
RPS

USING THE NEW BDF2FDT UTILITY

Use this utility to load a complete tree based on the specified BDF file and then convert it to a pre-version 11.x format FORM.DAT file.

NOTE: This utility is typically used by customers who have purchased tools from 3rd party vendors that parse FORM.DAT files. These tools extract a list of the forms in the library as a part of the data exchange mapping process.

Program names

Windows BDF2FDT.EXE

Syntax

BDF2FDT /I /INI /O /D

Parameter	Description
/I	Optional. Enter the name of the BDF file you want to convert. If you omit this parameter, the system looks for the one in the INI file you specify using the INI parameter.

Parameter	Description
/INI	Optional. Enter the name of the INI file which contains the name of the BDF file you want to convert. The default is FSIUSER.INI.
/O	Optional. Enter the name of the output FORM.DAT file. The default is FORM.DAT.
/D	Optional. Enter the effective date in YYYYMMDD format. The default is the current date.

NOTE: Not all document features and options supported by Documaker Studio can be represented in a FORM.DAT file. You must make sure the forms you develop in Studio convert appropriately.

2063
RPS

CONVERTING EPS FILES INTO LOG FILES

You can now convert Encapsulated PostScript (EPS) images into Docucorp logo (LOG) files. This feature is available when you are working with logos using Logo Manager in either Documaker Studio or Docucreate and with the Conversion wizard in Documaker Studio.

To convert EPS files into LOG files, you must have AFPL Ghostscript 8.00 or higher installed. This free application is required to load EPS files.

NOTE: There are several variants of Ghostscript, such as GPL Ghostscript, GNU Ghostscript, ESP Ghostscript, and AFPL Ghostscript. This feature requires AFPL Ghostscript and will not work with the other versions of Ghostscript. For more information on Ghostscript, visit this web site:

<http://www.cs.wisc.edu/~ghost/>

2064
RPS

THE AFP LOADER NOW USES CHARACTER SET AND CODE PAGE FONT INFORMATION

When loading AFP, the system now uses the information in the new Character Set and Code Page Font fields in the FXR file (see Feature 1990) instead of using the font information contained in the IBMXREF.TBL.

The AFP loader expects the AFP file's Map Coded Font (MCF) structured fields to contain references to AFP coded fonts. However, MCF structured fields can contain character set and code page information instead of the coded font information the FXR file requires.

Before this enhancement, for MCF structured fields that contained character set and code page information instead of coded fonts, you had to manually set up the IBMXREF.TBL file to resolve the character set/code page information to coded fonts in the FXR file.

Since the system now includes character set and code page information in the FXR file, the AFP loader first checks the FXR file for this information and, if it exists, uses it. If the information does not exist, the AFP loader loads the information from the IBMXREF.TBL file.

2067
RPS

PROMPTING FOR SEARCH CRITERIA

In Documaker Workstation, the system now prompts you to enter at least one search criteria on the Retrieval window if you click Ok without entering any search criteria. This prevents you from inadvertently searching the entire archive.

2068
RPS

CREATING SMALLER POSTSCRIPT OUTPUT

The PostScript Print Driver now automatically downloads (embeds) only the fonts that are needed. This will result in smaller output files.

NOTE: To produce a PostScript print stream that only downloads (embeds) the minimum set of fonts required by the PostScript print stream, you *cannot* use overlays.

All PostScript fonts referenced in the FXR file are downloaded if the SendOverlays option is set to Yes because the system does not know which fonts are used by the overlays.

You must set the following PostScript INI options as shown to tell the PostScript Print Driver to download the minimum set of fonts required by a print stream:

```
< PrtType:PST >
  DownloadFonts = Yes
  SendOverlays = No
```

If you are running the GenPrint program, you will need to tell GenPrint to load the FAP files (instead of overlays) by using the DownloadFAP option:

```
< RunMode >
  DownloadFAP = Yes
```

2072
RPS

SPECIFYING A DFD FILE FOR THE LIBRARY DATA TABLE

A Data Format Definition (DFD) file is used to describe the layout of the fields in another file. The description includes the name, data type, and length of each field in the file.

Several files used by Documaker adhere to pre-defined internal structure definitions so there is usually no need to create and specify DFD files for those files. The library's data and index portions are two examples of files that usually adhere to a pre-defined internal structure.

There are cases, however, when you must specify the structure of the data portion of the library using a DFD file. In prior versions, you made this DFD specification via the CARFileDFD option. For example, if your library data table was called *LBYP* and the DFD was named *CARDB2.DFD*, you would have specified the DFD as shown here:

```
< ArcRet >
CARFileDFD = DEFLIB\cardb2.dfd
```

This CarFileDFD option, however, is also used by the GenArc program and if a different value was needed for archival purposes, you had to maintain a different INI file for that process. Now, instead of using the CARFileDFD option, you can specify the library data file's DFD using the DFD option, as shown here:

```
< DBTable:LBYP >
DBHandler   = DB2
DFD         = DEFLIB\LBYP_DB2.dfd
```

This DFD option does not conflict with the CARFileDFD option used by the GenArc program.

2073
RPS

SPECIFYING A CATALOG TABLE FOR EACH LIBRARY

Now you can specify a catalog table for each library stored in a database management system (DBMS). If you do this, Studio will only update the catalog table once per library per session, which reduces the chances of you receiving the duplicate index or -803 SQL error message.

You specify the catalog table using INI options. Here is an example:

```
< Library:LBYP >
Catalog       = LBYC
< DBTable:LBYC >
DBHandler     = DB2
< DB2_FileConvert >
Catalog       = DAP112_LBYC_R1
```

2077
RPS

SUPPRESSING LOG FILES WHEN CONVERTING AFP TO FAP OR

PDF

You can use the new SuppressLogoUnload option when converting AFP files into FAP or PDF format to avoid possible conversion errors which can arise from converting the same AFP file multiple times.

```
< PrtType:AFP >
    SuppressLogoUnload = Yes
```

Option	Description
SuppressLogoUnload	Enter Yes to suppress the unloading of logo (LOG) files during a conversion of AFP files to FAP or PDF format. You should always set this option to yes when creating PDF files. The default is No.

When you enter Yes, AFP to PDF conversions, whether run from the MRG2FAP utility or IDS, do not result in new LOG files being unloaded from AFP page segments. The LOG files were unloaded for use by FAP files and are not needed for the conversion to PDF format.

Unless you need to look at or use the FAP files, set this option to Yes. If you do need the FAP files, you must also delete all LOG files before you run the MRG2FAP utility to make sure you are getting the intended results.

NOTE: This option does not delete any LOG files created by a prior conversion, so you should delete all LOG files before you use the new INI option and this feature.

2080
RPS

OUTPUTTING AFP CHARACTER SET AND CODE PAGE INFORMATION

Now you can output the AFP character set and code page combination instead of the coded font in the font list when you generate normalized AFP files.

If you want the character set/code page combinations to be output in the AFP font list, instead of the coded fonts, you must add the DisplayCodedFont option, as shown here:

```
< PrtType: AFP >
    DisplayCodedFont = No
```

Option	Description
DisplayCodedFont	Enter No to include the character set/code page combinations in the AFP font list, instead of the coded fonts. The defaults is Yes, which tells the system to include the coded fonts.

Keep in mind the FXR file must contain the character set and code page entries in the AFP font record for this option to work. If you set the INI option to No and the character set and code page entries are not in the FXR file, the font list in the AFP file will contain only the coded fonts.

NOTE: The AFP output record can only contain *either* coded fonts *or* character set/code page entries — it cannot contain a combination. It will default to coded fonts for all if the font for one or more objects does not contain character set/code page entries.

2085
RPS

SUPPORT FOR VERSION 1 XEROX FRM FILES

This feature adds conversion support for older, version 1 Xerox FRM files. Previously, only version 2 Xerox FRM files were supported. This change affects the FRM2FAP utility, the FRMDUMP utility, and Documaker Studio's conversion feature. In all cases the system automatically detects and supports the version of Xerox FRM file it encounters without you having to take any action.

2086
RPS

ENHANCED ERROR CHECKING FOR THE MRGCHK UTILITY

Now the MRGCHK utility can detect Metacode files that have improper formats, such as if the files are not 4-byte variable blocked (MRG4 format). The utility now issues an error if such a file is detected. Previously no error message appeared.

2087
RPS

MARKING OBJECTS AS PRE-PRINTED

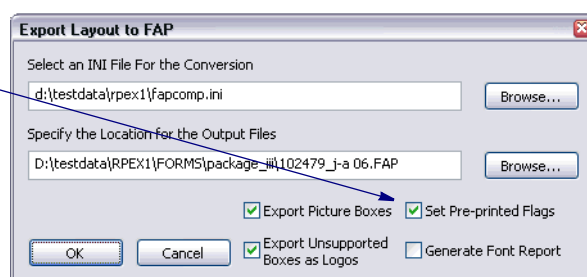
This feature adds the Set Pre-Printed Flags option to the QuarkXPress Documaker XTension's Export to FAP window. You can use this option to tell the XTension to set the pre-printed flag on every object (boxes, bitmaps, vectors, text areas, and text labels) it creates.

Having the pre-printed flag set on an object can be useful during forms creation or when test printing your forms, such as when you are using letterhead or special paper with the company name or logo embossed on the paper. Inserting a text label or logo on your form in the same location as the company name or logo helps make sure the space is not inadvertently written over by other objects. These objects would reserve that space on the form. You may want to include pre-printed objects when test printing the form to compare against your special paper to make sure there is enough space on the special paper for the non pre-printed objects.

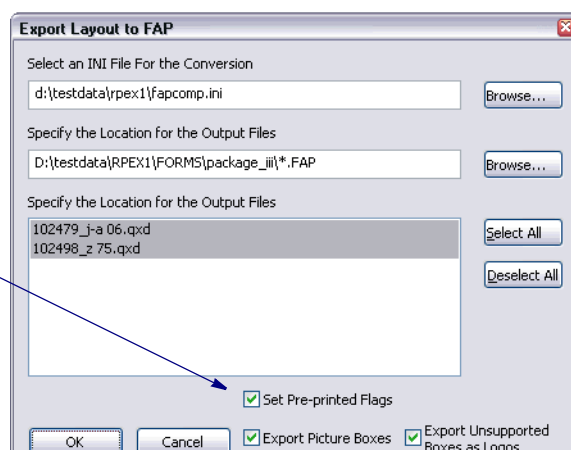
By default, the Print window does not display a Pre-printed option. Any objects flagged as pre-printed will not print on the form.

NOTE: This feature was released in version 11.1 in patch 13.

Click this option



Click this option



In addition, this feature enhances the conversion of Encapsulated PostScript images from QuarkXPress into Docucorp logo files (LOG). Previously, conversion was done at 96 dpi. Now the conversion is done at 300 dpi.

NOTE: In addition to ICUDT21.dll and ICUUC21.dll files, you must now place the LIBGFL254.dll file in the QuarkXPress XTension directory.

2088
RPS

ENHANCED DAL PROFILE OUTPUT

In addition to DAL profile information which includes the time spent per function (DAL subroutine), the system now outputs information into the trace file about the total time spent in each function and number of times each function is called.

NOTE: It was possible in prior releases to determine this information from profile output but you had to load the log file into a spreadsheet and perform calculations.

An example of the additional information is shown below. This example is from a GenData run which processed 600 transactions. The total processing time was 23 seconds. Only the beginning of the log is shown because of space considerations.

The log is sorted by the cumulative time spent in each script with longest running scripts at the top. The log information appears in the trace file and is written out when the program terminates.

You will find this information appears in the log:

Item	Description
Executed XXX	The number of times script was executed.
Cumulative run time X.XXX	The time in seconds dot milliseconds spent in this script and all scripts/ code that was executed from this script.
Compiled or Non-compiled	Whether or not the script was compiled.
Name	The name of the script or the actual script if it was not in an external file.

Some scripts look like they are listed twice, but are not. For instance, in the example below PostTrans_Prod() and PostTrans_Prod actually are the script that had a call to PostTrans_Prod (all it had was "PostTrans_Prod()") and the actual PostTrans_Prod DAL subroutine.

When you analyze the log, keep these things in mind:

- The scripts you need to review are usually the scripts at the top of the log.
- Review any scripts that are executed more times than number of transactions in the run. You can probably modify your implementation so the script is run no more than once per transaction or once per job.
- Review the scripts that run the longest and see if they can be optimized. For example, move assignment of variables outside the loop. Consider parts that can be executed only when needed.
- Typically, scripts that take longer to run or receive a higher number of calls are good candidates for review of either the script itself or the implementation.

- Clock resolution is set at one millisecond. If a script executes in less than one millisecond, the time spent equals zero (0). Scripts that show a high number of calls, even if the time is shown as zero (0), or a relatively small number are good candidates for optimization.

NOTE: This feature is designed to help you analyze DAL performance. The extra logging does affect total time spent executing the program being analyzed and should not be turned on in a production environment or left on when not needed.

```

Executed 600 times Cumulative run time 2.840 Non-compiled Script
PostTrans_Prod()
Executed 600 times Cumulative run time 2.824 Compiled Script
PostTrans_Prod
Executed 600 times Cumulative run time 2.451 Non-compiled Script
PREFILL_VARS()
Executed 600 times Cumulative run time 2.420 Compiled Script
PREFILL_VARS
Executed 600 times Cumulative run time 1.954 Compiled Script
DEFLIB\BarCode.DAL
Executed 534 times Cumulative run time 0.792 Compiled Script
DEFLIB\Delete_Images.DAL
Executed 1150 times Cumulative run time 0.784 Non-compiled Script
CALL("SERVPHONENUM")
Executed 1150 times Cumulative run time 0.737 Compiled Script
DEFLIB\SERVPHONENUM.DAL
Executed 600 times Cumulative run time 0.372 Non-compiled Script
COPYCOUNT()
Executed 1813 times Cumulative run time 0.359 Non-compiled Script
call("INSUREDNAME1")
Executed 1813 times Cumulative run time 0.312 Compiled Script
DEFLIB\INSUREDNAME1.DAL
Executed 600 times Cumulative run time 0.295 Compiled Script
COPYCOUNT
Executed 1180 times Cumulative run time 0.234 Non-compiled Script
call("INSUREDNAME2")
Executed 1200 times Cumulative run time 0.205 Non-compiled Script
call("BROKERNAMELIT")
Executed 1180 times Cumulative run time 0.203 Compiled Script
DEFLIB\INSUREDNAME2.DAL
Executed 567 times Cumulative run time 0.186 Non-compiled Script
Return ((?("POL.NUM.LIT")) & " " & (?("INS.POL.NUM")) &
(?("INS.POL.YREFF")))
Executed 1200 times Cumulative run time 0.186 Non-compiled Script
Call("DMGMERGESETID")
Executed 1137 times Cumulative run time 0.173 Non-compiled Script
call("POLEFFDATE")
Executed 534 times Cumulative run time 0.159 Non-compiled Script
MSGB03A()
Executed 534 times Cumulative run time 0.158 Non-compiled Script
MSGD12A1()
Executed 600 times Cumulative run time 0.158 Non-compiled Script
CALL("SERVADDR1DAL")
Executed 534 times Cumulative run time 0.142 Non-compiled Script
MSG04A()
Executed 534 times Cumulative run time 0.141 Non-compiled Script
MSGB07B()

```

```
Executed 1137 times Cumulative run time 0.139 Non-compiled Script  
call("POLEXPDATE")  
Executed 534 times Cumulative run time 0.126 Non-compiled Script  
MSG09B()  
Executed 1149 times Cumulative run time 0.126 Non-compiled Script  
call("DUEDATE")  
Executed 534 times Cumulative run time 0.126 Non-compiled Script  
MSGB11A()  
Executed 550 times Cumulative run time 0.126 Compiled Script  
DEFLIB\UPDATESCANABLE.DAL  
Executed 600 times Cumulative run time 0.125 Non-compiled Script  
CALL("SERVADDR3DAL")  
Executed 534 times Cumulative run time 0.125 Compiled Script  
DEFLIB\WITHDRBILLDAY2.DAL  
Executed 534 times Cumulative run time 0.125 Non-compiled Script  
CALL("WITHDRBILLDAY2")  
Executed 534 times Cumulative run time 0.125 Non-compiled Script  
MSGM11A()  
Executed 534 times Cumulative run time 0.124 Non-compiled Script  
MSGD12A3()  
Executed 1200 times Cumulative run time 0.124 Compiled Script  
DEFLIB\DMGMERGESETID.DAL  
Executed 534 times Cumulative run time 0.124 Non-compiled Script  
MSG08A()  
Executed 1137 times Cumulative run time 0.123 Compiled Script  
DEFLIB\POLEXPDATE.DAL  
Executed 534 times Cumulative run time 0.111 Non-compiled Script  
MSGC01A()  
Executed 534 times Cumulative run time 0.111 Compiled Script MSGB03A  
Executed 534 times Cumulative run time 0.110 Non-compiled Script  
MSGM07A()  
Executed 570 times Cumulative run time 0.110 Non-compiled Script  
call("COMPANYNAMELIT")  
Executed 534 times Cumulative run time 0.110 Non-compiled Script  
MSGD10C()  
Executed 534 times Cumulative run time 0.110 Non-compiled Script  
MSGM02A()  
Executed 600 times Cumulative run time 0.110 Non-compiled Script  
CALL("SERVADDR2DAL")  
Executed 534 times Cumulative run time 0.110 Compiled Script MSGD12A1  
Executed 534 times Cumulative run time 0.110 Non-compiled Script  
MSGD10G()  
Executed 600 times Cumulative run time 0.109 Non-compiled Script  
CALL("DMGTOTALSHEETS")
```

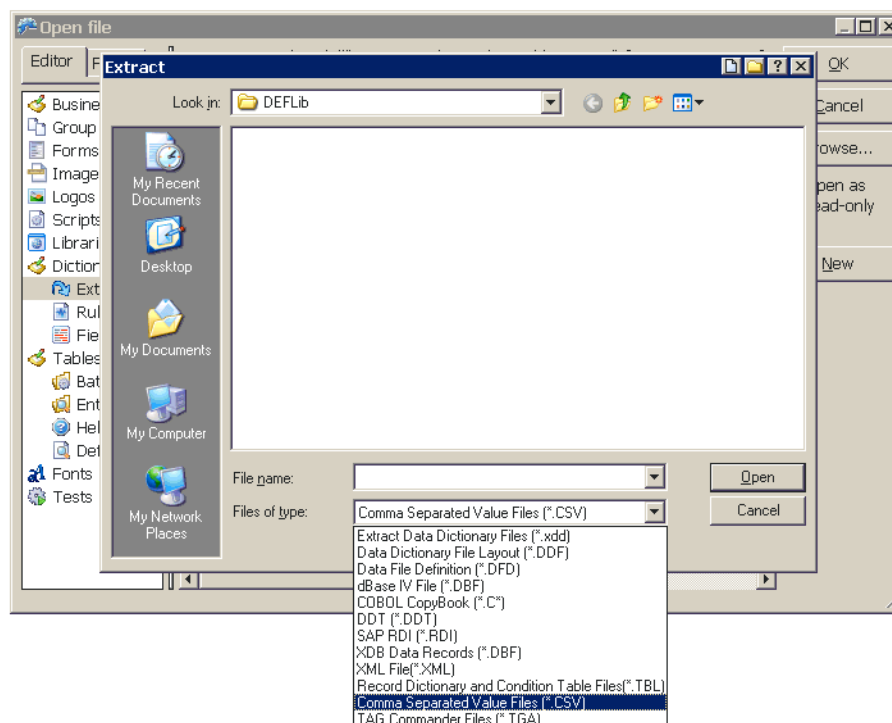
2091
RPS

IMPORTING CSV FILES INTO AN EXTRACT DICTIONARY

To make it easier to convert a non-Documaker dictionary into a Documaker extract (XDD) dictionary, you can now import comma-separated value (CSV) files.

To import the contents of a CSV file into a Documaker extract dictionary, open the extract dictionary, or create a new extract dictionary, then choose the File, Import option. The Open File window appears.

Click Browse and select the Comma-Separated Values Files (*.CSV) file type in the Files of Type field, as shown here:



NOTE: This format is already supported for the fields dictionary (FDB).

Creating a CSV File

The first record in a CSV file must be a header record that contains column names. Without this there is no way to determine how the columns should match those in the Extract Dictionary database. The header row is encoded just like any other CSV record in accordance with the rules stated below. The Extract Dictionary database contains a number of columns, but you do not have to provide a match for all columns.

Only the Name column is required. The order of the columns within the CSV file is not important. Any column named in the header record that does not have a corresponding column in the Extract Dictionary database is ignored.

Here is a list of the column names in the Extract Dictionary database:

```
Parent
Name
DataLength
Offset
XPath
CondFlag
MultiOccur
Type
Flags
Format
CusFormatFlag
Rule
Mask
Prompt
Data
OvflMulti
SrcOvfl
```

With the exception of the Format column, you can identify the type of data each column accepts by reviewing the contents of the property area in the Extract Dictionary manager. The Format column is reserved for future use.

Here are some rules on formatting a CSV file:

- Each record is one line of text followed by a record separator. This separator may be a line feed, or a carriage return and line feed pair. Records may not span more than one line.
- Fields are separated by commas.
- Leading and trailing spaces or tabs adjacent to comma field separators are ignored.
- Fields that contain quotation marks (") must be surrounded by quotation marks and the embedded quotation marks must each be represented by a pair of consecutive quotation marks. For example, John "Q" Doe would convert to "John ""Q"" Doe".
- Quotation marks (") are required if the data contains a comma.
- The CSV file format is used to transmit textual data. This means individual data values cannot contain any binary or other non-text components. There is no way to include binary data.

Here is an example of a CSV file:

```
NAME, FONTID, LENGTH, ENTRYTYPE, FORMAT
CheckBox, 16010, 25, m,
DATE, 16010, 25, d, 1/4
"Insured Name", 11010, 22, x,
Charges, 16010, 20, n, "$zz, zz9.99"
```

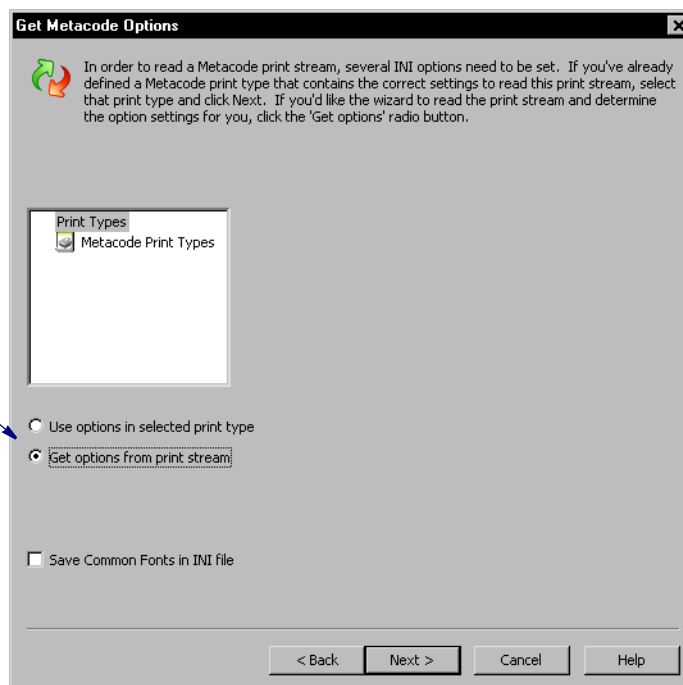
COMMENTRECORDS OPTIONS

The Get Metacode Options page of Studio's Metacode conversion wizard now lets you get the JDEName, JDLName, CodeDef, and CommentRecords options from the normalized Metacode file.

NOTE: Previously, the Metacode conversion wizard only got the OutMode, DJDEIden, DJDEOffset, DJDESkip, and JDLCCode options from the normalized Metacode file.

You can then save these values into the corresponding INI options in the `PrtType:XER` control group where they can be used by the system when it reads and writes to normalized Metacode files.

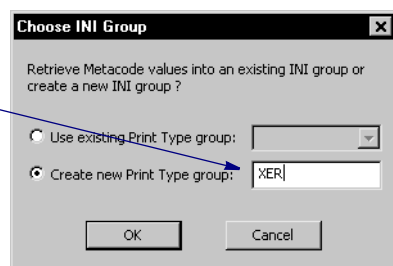
Click here to get the JDEName, JDLName, CodeDef, and CommentRecords options from the normalized Metacode file.



Studio prompts you for the name of the INI control group into which you want the values placed.

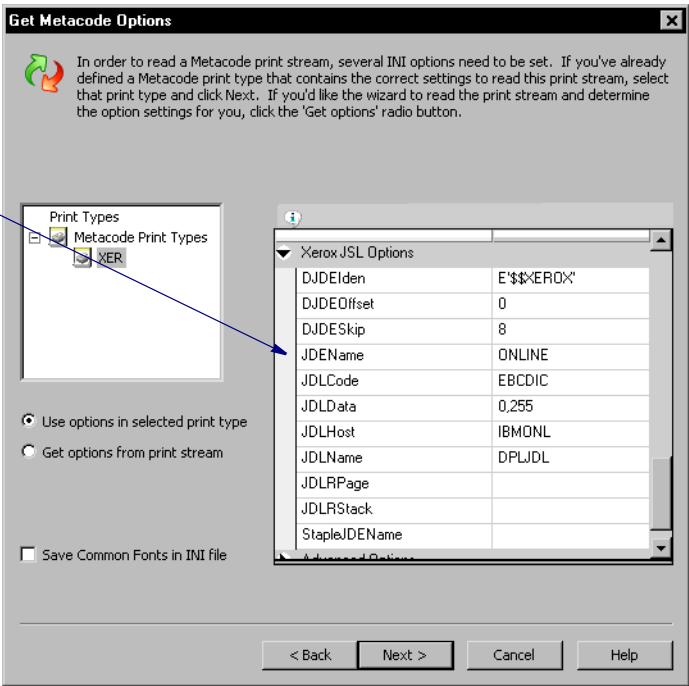
You can select a print type control group or set up a new one, as shown here.

Note that Studio adds the *PrtType:* for you. For instance, this entry would be recorded in your INI file as *PrtType:XER*.



When you click Ok, Studio retrieves the values:

Here are the values Studio retrieved from the normalized Metacode file.



2096
RPS

USING THE NEW FONTLIST UTILITY

Use the new FONTLIST utility to create a report which lists the fonts used in the specified FAP files. This report includes information on...

- The font IDs used in each FAP file
- The font IDs used in all specified FAP files
- A list of the font IDs contained in the FXR file but not used in the FAP files
- A list of all FAP files that use a specific font ID (optional)

Syntax

FontList /I /X /O /F

Parameter	Description
/I	Enter the name of the FAP file from which you want to compile a list of fonts. You can use wild cards (*) in the name to specify multiple files.
/X	Enter the name of the font cross-reference (FXR) file to search.
/O	Enter the name you want to assign to the report. The default report name is FONTLIST.DAT.
/F	Optional. Enter a specific font ID to search for.

Example

For example, this command:

```
fontlist /i=forms\c* /x=deflib\rel102sm /o=.\fonts.txt /f=11020
```

would produce a report that looks similar to this one:

```
--- FONTLIST Copyright (C) 1997-2006 Docucorp International
--- Produces a font usage report

FontList Utility

FAP: D:\FAP\mstrres\RPEX1\forms\c*
FXR: D:\FAP\mstrres\RPEX1\deflib\rel102sm

FAP and Font ID Report
-----

Fap: CODE128
12012          Courier 12 PT

Fap: Checkbox
11010          Times-Roman 10 PT
11020          Times-Roman 20 PT

Fap: Color
11020          Times-Roman 20 PT

Fap: cuscolor
16016          Univers-Medium 16 PT

Fap Count: 0000000005

Consolidated List of Fonts Report
-----
11010          Times-Roman 10 PT
11020          Times-Roman 20 PT
12012          Courier 12 PT
16016          Univers-Medium 16 PT

Fap List for Font ID 11020 Report
-----
Color
Checkbox

Fonts Included in FXR - Not Found in Fap(s)
-----
11006          Times-Roman 6 PT
11008          Times-Roman 8 PT
11012          Times-Roman 12 PT
11014          Times-Roman 14 PT
< and so on... >
```

2107
RPS

IMPROVED SUPPORT FOR POSTSCRIPT SYMBOL FONTS

The system now lets you select a code page for 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 (For both Documaker Studio and the older tools), 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*.

2108
RPS

GENERATING DDL FILES FROM DFD FILES

You can use the new DFD2DDL utility to generate DDL (Data Definition Language) files from your DFD files. DDL files consist of the SQL statements that create your tables and views.

This is helpful if you have an RDBMS database where an administrator is required to create tables and indexes and users are not typically granted those privileges.

Program names

Windows	DFD2DDLW.EXE
---------	--------------

Syntax

DFD2DDLW /I /O /D /T /S /P /L

Parameter	Description
/I	Enter the name of the DFD file. Include the full or relative path.
/O	Enter the name you want the utility to assign to the DDL file it will create. The default is the DDLFile root plus the extension <i>.SQL</i> .
/D	Enter the name of the target database. You can choose from: MSSQL, DB2, Oracle, MySQL. The default is MSSQL.
/T	Enter the name of the target table. The default is the DFD root name.
/S	(Optional) Enter the name of the target schema. This is typically used for Oracle schema, DB2 schema, and MSSQL dbowner.
/P	(Optional) Enter the name of the primary table index. This allows for unique constraints on key fields.
/L	(Optional) Enter the name of the target location. This is used for DB2 (location) and MySQL (database name).

You should avoid the following ExtTypes because they are not supported as external data types in Documaker's database handlers:

- CHAR
- UCHAR
- DECIMAL
- DOUBLE
- LONG_DOUBLE
- TIMESTAMP
- DATETIME

2109
RPS

ENHANCED SUPPORT FOR COLOR LOGOS

The PXL (PCL 6) driver now supports 8-bit color (256 color) logos. Previously, the PXL (PCL 6) driver only supported monochrome and 24-bit color logos.

Be sure to set the SendColor option to Yes, as shown below, and also select the Print in Color option on the Logo Properties/Color Selection menu.

```
< PrtType: PXL >
  SendColor = Yes
```

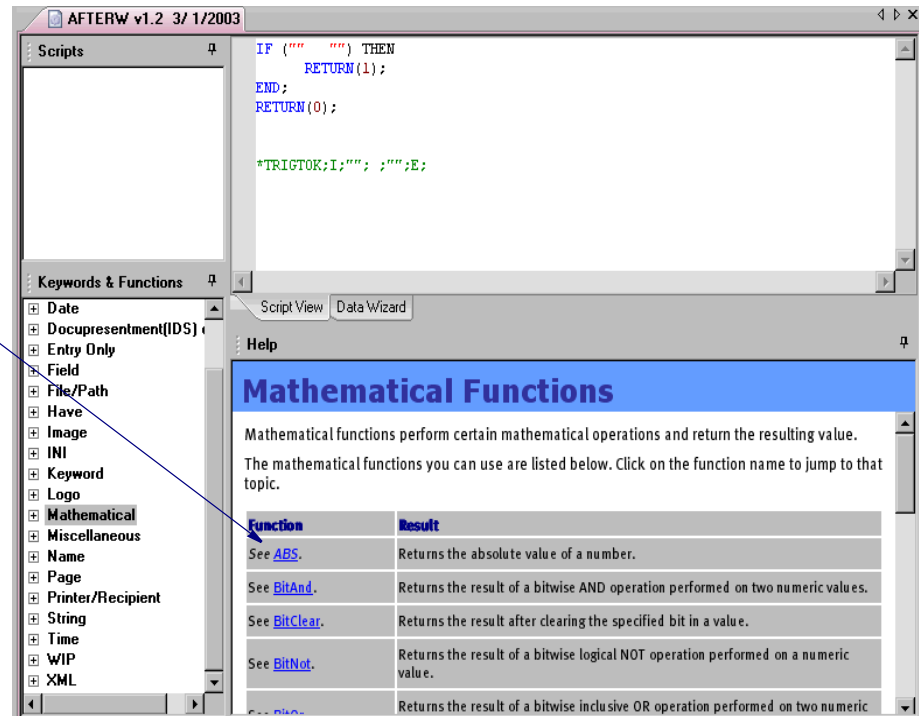
2111
RPS

ENHANCED HELP SYSTEM

The Help area generally located at the bottom of the window provides information about the choices you have or the manager you are working with. In version 11.2, Studio's Help system has been enhanced to use an Internet browser-type window that can have active links to other topics and even to the Internet.

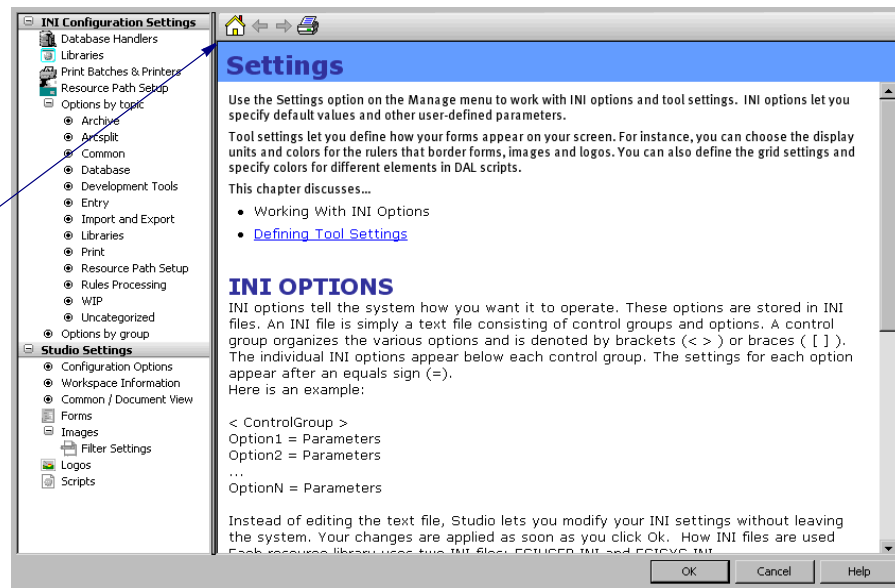
This windows shows Help information for the mathematical DAL functions.

You can click on these links to go directly to information about these functions.



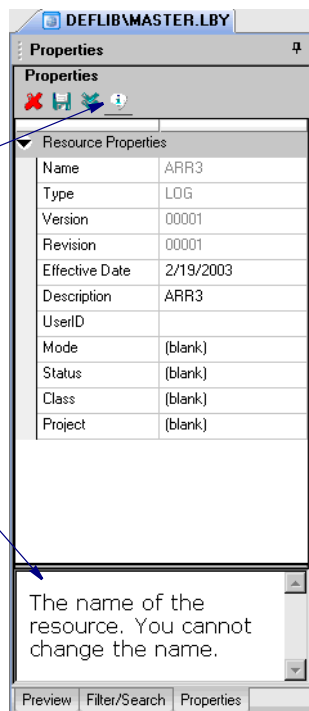
Where applicable, you can move forwards and back through information or press the Home button to return to the starting point. You can also print the topic.

Use these buttons to return to the starting point, move forwards and back through the information, or print the topic.



On the manager windows, such as this excerpt from Library Manager which shows logo properties, you can choose to display or hide the Help text in the Information pane.

Click here to display or hide the Help text shown here for the Name property.



NOTE: Studio remembers how you size the Information pane for most windows. So, if you make the Information pane large enough to cover all the options shown on the window, when you return to that window, you will only see the Information pane and you will have to resize the Information pane to see the properties.

2112
RPS

MAPPING ALTERNATE WIP INDEX COLUMNS FROM IMPORTED WIP DATA

You can add a new set of INI options to the INI control group associated with the import method you are using to identify globally-defined fields from which you want data transferred to the WIP index record.

Here is an example of how you would define one of these INI options:

```
WIPField = FAPField;DFDField
```

Option	Description
WIPField	FAPField is the name of a field stored in the global form set dictionary. The FAPField definition must name a field that is defined as Formset Global (globally) in the import file DFDField is the name of a WIP DFD field.

By default, the system looks for the Key1, Key2, and KeyID fields using their standard WIP names. After that, any additional WIPField definitions are located and mapped accordingly.

Here is an example:

```
< ImpFile_CD>  
WIPField = INSURED NAME;DESC
```

This definition looks for the globally-defined document field INSURED NAME and maps it into the WIP record DESC column. Since the DESC column is normally one of the WIP columns that would be automatically mapped in the V2 header, this definition overrides that and stores the value of the global field INSURED NAME in its place.

2113
RPS

SUPPRESSING DUPLICATE FORM DESCRIPTIONS

Now you can suppress duplicate form descriptions. For instance, suppose you have several versions of what is basically the same form, but each instance of the form has a different name. For your purposes they are the same form and have the same description, even if the names are not identical. And when building the form table using the Form Description Lines feature, you would like to suppress the duplicate form descriptions. You can use the new ExcludeDuplicateDescriptions option to do this:

```
< FormDescTable >  
ExcludeDuplicateDescriptions = No
```

Option	Description
ExcludeDuplicateDescriptions	Enter Yes if you want the system to look for and exclude duplicate form descriptions. The default is No.

NOTE: By default, the Form Description Line feature will eliminate duplicate forms.

2116
RPS

SHOWING FIELD INFORMATION IN PDF FILES

For PDF files, you can now have the system show the names of variable fields and display information about the field.

For example, you create or open a FAP file in either Image Editor or Documaker Studio's Image Manager. When you print the FAP file to a PDF file, you select the Template Fields option on the Print window. When you open the PDF file created from the FAP file, the field names appear instead of a series of Xs. Additional field information appears as you move the cursor over the field when viewing the PDF file.

The field names appear in the fields.

If you hold the cursor over the field, additional information about that field appears.

INSURED LNAME

Field Name: INSURED LNAME

Field Type: Alphanumeric

Length: 20

Scope: Global

No User Edit

Hidden

Rotation: 0 deg.: Normal

Font: 11012 (Times-Roman 12 PT)

Locale:

The additional information shown includes:

- Field name
- Field type
- Field length
- Whether a user can make entries into the field
- Scope (form, global, or image)
- Rotation

- Font ID
- Locale

To tell the system to display the field name and other information when you display the PDF output, set the TemplateFields option similar to how it is shown here:

```
< PrtType:PDF >  
TemplateFields = Yes, Enabled
```

Parameter	Description
TemplateFields	<p>For this option, you enter two parameters separated by a comma. Enter Yes as the first parameter to tell the system to print Xs in variable fields or, in for PDF files, display the field name and information about the field. The default is No.</p> <p>For the second parameter, you have these choices:</p> <p>Enabled - Option appears in the Print window and is active (available to be checked).</p> <p>Disabled - Option appears in the Print window but is grayed out (not available to be checked).</p> <p>Hidden - Option does not appear in the Print window.</p>

NOTE: With all other print drivers, if you set the TemplateFields option to Yes, the system prints a series of Xs in the field.

2119
RPS

USING MIXED MODE AFP DATASETS

This feature enhances the AFP2MVS utility on MVS to support Mixed Mode AFP datasets. Unlike regular AFP files, Mixed Mode AFP files typically contain records of line data — not AFP records.

Prior to this enhancement, the AFP2MVS utility ignored line data records and would not write them to the output dataset. Now, when the dataset is specified via JCL as a Mixed Mode AFP file, all records are written out.

Include the new /L parameter to tell the AFP2MVS utility that the dataset is a Mixed Mode dataset.

2121
RPS

MAKING BANNER FORMS COMPATIBLE WITH MULTI-FILE PRINT

Use the new RetainTransBeginForm INI option to make pre-transaction transaction banner form processing compatible with multi-file printing. Banner forms print separately from the rest of the document. When using multi-file printing with print drivers such as PDF or RTF, banner forms do not appear in the output file. This enhancement provides a way to let the banner form appear in the same print file.

NOTE: This option only affects the banner processing capability of Documaker Server. It does not affect Documaker Workstation.

Banner pages are, by design, not considered part of the form set. A pre-transaction banner page is designed to print separately, using data from the form set, but as if it were not physically part of the form set. For that reason, when *printing* to a single-file-per-transaction format such as PDF, RTF, XML, or HTML, and using the MultiFile print callback method to produce separate files, the banner output is not included in the output file.

It is possible to use pre-transaction banner forms as a way of producing a mailer sheet for a form set. This works for true printed output, but if you are producing a PDF file, for example, the banner (mailer page) does not appear within the PDF.

If, however, you use the RetainTransBeginForm option to retain the pre-transaction banner form, the banner process proceeds as before, but the printing of the banner is initially suppressed. The banner page is retained and remains inside the form set, as the first form in the form set. When the form set is processed by the PDF driver to produce the PDF file, the pre-transaction banner form (or mailer sheet) is then included in the resulting PDF file.

Keep in mind however that the document is only temporarily modified during the print step. The banner form is not included with the actual, intelligent form set when it is archived. For instance, if the intelligent document format is used for archiving, the mailer sheet does not appear as part of the form set, and will not print if retrieved from archive. If, however, you archive the PDF output, then the mailer sheet will appear in the PDF file.

You can place the RetainTransBeginForm option in the Printer control group as a global setting or you can place it at the recipient batch level. A setting at the recipient batch level overrides a setting in the Printer control group.

Here is an example of how you could set a global or default setting in the Printer control group and override that setting for a particular recipient batch:

```
< Printer >
  RetainTransBeginForm = Yes
  ... (other applicable options omitted - see the following note)

< Print_Batches >
  Batch1 = BATCH1.BCH
  Batch2 = BATCH2.BCH
< Batch1 >
  RetainTransBeginForm = No
  ... (other applicable options omitted - see the following note)
```

Option	Description
RetainTransBeginForm	Enter Yes if you want the system to include the transaction banner form in the form set. The default is No. If you are using the PDF, RTF, XML, or HTML print driver, this means the banner pages will be included in each transaction's print file.

NOTE: There are additional INI settings required for single- and multi-step processing. For more information about single-step processing, see the discussion of the PrintFormset rule in the [Rules Reference](#).

For more information about multi-step processing, see the discussion of the MultiFilePrint callback function in the [Using the PDF Print Driver](#).

2122
RPS

SUPPORT FOR QUARKXPRESS VERSION 7.0

The Documaker XTension, which lets you create Documaker resources from QuarkXPress, now works with QuarkXPress version 7.0. In addition, you can now export boxes as 300 dpi bitmaps. Previously, you could only export boxes as 96 dpi bitmaps.

2123
RPS

DEFINING THE CONTENT IN THE MULTIFILEPRINT LOG

Use the new MultiFileLogRecord option to control the content of the log file produced during multi-file printing.

For certain print drivers (PDF, RTF, XML, or HTML), you must generate a separate print file for every transaction in a batch.

For this processing mode	You set the
Multi-step processing (GenTrn, GenData, and GenPrint)	CallbackFunc option in the Print control group to MultiFilePrint
Single-step processing (GenData)	MultiFilePrint option in the PrintFormset control group to Yes

During this process, the system creates a log file to keep track of the print files it creates. The new MultiFileLogRecord option lets you control the contents of the log file produced.

For multi-step processing using the multi-file callback function, you must change the FSISYS.INI file as shown below:

```
< Print >
  CallbackFunc      = MultiFilePrint
  MultiFileLog      = {log file name and path}
  MultiFileLogRecord = ~DALRUN MyScript.DAL
  ... (other applicable options omitted - see the following note)
```

The system first looks for MultiFileLog option in the logical printer control group first, such as Printer1, Printer2, Printer3, and so on. If not found, it then looks for this option in the Print control group.

To control the information written to the MultiFileLog file, specify the name of the DAL script, such as MyScript.DAL, in the MultiFileLogRecord option. The system will then execute this script whenever a new output file needs to be created. If a string is returned, the string is used instead of building the log record as a set of semicolon delimited fields. If an empty string is returned, the current log record format is produced.

NOTE: A linefeed is appended to the string before it is written to the log file.

The DAL script could be as simple as one that returns the string from the DAL function, DeviceName. Here is an example:

```
RETURN( DeviceName() )
```

NOTE: For more information about multi-step processing, see the discussion of the MultiFilePrint callback function in [Using the PDF Print Driver](#).

In single-step processing (GenData), use the MultiFilePrint option in the PrintFormset control group, as shown here:

```
< PrintFormset >
  MultiFilePrint      = Yes
  LogFileType         =
  LogFile             = {log file name and path}
  MultiFileLogRecord = ~DALRUN MyScript.DAL
  ... (other applicable options omitted - see the following note)
```

The PrintFormset rule checks for the MultiFileLogRecord option and if a string is returned, it uses the string instead of building the log record as a set of semicolon delimited fields. If an empty string is returned, the current log record format is produced.

If you set the LogFileType option to XML, the system generates a log file using XML and ignores the MultiFileLogRecord option.

NOTE: There are additional INI settings required for single- and multi-step processing. For more information about single-step processing, see the discussion of the PrintFormset rule in the [Rules Reference](#).

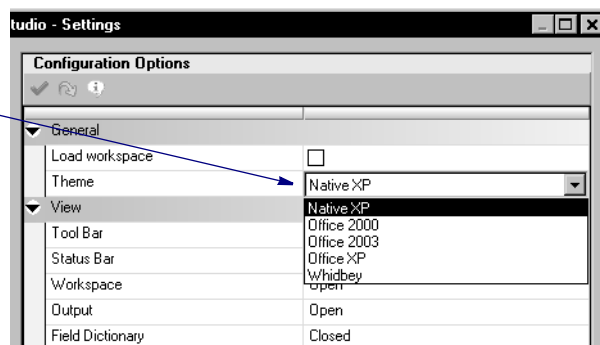
NEW LOOK AND FEEL FOR DOCUMAKER STUDIO

Documaker Studio has been updated to offer a new, more contemporary, look that is comparable to many Microsoft Windows programs. This includes the use of color themes, based on your Windows settings, and new docking/auto-hide windows for many views that typically do not have to remain open all the time.

NOTE: See feature 1938 for more information on docking and auto-hide windows.

To choose a color theme, choose Manage, Settings and then click on Configuration Options. Use the Theme field to select the theme you want to use in Studio.

Choose from these themes.



You can choose the theme you like best and change it whenever you like.

In addition, open documents are now shown using a *tabbed-view* that lets you quickly switch between open windows without having to go through the Window menu. Studio now remembers the layout you chose and will automatically restore the views and windows to the same look when you return.

There are four new *global* windows that are not tied to any specific manager. These windows are similar to the workspace window and can be used to interact with different managers. The new windows are...

- Extract Dictionary
- Field Dictionary
- Navigation
- DAL Scripts

You can use these like any other dockable window. They can be hidden, moved, or closed to suit your needs. If the window is hidden or closed, it can be re-activated by selecting View from the menu, then selecting the menu choice corresponding to the appropriate window.

2128
RPS

USING THE NEW CHECKZEROFONTID RULE

Use this new form set level (level 2) rule to see if the form set has any fields with a zero font ID. The rule will produce an error or warning for any fields it finds which have font IDs equal to zero. You can then correct the font ID problems and restart the processing cycle.

Syntax `::CheckZeroFontID;;Message;`

Parameter	Description
Message	Enter E if you want to see error messages. Leave this parameter blank if you want to see warning messages.

Example This example produces error messages:

```
;CheckZeroFontID; ;E;
```

This example produces warning messages:

```
;CheckZeroFontID; ; ;
```

The error or warning message includes information about the form, image, and field:

Message	Description
DM30059	⌞Error⌟ in CheckZeroFontID: zero font ID: form ⌞PXWORKSHT⌟ image ⌞PXPOLICY⌟ field ⌞TESTFONTID⌟
DM30059	⌞Warning⌟ in CheckZeroFontID: zero font ID: form ⌞PXWORKSHT⌟ image ⌞PXPOLICY⌟ field ⌞TESTFONTID⌟

2130
DAP

ADDITIONAL DOCUMANAGE DATA TYPE SUPPORT IN DOCUMAKER SERVER ARCHIVE

When you are using Documanager Bridge for retrieval and presentation purposes, you can now store data representing date/time values in a native format. This provides greater flexibility for searching and querying using ranges and calendar functionality in web page design.

You can also use data representing numerical values in range searches and queries. Data mining and reporting also benefit from better data representation and storage.

The DMIA DBHandler (DMILIB module: [DBHandler:DMIA]) used with the GenArc program and other Documaker Server archive processes now lets you use additional Documanager Data Types in Documanager Folder fields instead of only supporting the varchar or char data types.

Keep in mind...

- The date/time data types must be in either a Documaker D4 string format:

```
YYYYMMDDHHMMSS
```

The hours, minutes, and seconds (HHMMSS) are optional. For example, the D4 format can be sent in as:

```
20070131 (Jan. 31, 2007)
2007013113 (Jan. 31, 2007 1PM)
200701311330 (Jan. 31, 2007 1:30PM)
20070131133055 (Jan. 31, 2007 1:30:55 PM)
```

Or in a Documanager client-supported string format:

```
YYYY-MM-DD HH:MM:SS.msec
```

The hours, minutes, seconds and milliseconds (HH:MM:SS.msec) are optional. For example, the Documanager format can be sent in as:

```
2007-01-31
2007-01-31 13
2007-01-31 13:30
2007-01-31 13:30:55
2007-01-31 13:30:55.800
```

- Documaker's Archive Application Index Data Format Definition file (APPIDX.DFD) fields must remain as CHAR_ARRAY for the INT_TYPE and EXT_TYPE with the appropriate INT_LENGTH and EXT_LENGTH values for representing the data in string format.

2131
RPS

AUTOMATIC CATEGORY OVERRIDES

Now you have more flexibility when categorizing DPA documents from Documaker Server Archive into Documanager. This provides better search and query functions when retrieving via Documanager Bridge. It also provides more flexibility in using Extended Document Properties (XDPs), which allows for different XDPs in the different document categories so transactions can store different relative data in the XDPs.

You can now use input data to set the Documanager document's Category property during archival via the Documaker Server Archive interface (DMIA). The default value for this property still comes from the FileType INI option during archival, but now you can dynamically override the default with input data using this INI option:

```
< POField2Document >
  ObjectClass = AppIdx_Field
```

During retrieval, the Category Document property can be loaded into the Documaker AppIdx_Field using this INI option:

```
< PODocument2Field >
  AppIdx_Field = ObjectClass
```

Extended Document Properties (XDPs) are based on the Category value set during ingestion. Mappings to XDPs only occur if the XDP for the Document Category exists by name. Otherwise, they are ignored and no error is generated. This allows different data to be populated into the XDPs based on the category used.

Here is an example of how you would override the default document category of DPA with the APPIDX.DFD field value of the field FormSet:

```
< DMIA:RPEX2ARC >
;   FileType is the default Category/ObjectClass value
  FileType = DAP
< PODocument2Field >
;   Category/ObjectClass is overridden by the value in the AppIdx
;   field FormSet
  FormSet = ObjectClass
< POField2Document >
;   Category/ObjectClass is overridden by the value in the AppIdx
;   field FormSet
  ObjectClass = FormSet
```

Keep in mind the APPIDX.DFD field used to override the document Category in the INI options POFld2Document and PODocument2Field can not be used to also set other folder or document properties. For instance, in the example another entry for FormSet can not be used to map FormSet to another folder or document or XDP field.

2140
RPS

TRUNCATING FIELDS

Now you can truncate fields to their FAP defined field length when using the ImportXMLFile rule. To do so, include the new TF (Truncate Field) option.

Option	Description
TF	Enter TF to truncate fields to the field length defined by the FAP file. Make sure you specify this parameter before FILE option.

This example will truncate the fields lengths:

```
;ImportXMLFile;;TF,SCH=1,XML_FILE_NAME 15,55;
```

These examples truncate fields and suppress pagination:

```
;ImportXMLFile;;SP,TF,SCH=1,XML_FILE_NAME 15,55;
```

```
;ImportXMLFile;;TF,SP,SCH=1,XML_FILE_NAME 15,55;
```

This example does not truncate fields or suppress pagination:

```
;ImportXMLFile;;SP
```

NOTE: No formatting is allowed on the multi-line text field when you include the TF option.

2141
RPS

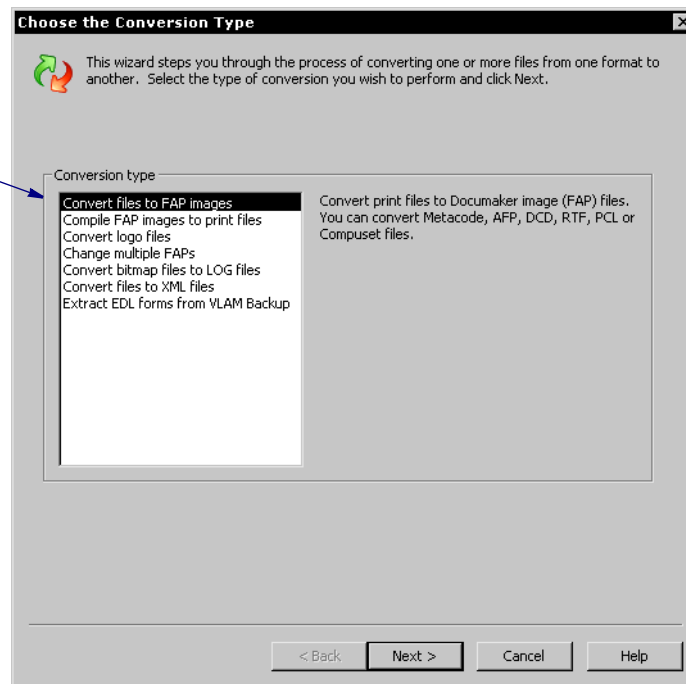
CREATING FOR FILES WHEN CONVERTING FAP FILES

When converting files into FAP files in Studio, you can now tell the Conversion wizard to create a form file (FOR) for each FAP image it creates. For each image created, a form (FOR) file is created that contains a reference to the corresponding image.

NOTE: The FOR file will have the same name as the FAP image, but with the FOR extension.

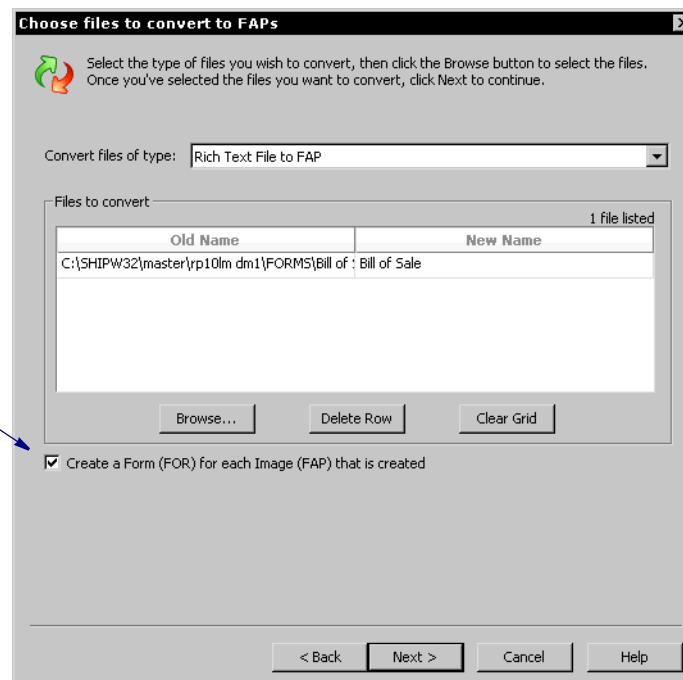
To create FOR files, open Conversion manager. The Choose the Conversion Type window appears.

Choose the option to convert files into FAP images.



Choose the option to convert files into FAP images and click Next. The Choose Files to Convert to FAPs window appears.

Click here to tell Studio to create a FOR file for each FAP image it creates.



Select the conversion type and then the files you want to convert. Then check the Create a Form (FOR) for each Image (FAP) that is Created box and click Next. The Convert Files and Finish window appears.

On the Convert Files and Finish window, you can choose to check the files into the library or save the files to a location on disk. If you choose to save the files to disk, Studio prompts you to specify the location.

2142
RPS

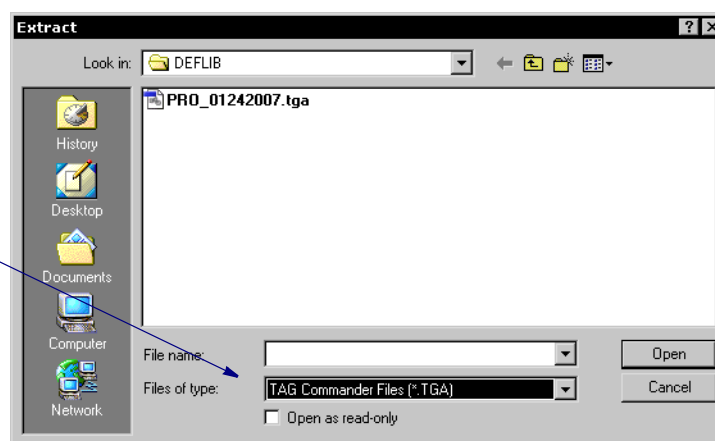
IMPORTING TGA FILES INTO YOUR EXTRACT DICTIONARY

Now you can import tag definition (TGA) files into your extract dictionary (XDD). TGA files are created by Tagcommander and used by Documerge. The fields in a TGA file contain name and length information.

To import the TGA fields into your extract dictionary, first open Database manager and check out the appropriate XDD file.

Next, choose the File, Import option. The File Open window appears. Click on Extract in the left column if it is not already selected to select the dictionary type. Then click Browse.

Select the TAG Commander Files option in the Files of Type field.



Select TAG Commander Files (*.TGA) in the Files of Type field. Then choose the file you want to import and click Ok. During the import, the fields in the TGA file are imported into your XDD file.

Because TAG definitions do not have much in common with Documaker rule definitions, the import only includes the name of the field and the specified length. All of the imported fields will default to a Move_It rule.

You must review each field and make any appropriate changes to map the data from your expected input file.

NOTE: Previous releases let you import TGA files into the field base (FDB).

2143
RPS

PAGINATING FORMS USING DAL FUNCTIONS

Version 11.2 includes a new DAL function/procedure (PaginateForm) and changes to the AddImage and DellImage DAL functions to make it easier to control the pagination of forms.

The PaginateForm function lets you apply image origins and re-paginate the form if necessary. During this re-pagination, the function will create or delete pages as needed.

The AddImage and DellImage DAL functions have been enhanced to include a new parameter you can use to force re-pagination after the affected image has been manipulated.

PaginateForm

Use this function/procedure to apply image origins and re-paginate the form if necessary. During this re-pagination, the function will create or delete pages as needed.

Syntax PaginateForm (Form, Group)

Parameter	Description
Form	(Optional) If you omit this parameter, the current form controlling the active script is paginated. If you include the name of a form, that form is located and paginated. You can include the occurrence indicator (a backslash followed by a number, such as BIZ\3) to indicate a specific occurrence of the form to find and paginate. If you do not specify an occurrence with the name, the first occurrence of the form is paginated.
Group	(Optional) This parameter identifies the Key2 or GroupName2-level parent that contains the form. This is sometimes referred to as the <i>line of business</i> that contains the form. If you omit the Group parameter, the system tries to locate the named form within the current group that is controlling the execution of the script.

You can call PaginateForm as a function or procedure. As a function, it returns a one (1) if the requested form is located or a zero (0) if it could not be located.

Note that if the form is found and paginated, there may not be any visible change to the document. The form layout is determined when you design the form and by the application of image origin rules.

AddImage

The AddImage function now includes an optional parameter that indicates whether form pagination should occur after the image is added. Form pagination includes the application of image origin rules to determine whether new pages are required for the pre-defined page sizes.

AddImage (FAP, Image, Form, Group, Flag, **Paginate**)

Parameter	Description
Paginate	<p>(Optional) This parameter follows the Flag parameter. If you enter anything other than a zero (o), it tells the system you do want form pagination to occur upon successful inclusion of the new image.</p> <p>If the image does contain an origin rule and you omit the Paginate option or set it to zero (o), the image origin rule executes upon insertion.</p> <p>Whether the inserted image has an origin rule or not, the positioning of this image when the Paginate option is omitted or zero (o) does not cause the entire form to be re-paginated. This means if the placement of the image causes it to overlap another image or to be out of the page boundary, no additional re-pagination occurs. If you are manipulating multiple images in series, you may want to conclude your script with a call to <code>PaginateForm</code> to make sure the entire form is re-paginated.</p> <p>Here is an example:</p> <pre>AddlImage("myFAP", "mainImage" , , , 1,1)</pre> <p>This example omits the Form and Group parameters, but does specify the Flag parameter as well as the Pagination parameter.</p>

NOTE: If you enter zero (o) or omit this parameter, the function works as it did in prior versions.

DelImage

The `DelImage` function now includes an optional parameter that indicates whether form pagination should occur after the image is deleted.

```
DelImage( Image, Form, Group, Paginate)
```

Parameter	Description
Paginate	<p>(Optional) This parameter follows the Group parameter. If you enter anything other than a zero (o), it tells the system that you want form pagination to occur upon the successful removal of the image.</p> <p>If you omit this parameter or enter zero(o), the image is deleted, but no other images are moved to occupy the space left vacant. Subsequent form re-pagination and the application of image origins may change the layout of the form.</p> <p>Here is an example:</p> <pre>DelImage("myImage", , , 1)</pre> <p>This example omits the Form and Group parameters, but does include the Paginate parameter.</p>

NOTE: If you enter zero (o) or omit this parameter, the function works as it did in prior versions.

2144
RPS

CONTROLLING PAGINATION WHEN EDITING A FORM SET

You can use the new AutoPagination option to control pagination when you edit a form set in Documaker Workstation or iDocumaker Workstation.

```
< EntryOptions >  
    AutoPagination = No
```

Option	Description
AutoPagination	Enter No to turn off automatic re-pagination when editing a form set in Documaker Workstation or iDocumaker Workstation. The default is Yes.

NOTE: Keep in mind you can also use the AutoPagination option in the Control control group to prevent the system from re-paginating the form set when an image grows because of expanding text in a text area. Here is an example:

```
< Control >  
    AutoPagination = No
```

This tells the system not to re-paginate the form.

2146
iDMS

AUTOMATICALLY SAVING FORM SETS IN IDOCUMAKER WORKSTATION

You can use the new AFEPluginTimerSave timer service function with iDocumaker Workstation to automatically save the current form set.

NOTE: This only affects iDocumaker Workstation.

To use this timer service function, add an AutoSave option to the WIPEDIT.INI file, as shown here:

```
< TimerFuncs >  
    AutoSave = ;State;Urgency;Seconds/Time-of-day;Function
```

Parameter	Description
State	Always set this parameter to one (1).
Urgency	You should set this parameter to one (1). Zero (0) means the call can be skipped if the user is engaged in an operation. A one (1) tells the system to automatically save the current form set as soon as the user finishes the current operation.
Seconds/ Time-of-day	Enter the number of seconds to wait before automatically saving the current form set. The default is 60 which means the system will automatically save the form set every 60 seconds, as long as the desktop is open and the user is not selecting a window or menu option. You can also enter a specific time of the day at which time you want the system to call this function. Keep in mind the system uses a 24-hour clock, so you would enter 13:00 to indicate 1 pm.
Function	Enter the name of the timer service function exactly as shown here: AFEW32-> AFEPluginTimerSave Keep in mind that case is important when entering the timed service function name.

Here is an example:

```
< TimerFuncs >
AutoSave = ;1;1;300;AFEW32->AFEPluginTimerSave
```

This example tells iDocumaker Workstation to automatically save the form set every five minutes. Here is another example:

```
< TimerFuncs >
AutoSave = ;1;1;17:00;AFEW32->AFEPluginTimerSave
```

This example tells iDocumaker Workstation to automatically save the form set at 5 p.m.

2147
RPS

USING THE NEW FONT USAGE REPORT

You can now print a Font Usage report from Documaker Studio. This report provides information about the fonts images you selected. The information comes from the font cross-reference (FXR) file for those images.

There are three sections in this report:

- The first section shows the information about all fonts which are used in the images you selected.
- The second section shows the information about fonts which are not used in the images you selected.
- The third section shows the fonts used in each of the selected images for each version and revision of the image.

To run the report, follow these steps:

1 Start the Reports wizard, then select Font Usage Reports.

You can start the Reports wizard by clicking Reports on the Project tree or while using Font Manager. From the Font Manager, the Reports wizard is available from the:

- Font menu, choose Fonts, Reports.
- Popup menu, right click and choose Reports.
- Main menu, choose File, Reports.

2 Select the FXR file. This file contains all of the font information. You can choose any FXR file, not just the one selected for your current workspace.

NOTE: Be sure to choose the right FXR file for the images you will select. Documaker images (FAP files) only contain the ID of the font being used and rely upon the FXR file to provide the font information. If you have two FXR files that define the same font IDs, but with differing font characteristics, this report cannot tell you which FXR was used in the image.

3 Select the library (LBY) file. You can choose any library file listed under the Manager, Settings, Libraries option. Click Browse to select a library. The library type can be CB5, ODBC, DB2, and so on.

4 Select images you want included. You can select all images in the library or individual images if the library you chose is the current workspace library.

5 Click Finish to generate the report.

2150
RPS

GENERATING PDF417 BARCODES

Documaker Server now lets you create PDF417 barcodes that can contain any type of information. For instance, this capability makes your Documaker Server system compatible with the New York State Insurance Department's (NYSID) regulation that requires barcodes on driver ID cards.

In prior releases, the PDF417 capability was distributed as an add-on product and was only available via a separate license. This capability and the rules required to use it are now incorporated into Documaker Server and will be available to you when you install version 11.2.

Previously, the PDF417 add-on included the PDF417 fonts and these FXR files:

- * PDF417_2.FXR
- * PDF417.FXR

The new REL112.FXR file (see feature 2179) includes the PDF417 font references found in these two PDF417 FXR files. Prior to version 11.2, you would use the PDF417_2.FXR file when your primary printer was an AFP 240 DPI printer. You would use the PDF417.FXR file in all other cases. The two PDF417 FXR files shared the same two font IDs (0912 and 1216) for the two sizes of PDF417 barcodes.

The 0912 and 1216 font IDs from PDF417.FXR (used for 300 DPI printing) have been included in the new REL112.FXR file.

The 0912 and 1216 font IDs from PDF417_2.FXR (only used if your primary printer is an AFP 240 DPI printer) have been included in the new REL112.FXR file as font IDs 0911 and 1215 to avoid conflicts with the 0912 and 1216 font IDs used for 300 DPI printing.

NOTE: For more information on creating PDF417 barcodes, see the separate manual entitled, *Implementing PDF417 Barcodes*.

2152
RPS

OPENING DOCUMAKER STUDIO FILES

You can now double-click on a supported file type in Windows Explorer to open Studio and open that file. When you run Studio for the first time, it automatically registers its applicable file types with Windows.

You can also enter a command on the command line similar to the one shown here to start Studio and open a file.

```
dmstudio.exe rpex1.dxm
```

This command would open the rpex1 workspace.

These file types are supported:

File type	Description
DXM	Workspace files. Studio starts and loads the workspace you selected.
FAP, FOR, GRP, DAL, BDF	Studio starts and opens the file in the most recently used workspace. You must have at least one workspace in the most recently used file list for these files to open.

If you select multiple files, you will open a session of Studio for each file.

2155
RPS

DEFAULTS FOR THE MODULE AND PRINTFUNC OPTIONS

Default values for the Module and PrintFunc options in the PrtType:xxx control group are now provided when you use a standard print type name or print class, such as AFP, PCL, PDF, PST, VPP, XER, XMP, or GDI.

These defaults keep you from having to enter the Module and PrintFunc names in your INI file. For example, if you want to generate AFP print files, you can specify these INI options:

```
< Printer >
  PrtType      = MYAFP
< PrtType:MYAFP >
  Class        = AFP
```

And the system will default these options for you:

```
< PrtType:MYAFP >  
Module      = AFPPRT  
PrintFunc   = AFPPrint
```

2156
RPS

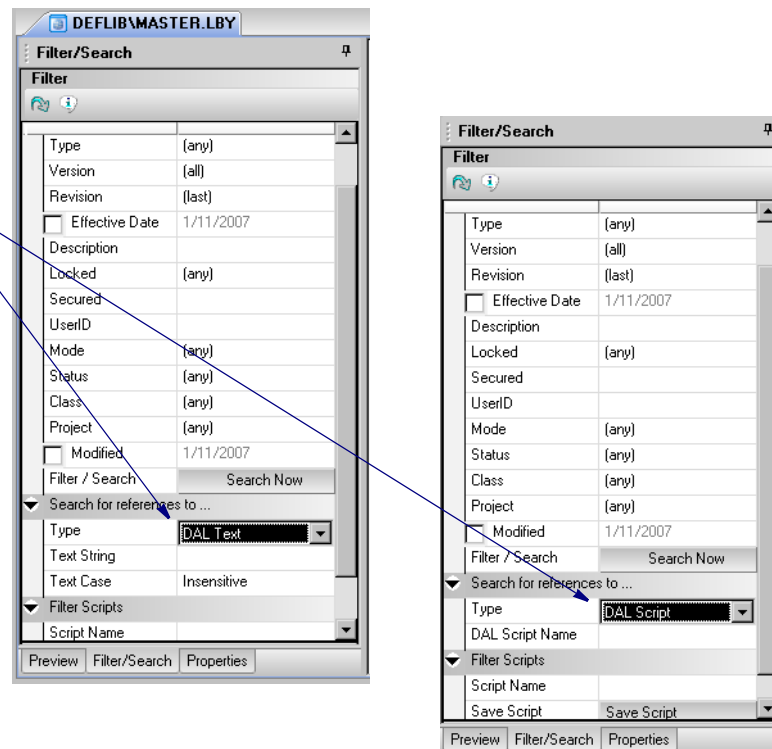
SEARCHING FOR DAL SCRIPTS

When using Library Manager in Documaker Studio, you can now search...

- DAL scripts and DAL subroutine libraries for references to a text string
- DAL subroutine libraries for a specific DAL script
- FAP files for references to a specific DAL script invoked using the DAL CALL statement

Here is an example:

You can select DAL Text to enter a string for which you want to search or you can select DAL Script and enter the name of the script.



To search for a string, enter the string and select whether you want a case sensitive or case insensitive search. To search for a script, enter the name of the script.

NOTE: As with any other type of filter or search, you can save the search criteria for DAL text and DAL scripts in a library filter script (LSC) file and later use this file as input to the LBYPROC utility. The LBYPROC utility processes library scripts in a batch (or command line) mode.

2158
RPS

USING THE NEW STRCOMPARE FUNCTION

Use this function to compare two strings with case a consideration. In normal DAL string expressions, strings are compared in a case-insensitive manner. For example, the system would normally evaluate the following strings to be equal:

```
ABC
abc
```

If, however, you use the STRCompare function, the system will consider case and judge these strings to not be equal.

NOTE: The best way to use this function is to test for equality. For instance, use this function to test two strings and compare for a zero (0) value being returned to indicate the strings are equal or a non-zero value to indicate they are unequal.

You can use this function to determine if one string is greater or less than the other, but the result can be confusing if the strings contain mixed case or have different lengths.

Syntax

```
#RTN = STRCOMPARE ( String1 , String2 , #count )
```

Parameter	Description	Defaults to...
String1	The text for the first string you want to compare	an empty string
String2	The text for the second string you want to compare	an empty string
#Count	Optional. The number of characters to compare. If you enter a value greater than zero, the system compares that number of characters. If you enter zero (0) or less, the system compares all characters. If you enter a value greater than the length of either string, the system pads the strings with blank characters to match the number of characters you specified.	-1 which indicates that all characters will be compared.

If String1 and String2 compare as equal, the system returns a zero (0).

The system returns a negative one (-1) if String1 is less than String2.

The system returns a one (1) if String1 is greater than String2.

Example Assume String1 is *ABCDEF* and String2 is *ABCdef* in these examples:

This example	Returns
#RTN = STRCompare(string1 , string2)	-1
#RTN = STRCompare(string2 , string1)	1
#RTN = STRCompare(string1 , string2 , 3)	0

2159
RPS

LOADING LOGOS FROM THE LIBRARY WHEN CONVERTING AFP OR METACODE FILES

In previous versions, if you tried to load AFP or Metacode files that reference logos which were in the library, you received an error message. The system required the files to be on disk.

With version 11.2, the system now first checks for the logos in the library. If the logos are not in the library, it then checks to see if they are on the disk.

2160
RPS

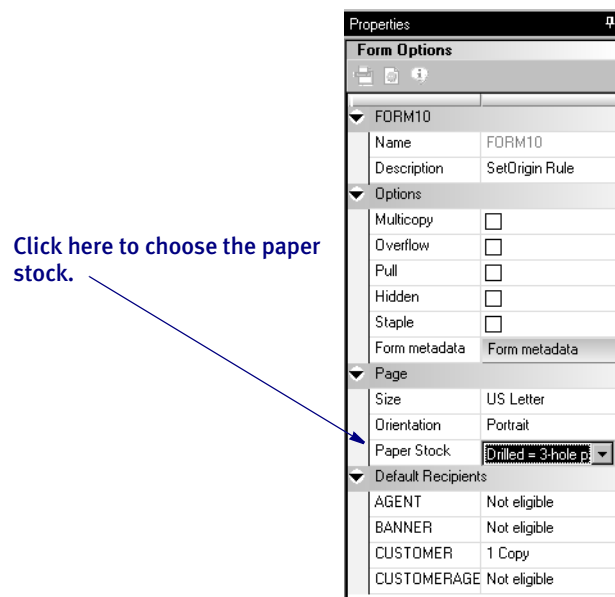
PAPER STOCK SUPPORT IN THE METACODE PRINT DRIVER

In Documaker, you can specify that a form should print from any of nine paper trays. In version 11.2 of Documaker Studio, for Metacode printers you can additionally specify what paper stock the form should print on. This will help users who have more than nine types of paper stocks.

Here is an example of the INI options you could set up:

```
< PaperStockIDs >
  PaperStockID   = Drilled
  PaperStockID   = 201b
  ... (and so on)
< PaperStockID:Drilled >
  Description     = 3-hole paper
< PaperStockID:201bW >
  Description     = 201b White Paper
  DJDE            = DJDE name
```

Once you have set up the appropriate PaperStockID INI options, you will see those options available via Studio's Form manager. Just open a form and select the appropriate paper stock in the Paper Stock field on the Properties tab, as shown here:



Your selection is reflected in the POL file produced by the GenData program. In this example, the form called DEC PAGE has a paper stock ID of *Drilled*.

```
; SAMPCO; LB1; DEC
PAGE; | | FORMPAPERSTOCK=Drilled| | ; R; ; QPRUNA | DL(3360,18600) <AGENT, COMP
ANY, INSURED>
```

In the Metacode printer control group, you must set the TrayUsePaperStockID option to Yes, as shown here:

```
< PrtType:XER >
  TrayUsePaperStockID = Yes
```

If the TrayUsePaperStockID option is set to Yes, the Metacode print driver will take the form's PaperStockID and try to find the DJDE INI option for it in the INI file when it emits the tray command.

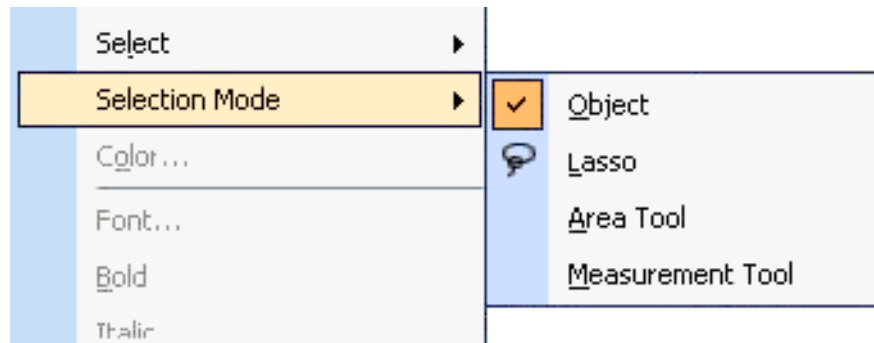
Keep in mind...

- The paper stock selection applies to the entire form
- Only the Metacode print driver uses the paper stock selection
- Only Documaker Studio lets you select the paper stock

2163
RPS

NEW WAYS TO SELECT OBJECTS

In both the Image and Form managers, the Edit Menu now includes a Selection Mode option with these choices:



Option	Description
Object	Normal selection mode
Lasso	Lets you lasso objects.
Area Tool	Lets you mark an area, just like the Tools, Select Area option which only applies to images.
Measurement Tool	Lets you drag an area to get measurements. (See feature 2171 for more information.

In addition, the Lasso option is available from the tool bar by click on this button:



2164
RPS

USING THE NEW COMPLETE AND EXIT OPTION

You can now add an option to your MEN.RES file, which defines the system menus, that lets you complete the current form set and then exit the system. This new option combines the Complete and Exit menu options.

NOTE: If there is an error on the form set, the system will not complete the form set and exit the system until you correct the error.

This option is typically used with iDocumaker Workstation or iPPS implementations but it can also be used if you have created specialized applications that launch entry via the RACLIB interface. Documaker Workstation and PPS users can also use this option if users typically exit the application after completing a form set.

To add the Complete and Exit option to your system, you must first add the following line to the MEN.RES file:

```
MENUITEM "Complete and &Exit" 106 "AFEOS2->AFECmplShutDown"
"Complete the form set and exit the system"
```

You can add this line, or a line similar to it, under any menu group you like. You can also change the text of the option.

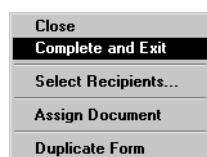
NOTE: In this example, 106 is used as the ID for the menu function. You can assign any ID between 100 and 200, as long as it is not used by another menu function.

The ampersand (&) indicates that the next character is the accelerator for this menu option. You can omit the ampersand if you like.

For instance, you could place this option on the Formset menu, as shown here:

```
POPUP "Form&set" 170 "Formset"
BEGIN
    MENUITEM "&Close" 1069 "Close document" "NULL"
    MENUITEM "Complete and &Exit" 106 "AFEOS2->AFESComplShutDown"
    "Complete the form set and exit the system"
    SEPARATOR
    MENUITEM "Select &Recipients..." 187 "AFEW32-
>AFESelectRecipients" "View forms by recipient"
    SEPARATOR
    MENUITEM "&Assign Document" 150 "AFEW32->AFESAssignDocument"
    "Assign Document"
    SEPARATOR
    MENUITEM "&Duplicate Form" 104 "AFEW32->AFEDuplicateForm"
    "Duplicate current form"
END
```

To see this menu:



2165
RPS

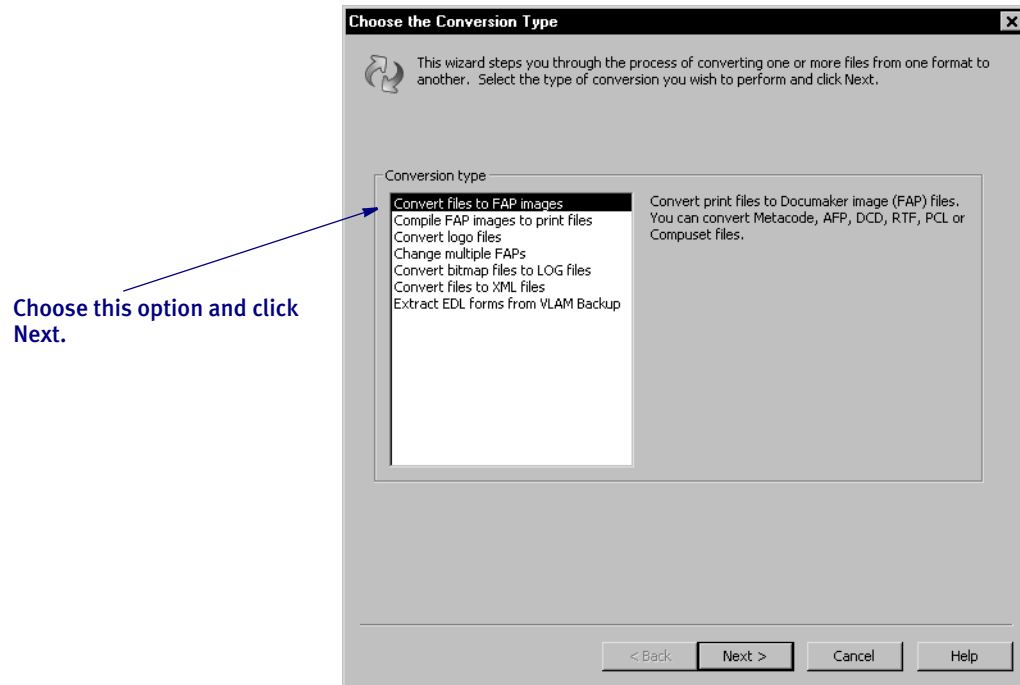
ENHANCED AFP CONVERSIONS

The Conversion wizard in Documaker Studio contains several enhancements to make the conversion of normalized and non-normalized AFP print streams into images (FAP files) easier. With these enhancements the Conversion wizard can now...

- Read the AFP print streams and determine whether there are fonts that are referenced in the print streams which are not already in the font cross-reference (FXR) file. If so, the wizard lets you import these fonts into the FXR file before the conversion starts.
- Determine whether the print streams contain references to page segments which have not already been added to the workspace's library as logo (LOG) files. If so, the wizard lets you import these LOG files into the library before the conversion starts.
- Convert AFP files that contain fonts referenced either by a Coded Font (Xo*****) or by a Character Set (Co*****) and Code page (T1*****) combination.

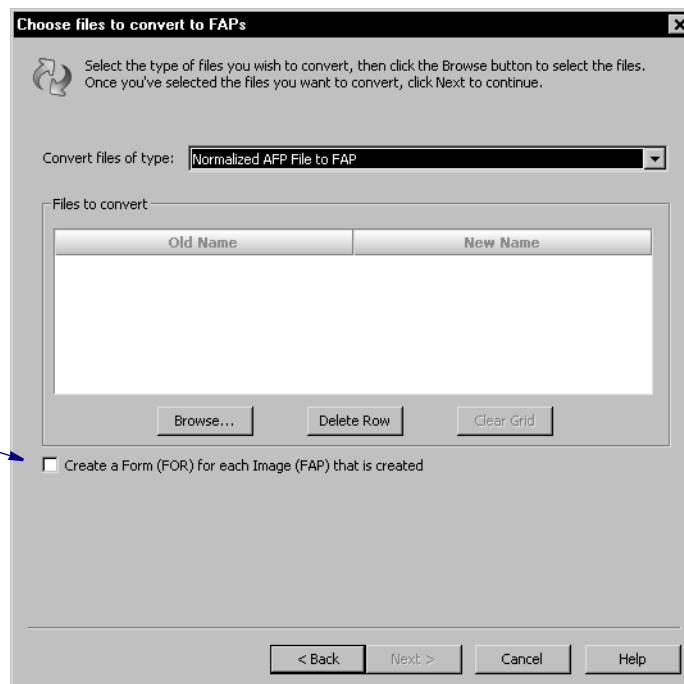
To implement these enhancements, two pages have been added to the Conversion wizard.

To perform a normalized AFP to FAP conversion, start the Conversion manager and select Convert files to FAP images, as shown here:



Click Next. The Choose Files to Convert to FAPs page appears.

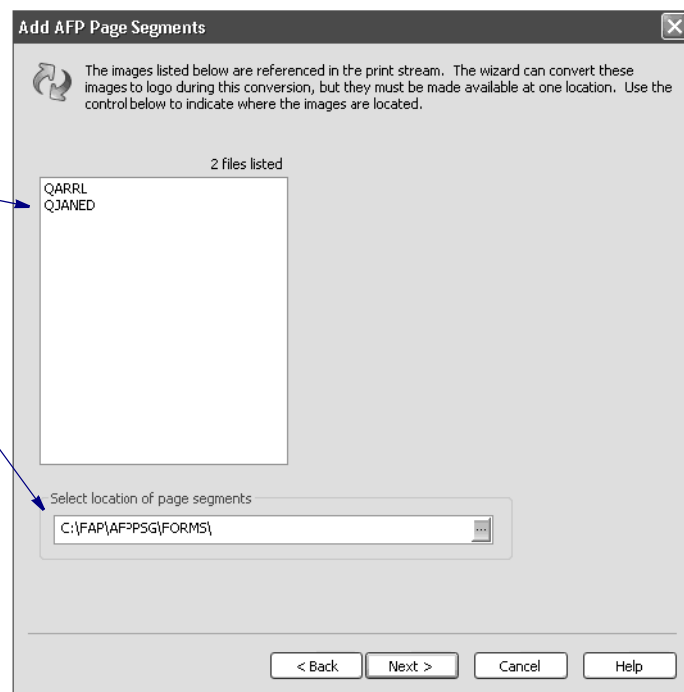
Click here if you also want to create a FOR file for each FAP file.



Choose the option to convert normalized AFP files into FAP files and then click Browse to select the normalized AFP files you want to convert. Then click Next to continue.

If there are page segments in the AFP files, but there are no corresponding LOG files in the library, the Add AFP Page Segments window appears.

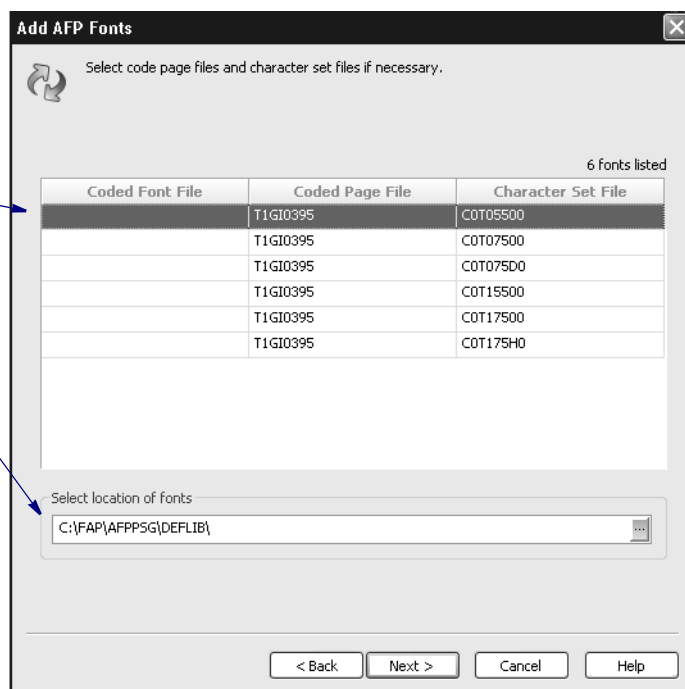
Use the Browse button to find the location of the files listed here.



Specify the location of the missing page segment (SEG or PSG) files and click Next.

If there are fonts in the AFP print stream which are not in the FXR file, the Add AFP Fonts window appears.

Use the Browse button to find the location of the fonts listed here.



Specify the location of the missing fonts and click Next. The Finish window appears. Click Finish to start the conversion.

NOTE: If you selected to use an EDL effective date and description to check into the library, the effective date and description on this window are disabled. You can also select to save the converted FAP files to the disk.

2166
RPS

COPYING THE EDL FORM DESCRIPTION INTO THE LIBRARY DESCRIPTION FIELD

When using the Conversion manager in Studio to extract Documerge EDL forms from a VLAM backup, the Description field for the EDL form is extracted from the VLAM backup and is displayed on the Extraction Information page of the Conversion wizard. From this page you can choose to either extract the EDL forms to disk or continue on to the Metacode or AFP to FAP Conversion wizard.

The EDL form description from this page, however, was not copied to the Library Description field when the image (FAP) and, optionally, form (FOR) was checked into the library following the conversion from Metacode or AFP to FAP and FOR.

This enhancement changes the system so the Documerge description for each Documerge EDL form will be used as the description for the FAP (and optionally FOR) resource when that resource is checked into the library by the Conversion wizard.

Here is an example of the Extraction Information window:

Extraction Information

Use the control below to select the type of forms you would like to extract from the VLAM backup file. Once you select a type, the grid will be updated to display the forms of that type.

Extraction Type: AFP

29 forms listed

Name	Description	Effective Date
LM.LIFE.BANNER	INSTALLATION TEST BANNER	1/1/1900
I.M.LIFE.BANNER	INSTALL TEST 14	1/1/1900
I.M.LIFE.BANNER	INSTALL TEST 380	1/1/1900
I.M.LIFE.CONTRACT	INSTALLATION TEST CONTRACT	1/1/1900
I.M.LIFE.CONTRACT	INSTALL TEST 140	1/1/1900
I.M.LIFE.CONTRACT	INSTALL TEST 38	1/1/1900
I.M.LIFE.CONTRACT	INSTALL TEST 97	1/1/1900
I.M.LIFE.DEC	INSTALLATION TEST DEC	1/1/1900
I.M.LIFE.DEC	INSTALL TEST 1403	1/1/1900

☒ Proceed to print stream to Image (FAP) conversion

☐ Use the Description above when checking resources into the library

☐ Use the Effective Date above when checking resources into the library

Extracted file location: C:\fap\AfpPsg\AFP

< Back Next > Cancel Help

If you want to get a FAP image, check Proceed to print stream to Image (FAP) Conversion box. After you check this box, you can decide if you want to use the description and effective dates shown on the grid when the converted FAP files are checked into library by selecting or deselecting the next two boxes.

You can also edit the description or effective date by simply clicking in the appropriate cell in the grid.

Click Next to continue.

2167
RPS

IMPROVED WIP SELECTION PERFORMANCE DURING BATCH PROCESSING

The system now limits the result sets queried from DBMS systems by the MergeWIP rule to the rule's parameters for the STATUSCODEs to significantly improve query speeds when this rule is run against large WIP index tables.

You do not have to make any changes in the way you use the MergeWIP rule to get these performance enhancements. There are no new parameters or options for the MergeWIP rule.

When the WIP index table is in a DBMS, the appropriate WHERE clause for the STATUSCODE is added to the query automatically. A separate query for each STATUSCODE provided to the MergeWIP rule is used to retrieve the WIP list.

To improve performance on very large WIP tables, you must add an index on the DBMS WIP index table on the STATUSCODE column to avoid full table scans. An index on the WIP index table on the CURRUSER and STATUSCODE columns can also improve the performance of other queries to the WIP index table when the query to build a WIP list for a specific user is performed.

2168
RPS

EXPANDED SEARCH CAPABILITY

When using the library Filter/Search control in Studio, you can now specify a search mask consisting of an asterisk, a string of characters, followed by another asterisk to search for resources that contain that string of characters anywhere in their name.

For example, you could enter a search mask of **FL** to search for library resources that have the letters *FL* anywhere in their name.

This provides more flexibility when specifying a library resource name search mask in the Filter/Search control. Previously, you could only specify a string of characters followed by an asterisk to search for the resources that began with those characters.

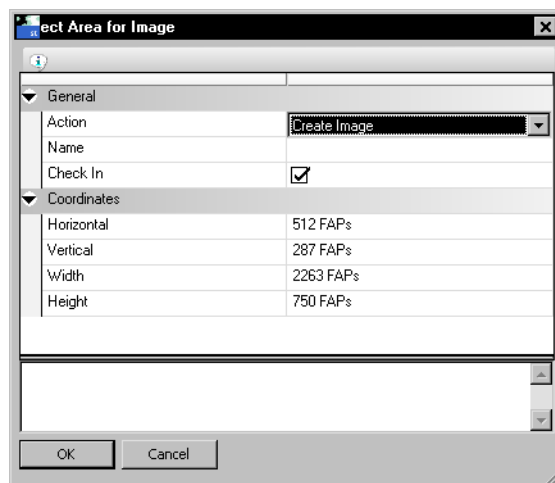
2170
RPS

CREATING AN IMAGE FROM AN AREA

You can use the new Select Area for Image option on the Tools menu of the Form and Image managers to draw an area into which you then create or insert an image. If existing objects fall within the area you draw, those objects are incorporated into the new image.

NOTE: For an object to be incorporated into the new image, the entire object must be inside the area you drew. If any part of the object is outside the area, it will not be included in the new image.

You can right click and choose the Select Area for Image option or choose it from the Tools menu. Then click and drag the cursor to create an area that outlines where you want the image. The Select Area for Image window appears.



The Action field provides these options:

Option	Description
Create Image	(Form and Image manager) This option lets you create an image that you assign a name in the Name field.
Use Existing Image	(Form manager only) Select this option to name or look up an image in the library to insert.
Clear Area	(Image manager only) Select this option to clear the area when you click Ok.

In the Name field, enter the name of the object or use the Browse button to select an object from your library.

If you are creating an image, you can click in the Check In field to check the image into your library. This field is not available if you selected an existing image.

You can modify the coordinates if necessary and see the highlighted area change. When you click Ok, the highlighted area is used for the new image. The white space is maintained.

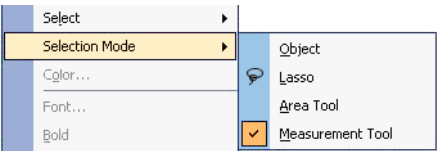
2171
RPS

USING THE MEASUREMENT TOOL

You can use the new measurement tool to, for instance, measure the distance between lines of text or other objects to make sure the spacing meets a certain specification.

For example, you can turn on this tool, position the cursor at one point in the image and then, while holding down the mouse button, draw a box around the area you want to measure. Studio then shows you the measurements and coordinates for the area you drew.

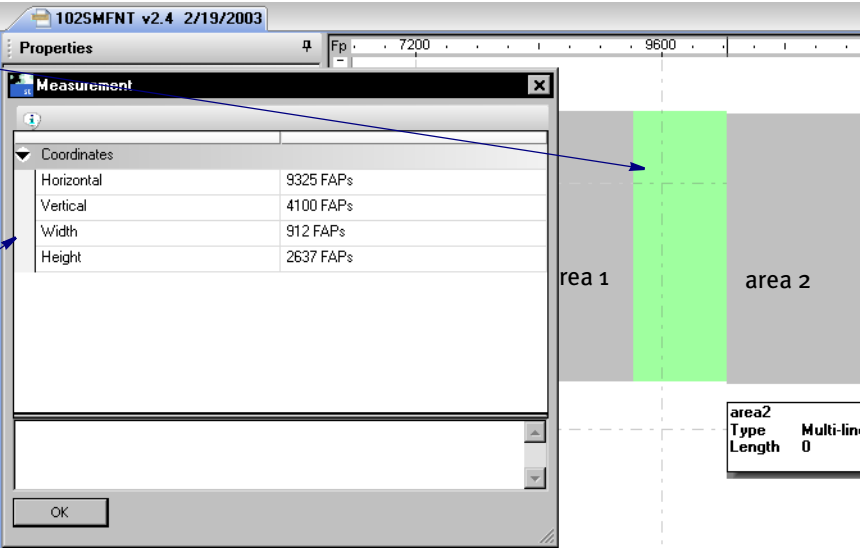
To turn on this tool, open an image and select the Edit, Selection Mode, Measurement Tool option, as shown here.



While in this mode, you draw a box that covers the area you want to measure. Studio highlights the area and displays the Measurement window which give you its measurements and coordinates.

This is the box drawn with the measurement tool.

Here are the coordinates and measurements for that box. You can modify these measurements and the box will change appropriately.



In the example shown above, based on the box drawn by the measurement tool, you know the distance between *area2* and *area1* (partially covered by the Measurement window) is 912 FAP units. Since the box is equal in height to *area1*, you also know that *area1* is 2637 FAP units tall.

The horizontal and vertical coordinates tell you where the box lies on the image. And, if you look closely, you can see that the top edges of *area1* and *area2* are not aligned.

Click Ok to discard the box you drew. You remain in measurement tool mode until you deselect the Edit, Selection Mode, Measurement Tool option.

2179
RPS

USING THE NEW FONT CROSS-REFERENCE (FXR) FILES

Version 11.2 includes these new FXR files:

- REL112.FXR
- REL112SM.FXR

These files differ from the REL103.FXR and REL103SM.FXR files in that...

- The PDF417 fonts were added into the base FXR file. See feature 2150 for more information.

- Character widths were corrected for font records previously created by importing TrueType fonts
- Font heights were corrected for the Times fonts so Windows will select the correct screen font

2183
RPS

USING THE FDT2DB UTILITY'S NEW PURGE PARAMETER

Use the new Purge parameter with the FDT2DB utility to delete the data currently in the database table. Here is an example:

```
FDT2DB /purge /ini=fsiuser.ini /i=form.dat
```

This example tells the utility to purge the contents from the tables before it converts the resources defined by the FSIUSER.INI and FORM.DAT files into those database tables. For more information on the FDT2DB utility, see the Docutoolbox Reference.

2185
RPS

USING THE NEW RUNTRIGGERS RULE

Version 11.2 includes a new triggering rule named RunTriggers. This rule replaces the LoadRcpTbl and RunSetRcpTbl rules in implementations created by Documaker Studio.

NOTE: This rule is only available if you are running Documaker Studio.

The RunTriggers rule represents a new and improved triggering process tailored for the new file structure implemented with Documaker Studio. This new process removes some of the complexity and improves performance. The RunTriggers rule is only available in Documaker Studio.

For instance, in the old triggering process, if your form trigger asked for three copies of a form and then you processed the image triggers, only the last copy of the form was affected by the image triggers. The first two copies, would have been static based upon the default image configuration in the FORM.DAT file. Using the RunTriggers rule, each copy of the form will process image triggers. This means each form can differ from the prior one — assuming your image triggers check unique information to achieve that goal.

If you do not use Documaker Studio, see the discussion of the new ProcessTriggers rule in Feature 2011.

