

## Oracle® Documaker

# iDocumaker Features and Enhancements

version 3.2

Part number: E14902-01

May 2009

Oracle Insurance announces a new release of iDocumaker version 3.2.

This document introduces iDocumaker version 3.2 and describes its features and enhancements. In addition, this document includes information about installing the system and our development and support policies.

To receive the full benefits of the new product features included in this and earlier releases, Oracle University offers a comprehensive range of training classes. For a list of courses, including fees and availability, please call 404.439.5500.

---

**NOTE:** For installation instructions, see [Installing iDocumaker](#).

---

Copyright © 2009, Oracle. All rights reserved.

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

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

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

#### **U.S. GOVERNMENT RIGHTS**

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

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

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

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

## 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, L.L.C., 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

## Chapter 1, iDocumaker Features and Enhancements

---

- 2 System Enhancements
  - 4 Locking WIP Items
  - 4 Using the New Default Key1/Key2 Profile
  - 4 The XML Export Now Includes Image Options
  - 5 Previewing TERSUB Paragraphs
  - 5 Handling Required Fields
  - 6 Editing WIP Transactions Assigned to Batch Print
  - 6 Using a URL Entry Point to Flood Data
    - 7 SERVICE\_ENTRY\_COMPOSE
    - 8 SERVICE\_ENTRY\_XMLDATAIMPORT
    - 8 SERVICE\_ENTRY\_FLOODFROMCSN
  - 10 Using a URL Entry Point for Importing Data
    - 10 SERVICE\_ENTRY\_COMPOSE
    - 11 SERVICE\_ENTRY\_XMLDATAIMPORT
    - 11 SERVICE\_ENTRY\_FLOODFROMCSN
  - 14 Configuring WIPMODE Changes
    - 16 Predefined View Action Attribute of the Screens.xml File
  - 16 Creating a List of Favorites
  - 19 Support for LDAP Authentication
  - 19 Controlling MRL Access by Users
  - 21 Using Metadata in Form Searches
    - 22 Changing the GLOBAL.XML File
    - 22 Modifying the Velocity Template
    - 23 Setting Up Filters
    - 31 Limitations
  - 32 Additional Package Policy Support



## Chapter 1

# iDocumaker Features and Enhancements

Oracle Insurance proudly announces iDocumaker version 3.2.

This document provides detailed information on the specific features and enhancements to iDocumaker. For a list of the new features and enhancements, see [System Enhancements on page 2](#).

## SYSTEM ENHANCEMENTS

The following pages describe the features which have been included in iDocumaker version 3.2.

As you review the descriptions of the new features, keep in mind that XML standards, as defined by the W3C, require you to substitute text characters that are not in XML tags (for example, between <entry> and </entry> tags) as *escape sequences*. The characters that require substitution are listed in the following table. If you cut and paste an XML example from this or other iDocumaker documentation into an XML configuration file, you will have to manually make these substitutions.

For this character	Use this escape sequence
< (less than)	&lt;
> (greater than)	&gt;
& (ampersand)	&amp;
' (apostrophe)	&apos;
“ (quotation mark)	&quot;

Here is a list of new features:

Feature	License	Description
1737	iDocumaker	Now you can lock and unlock items in WIP. For more information, see <a href="#">Locking WIP Items on page 4</a> .
1738	iDocumaker	You no longer need to list the keys in the profile table of the user database if you want the user to have access to all keys. For more information, see <a href="#">Using the New Default Key1/Key2 Profile on page 4</a> .
1740	iDocumaker	The XML export file from iDocumaker now includes the image options you can use to make the output of the XML file more useful to third-party systems. For more information, see <a href="#">The XML Export Now Includes Image Options on page 4</a> .
1741	iDocumaker	Now you can preview TERSUB paragraphs before you select them. For more information, see <a href="#">Previewing TERSUB Paragraphs on page 5</a> .
1745	iDocumaker	Now you can set up iDocumaker so users must fill in all required fields before completing a form set. This eliminates the need to return transactions for more manual data entry. For more information, see <a href="#">Handling Required Fields on page 5</a> .
1761	iDocumaker	Now you can move a WIP transaction from the batch queue, where it has been assigned to a batch print, back into edit queue so you can edit the transaction. For more information, see <a href="#">Editing WIP Transactions Assigned to Batch Print on page 6</a> .
2021	iDocumaker	Now you can use a URL entry point to flood data into the application. This eliminates the need for customizations when you post data to populate fields. For more information, see <a href="#">Using a URL Entry Point to Flood Data on page 6</a> .

Feature	License	Description
2022	iDocumaker	Now you can use a URL entry point to import data into the application. This eliminates the need to customize the system to import data before starting a transaction. For more information, see <a href="#">Using a URL Entry Point for Importing Data on page 10</a> .
2294	iDocumaker	This feature makes it easier to make changes to the WIPMODE setting in the global.xml file. Now you only have to change the screens.xml file. For more information, see <a href="#">Configuring WIPMODE Changes on page 14</a> .
2296	iDocumaker	Now you can create a list of your favorite, or most often used, forms for display on iDocumaker's Form Selection window. This makes it easier, for instance, to select the forms you need when creating new transactions. For more information, see <a href="#">Creating a List of Favorites on page 16</a> .
2340	iDocumaker	Now you can authenticate users stored in LDAP groups. For more information, see <a href="#">Support for LDAP Authentication on page 19</a> .
2341	iDocumaker	Now you can control access to master resource libraries on a per user basis. For more information, see <a href="#">Controlling MRL Access by Users on page 19</a> .
2350	iDocumaker	Now you can search for forms based on metadata elements. For more information, see <a href="#">Using Metadata in Form Searches on page 21</a> .
2395	iDocumaker	Now you can choose forms from multiple lines of business (Key2) on the Form Selection window. For more information, see <a href="#">Additional Package Policy Support on page 32</a> .

---

**NOTE:** With iDocumaker 3.2, you can use either Documaker (GenArc) or Documanager for archiving.

---

1737  
iDocumaker

## LOCKING WIP ITEMS

With iDocumaker version 3.2, you can lock and unlock items in WIP.

To enable the locking and unlocking of WIP items, you must modify the global.xml file. Set the INUSE tag/element DISPLAY attribute to “YES” as shown below. This change affects the WIPLIST view.

```
<INUSE KEY="NO" NAME="INUSE" DOCSETHEADINGS="LOCKED" DISPLAY="YES" />
```

You must also add a DPRLockWip rule to the i\_DPRGetWIPEntry request type in the docserv.ini file. The new request type should look like this:

```
[ReqType:i_DPRGetWIPEntry]
function = atcw32->ATCLogTransaction
function = atcw32->ATCLoadAttachment
function = atcw32->ATCUnloadAttachment
function = dprw32->DPRSetConfig
function = dprw32->DPRInitLby
function = dprw32->DPRGetWipFormset
function = dprw32->DPRLockWip
function = dprw32->DPRLoadFAPImages
function = dprw32->DPRUnloadExportFile
```

1738  
iDocumaker

## USING THE NEW DEFAULT KEY1/KEY2 PROFILE

With version 3.2, you no longer have to specifically profile the user for all the Key1/Key2s in the master resource library (MRL). You only have to specify the new library/config in the Lib\_Config table of the user database. This means you do not need to list the keys in the profile table of the user database if you want the user to have access to all keys.

This simplifies the process of adding a new user or configuration by letting you grant the new user access to all Key1/Key2 combinations without having to make entries in the profile table.

---

**NOTE:** You must configure Docupresentation to be the source of the MRL resources.

---

1740  
iDocumaker

## THE XML EXPORT NOW INCLUDES IMAGE OPTIONS

The XML export file from iDocumaker now includes the image options on the <SECTION> node of the XML file. This makes the output of the XML file more useful to third-party systems.

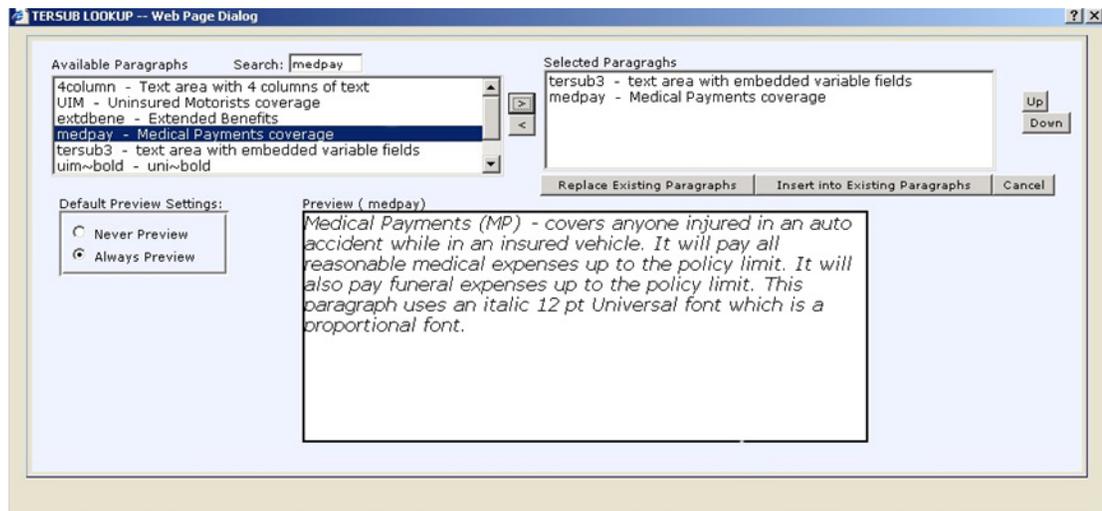
Here is an example of the XML file type:

```
<SECTION NAME="APPLICATION" OPTIONS="DS" EFFDATE="20061113"
EXPDATE="99991231" VERSION="1" REVISION="23"></SECTION>
```

## 1741 iDocumaker

### PREVIEWING TERSUB PARAGRAPHS

Now you can preview TERSUB paragraphs before you select them. When selected, the paragraphs replace or are appended to existing paragraphs in the form. Here is an example:



You must set up iDocumaker so IDS is the provider of the MRL resources and is using HTML forms/pages. You must also have an MRL that includes TERSUB forms.

To implement this feature, add this option in the global.xml file:

```
< iForms >
IFORMS PROTOCOL = "IDS"
```

**NOTE:** Keep in mind that you cannot preview TERSUB paragraphs using the WIP Edit plug-in.

## 1745 iDocumaker

### HANDLING REQUIRED FIELDS

You can now set up iDocumaker so users must use the WIP Edit plug-in or Real Time HTML to fill in all required fields before they can complete a form set. This eliminates the need to return transactions for more manual data entry.

Previously, iDocumaker only knew about the required fields on the page currently being viewed. With the use of Docupresentation as the MRL (master resource library) provider, iDocumaker will now know all of the required fields in the form set.

To set up this feature, you must modify the CONFIGURATION.INI and WIPEDIT.INI files. The CONFIGURATION.INI file is located under the Docupresentation Server. Add the following option to both INI files:

```
< Control >
RequiredFields = Yes
```

Option	Description
RequiredFields	Enter Yes if you want iDocumaker to force users to complete all required fields in the form set. The default is No.

If there are required fields in the form set that are incomplete, the application will not complete the transaction. Instead it takes you to the first incomplete required field on the form. Once you enter data into that required field, the application takes you to the next required field which needs to be filled. Once all required fields are filled, the application then lets you complete the transaction.

---

**NOTE:** If you cannot enter all the required data, you can save the transaction to WIP and complete the transaction at a later time.

---

## 1761 iDocumaker

### EDITING WIP TRANSACTIONS ASSIGNED TO BATCH PRINT

Now you can move a WIP transaction from the batch queue, where it has been assigned to a batch print, back into edit queue so you can edit the transaction. You do this by changing the status code on WIP List page to EN (ENtry).

To set up the system so you can change the status code, you must first set Display to Yes in the STATUSCODE field in the WIPKeys section of the global.xml file.

Here is an example:

```
<STATUSCODE KEY="NO" NAME="STATUSCODE" DOCSETHEADINGS="Status Code"
DISPLAY="YES">
```

## 2021 iDocumaker

### USING A URL ENTRY POINT TO FLOOD DATA

You can now use a URL entry point to flood data into the application. This eliminates the need for customizations when you post data to populate fields.

For instance, this lets you bypass the New Transaction and Form Selection/Forms Basket windows and go directly into the form via the WIP Edit plug-in for data entry. The data defined in the ientrycompose.html file for the fields on the form set is flooded into the form set.

You can set up iDocumaker to assemble a form set via a URL request (SERVICE\_ENTRY\_COMPOSE), merge the fields on the form set into the current form set (SERVICE\_ENTRY\_FLOODFROMCSN), and import FORMSET FIELDS (SERVICE\_ENTRY\_XMLDATAIMPORT) for merging at a later time.

## SERVICE\_ENTRY\_COMPOSE

This service builds a form set based on the KEY1, KEY2, and FORM sent in the HTTP POST and lets you edit those forms. This service automatically floods any data stored in the <CUSTOMSESSIONNODE><IMPORTDATA> node of the XMLSession into the working form set.

To set up this service, open the screens.xml file and follow these steps:

- 1 Find the <screen id = > node you want to update.
- 2 Insert this node:

```
<parameter name="import" global="true" value="/
ientryimportcompose.jsp" direct="false"/>
```

Make sure the URL querystring parameters or the HTTP POST parameter supplied are valid for the form set used in the application. Case is very important when sending these values:

- KEY1
- KEY2
- KEYID (Optional) The default is a unique ID supplied by the application.
- DESC (Optional) There is no default value.
- FORM
- UserId (Optional) The default is IPPUSER.
- config
- IMPORTDATA (Optional) There is no default value.

This URL/HTTP POST/GET represents data used to build or merge into the form set. The application can flood data at the DOCSET (global), FORM (form), or SECTION (local) scope.

This HTTP FORM uses these types of encodings. See the RFC prerequisite information.

- multipart/form-data
- application/x-www-form-urlencoded

Here is an example URL:

```
http://localhost/jeds/ientryimportcomposeHTML?
KEY1=DOCCDEMO&KEY2=PC&config=DOCCDEMO_IDS&UserId=DEMO1&DESC=iPPsj%2
0User&FORM=LETTER
```

The other parameters define the components required to satisfy this screen definition. The screens.xml file defines a view called:

```
<screen id="/ientryimportcompose" tcache="true">
```

This view displays an editable form set as long as the URL includes the required items.

## SERVICE\_ENTRY\_XMLDATAIMPORT

This service allows insertion of data into <CUSTOMSESSIONNODE><IMPORTDATA> of the XMLSession. To set up this service, open the screens.xml file and follow these steps:

- 1 Find the <screen id = > node you want to update.
- 2 Insert this node:

```
<parameter name="import" global="true" value="/ientryloadcsn.jsp"
direct="false"/>
```

## SERVICE\_ENTRY\_FLOODFROMCSN

This service allows data to be taken from the <CUSTOMSESSIONNODE><IMPORTDATA> node of the XMLSession and be merged into the current form set. Once the data is merged into the form set, it is removed from the <CUSTOMSESSIONNODE><IMPORTDATA> node. <CUSTOMSESSIONNODE> can contain multiple copies of the <IMPORTDATA> nodes.

To set up this service, open the screens.xml file and follow these steps:

- 1 Find the <screen id = > node you want to update.
- 2 Insert this node:

```
<parameter name="import" global="true" value="/ientrymergecsn.jsp"
direct="false"/>
```

### ATTACHMENT (Screens.xml)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <screens>
- <definition name="ippsj" config="yes">
- <screen id="/ientryimportcompose" tcache="false">
  <parameter name="import" global="true" value="/
ientryimportcompose.jsp" direct="false" />
  <parameter name="JEDSENTRYVIEWTITLE" global="true" value="iEntry
WorkOn Form(Plugin)" direct="true" />
  <parameter name="JEDSFORM" global="true" value="/
ientrypluginform.jsp" direct="false" />
  <parameter name="JEDSPAGECONTROL" global="true" value="/
ientrytoolbar.jsp" direct="false" cache="true" />
  <parameter name="JEDSRPVIEW" global="true" value="/
ientryrprview.jsp" direct="false" />
  <parameter name="JEDSJSCRIPTS" global="true" value="/
ientryinlinescript.jsp" direct="false" />
</screen>
- <screen id="/ientryimportcomposeHTML" tcache="false">
  <parameter name="import" global="true" value="/
ientryimportcompose.jsp" direct="false" />
  <parameter name="JEDSENTRYVIEWTITLE" global="true" value="iEntry
WorkOn Form(Plugin)" direct="true" />
  <parameter name="JEDSFORM" global="true" value="/
ientrypluginform.jsp" direct="false" />
  <parameter name="JEDSPAGECONTROL" global="true" value="/
ientrytoolbar.jsp" direct="false" cache="true" />
  <parameter name="JEDSRPVIEW" global="true" value="/
ientryrprview.jsp" direct="false" />
```

```

    <parameter name="JEDSJSCRIPTS" global="true" value="/
ientryinlinescript.jsp" direct="false" />
  </screen>
</definition>
</screens>

```

ATTACHMENT  
(XMLSESSION)

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<SESSION>
<CUSTOMSESSIONNODE>
<IMPORTDATA><?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36"> <FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD> </DOCSET>
</DOCUMENT>
</IMPORTDATA>
<IMPORTDATA>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36"> <FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD> </DOCSET>
</DOCUMENT>
</IMPORTDATA>
</CUSTOMSESSIONNODE>
</SESSION>
<DOCSET EFFDATE="00000000" EXPDATE="99991231" VERSION="-1">
</DOCSET>
</DOCUMENT>

```

ATTACHMENT (SAMPLE  
HTML FILE)

```

- <form action="http://localhost/ippsapp/jeds/
ientryimportcomposeHTML" enctype=" application/x-www-form-
urlencoded" method="POST">
- <select name="FORM" multiple="">
  <option value="Notification-No payment to third party"
selected="yes">Notification-No payment to third party</option>
  <option value="LETTERWITHATTACHMENT"
selected="yes">LETTERWITHATTACHMENT</option>
  <option value="ATTACHMENT REPORT OF LOSS">ATTACHMENT REPORT OF
LOSS</option>
  <option value="PAYMENT MADE">PAYMENT MADE</option>
  <option value="ADDTIFF">ADDTIFF</option>
</select>
<input type="hidden" name="config" value="DOCCDEMO_IDS" />
<input type="hidden" name="KEY1" value="DOCCDEMO" />
<input type="hidden" name="KEY2" value="PC" />
<input type="hidden" name="DESC" value="Home Owner Form" />
<input type="hidden" name="KEYID" value="2909089234890289058905" /
>
<input type="hidden" name="IMPORTDATA" value="

```

## ATTACHMENT (IMPORTDATA SAMPLE)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36"> <FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD>
</DOCSET>
</DOCUMENT>" />
<input type="submit" name="submit" value="submit" />
</form>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36">
<FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD>
</DOCSET>
</DOCUMENT>
```

2022  
iDocumaker

## USING A URL ENTRY POINT FOR IMPORTING DATA

You can now use a URL entry point to import data into the application. This eliminates the need to customize the system to import data before starting a transaction.

You can configure iDocumaker to assemble a form set via a URL request (SERVICE\_ENTRY\_COMPOSE), merge the fields on the form set into the current form set (SERVICE\_ENTRY\_FLOODFROMCSN), and import FORMSET FIELDS (SERVICE\_ENTRY\_XMLDATAIMPORT) for merging at a later time.

### SERVICE\_ENTRY\_COMPOSE

This service builds a form set based on the KEY1, KEY2, and FORM sent in the HTTP POST and lets you edit those forms. This service automatically imports data stored in the <CUSTOMSESSIONNODE><IMPORTDATA> node of the XMLSession into the current form set.

To set up this service, open the screens.xml file and follow these steps:

- 1 Find the <screen id = > node you want to update.
- 2 Insert this node:

```
<parameter name="import" global="true" value="/
ientryimportcompose.jsp" direct="false"/>
```

Make sure the URL parameters in the ientryimportcompose.jsp are valid for the form set used in the application.

- KEY1
- KEY2

- KEYID (Optional) The default is unique ID.
- DESC (Optional) There is no default.
- FORM
- UserId (Optional) The default is IPPUSER.
- config
- IMPORTDATA (Optional) There is no default.

This URL/HTTP POST represents the data that can be merged onto the form set. The application can import data at the DOCSET (global), FORM (form), or SECTION (local) scope.

This HTTP FORM uses two types of encodings. See the RFC prerequisite information.

- multipart/form-data
- application/x-www-form-urlencoded

Here is an example URL:

```
http://localhost/jeds/ientryimportcomposeHTML?
KEY1=DOCCDEMO&KEY2=PC&config=DOCCDEMO_IDS&UserId=DEMO1&DESC=iPPsj%2
0User&FORM=LETTER
```

The other parameters define the components required for this screen definition. The screens.xml file defines a view called:

```
<screen id="/ientryimportcompose" tcache="true">
```

This view displays an editable form set as long as the URL includes the required items.

## **SERVICE\_ENTRY\_XMLDATAIMPORT**

This service lets you insert data into the <CUSTOMSESSIONNODE><IMPORTDATA> node of the XMLSession. To set up this service, open the screens.xml file and follow these steps:

- 1 Find the <screen id = > node you want to update.
- 2 Insert this node:

```
<parameter name="import" global="true" value="/ientryloadcsn.jsp"
direct="false"/>
```

## **SERVICE\_ENTRY\_FLOODFROMCSN**

This service lets you take data from the <CUSTOMSESSIONNODE><IMPORTDATA> node of the XMLSession and merge it onto the current form set. Once the data is merged onto the form set, it is removed from the <CUSTOMSESSIONNODE><IMPORTDATA> node. <CUSTOMSESSIONNODE> can contain multiple copies of the <IMPORTDATA> nodes.

To set up this service, open the screens.xml file and follow these steps:

- 1 Find the <screen id = > node you want to update.

## 2 Insert this node:

ATTACHMENT  
(Screens.xml)

```
<parameter name="import" global="true" value="/ ientrymergecsn.jsp"
direct="false"/>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <screens>
- <definition name="ippsj" config="yes">
- <screen id="/ientryimportcompose" tcache="false">
  <parameter name="import" global="true" value="/
  ientryimportcompose.jsp" direct="false" />
  <parameter name="JEDSENTRYVIEWTITLE" global="true" value="iEntry
  WorkOn Form(Plugin)" direct="true" />
  <parameter name="JEDSFORM" global="true" value="/
  ientrypluginform.jsp" direct="false" />
  <parameter name="JEDSPAGECONTROL" global="true" value="/
  ientrytoolbar.jsp" direct="false" cache="true" />
  <parameter name="JEDSRPVIEW" global="true" value="/
  ientryrpview.jsp" direct="false" />
  <parameter name="JEDSJSCRIPTS" global="true" value="/
  ientryinlinescript.jsp" direct="false" />
  </screen>
- <screen id="/ientryimportcomposeHTML" tcache="false">
  <parameter name="import" global="true" value="/
  ientryimportcompose.jsp" direct="false" />
  <parameter name="JEDSENTRYVIEWTITLE" global="true" value="iEntry
  WorkOn Form(Plugin)" direct="true" />
  <parameter name="JEDSFORM" global="true" value="/
  ientrypluginform.jsp" direct="false" />
  <parameter name="JEDSPAGECONTROL" global="true" value="/
  ientrytoolbar.jsp" direct="false" cache="true" />
  <parameter name="JEDSRPVIEW" global="true" value="/
  ientryrpview.jsp" direct="false" />
  <parameter name="JEDSJSCRIPTS" global="true" value="/
  ientryinlinescript.jsp" direct="false" />
  </screen>
  </definition>
</screens>
```

ATTACHMENT  
(XMLSESSION)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<SESSION>
<CUSTOMSESSIONNODE>
<IMPORTDATA><?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36"> <FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD> </DOCSET>
</DOCUMENT>
</IMPORTDATA>
<IMPORTDATA>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36"> <FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
```

ATTACHMENT (Sample  
HTML file)

```

<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD> </DOCSET>
</DOCUMENT>
</IMPORTDATA>
</CUSTOMSESSIONNODE>
</SESSION>
<DOCSET EFFDATE="00000000" EXPDATE="99991231" VERSION="-1">
</DOCSET>
</DOCUMENT>

```

```

- <form action="http://localhost/ippsapp/jeds/
ientryimportcomposeHTML" enctype=" application/x-www-form-
urencoded" method="POST">
- <select name="FORM" multiple="">
  <option value="Notification-No payment to third party"
selected="yes">Notification-No payment to third party</option>
  <option value="LETTERWITHATTACHMENT"
selected="yes">LETTERWITHATTACHMENT</option>
  <option value="ATTACHMENT REPORT OF LOSS">ATTACHMENT REPORT OF
LOSS</option>
  <option value="PAYMENT MADE">PAYMENT MADE</option>
  <option value="ADDTIFF">ADDTIFF</option>
</select>
<input type="hidden" name="config" value="DOCCDEMO_IDS" />
<input type="hidden" name="KEY1" value="DOCCDEMO" />
<input type="hidden" name="KEY2" value="PC" />
<input type="hidden" name="DESC" value="Home Owner Form" />
<input type="hidden" name="KEYID" value="2909089234890289058905" /
>
  <input type="hidden" name="IMPORTDATA" value="
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36"> <FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD>
</DOCSET>
</DOCUMENT>" />
  <input type="submit" name="submit" value="submit" />
</form>

```

ATTACHMENT  
(ImportData sample)

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<DOCUMENT TYPE="DocuCorp Universal" VERSION="5.0">
<DOCSET EFFDATE="19800101" EXPDATE="99991231" VERSION="1"
REVISION="36">
<FIELD NAME="LossDt">12/25/2005</FIELD>
<FIELD NAME="INSURED LAST NAME">Jones</FIELD>
<FIELD NAME="INSURED FIRST NAME">David J</FIELD>
<FIELD NAME="PolicyNumber">1234567890</FIELD>
<FIELD NAME="DATE_OF_LOSS">December 25, 2005</FIELD>
</DOCSET>
</DOCUMENT>

```

2294  
iDocumaker

## CONFIGURING WIPMODE CHANGES

This enhancement minimizes the need for additional resource configuration when changing WIPMODE in the global.xml file. The screen.xml file's <screen> node's action attribute value allows the optional values shown below. The current view's parameters are executed first and then the action's attribute is evaluated and executed.

- DO.COMPOSE

This action is used for edit mode only. This enhancement allows the application to determine at runtime the configuration needed.

```
<!--Use this node if define html location
  PROTOCOL Attribute valid values:
  IDS, URL, or PLUGIN
-->
<IFORMS PROTOCOL="PLUGIN" DOMAIN="" HOST="127.0.0.1" ROOT=""
PORT="21" PASSWORD="guest" USERID="anonymous">http://
localhost:8080/skywireccm/html/forms</IFORMS>
```

You can choose from these options:

Option	Description
PLUGIN	Edit mode will use the WIPedit plug-in
IDS	Edit mode will use realtime HTML from IDS
URL	Application will use staged HTML from a URL location

- GLOBALXML.WIPMODE

The WIPMODE defines the targeted view a user wants once a transaction is imported from WIP, archive, or an extract or XML file import.

- Any predefined view in the screens.xml

This feature minimizes a previous requirement to change several resources when making changes to the WIPMODE setting in the global.xml file. With this enhancement the only change required is in the screens.xml file. The new feature automatically configures the system to present the proper view without editing template resources.

The  
GLOBALXML.WIPMODE  
ACTION attribute of  
SCREENS.XML

```
- <screen id="/ientryimport" tcache="true"
action="GlobalXML.WIPMODE">
```

This value tells the application to show certain views after executing a view's parameters. The Entry (Composition) mode in the global.xml file "PPS" child node "WIPMODE" value has these acceptable values:

- SERVICE\_ENTRY\_NEW\_TRANSACTION
- SERVICE\_ENTRY\_FORMS\_Selection
- SERVICE\_ENTRY\_WORK\_ON\_FORM
- SERVICE\_ENTRY\_WORK\_ON\_FORM\_ENTRY\_FORM

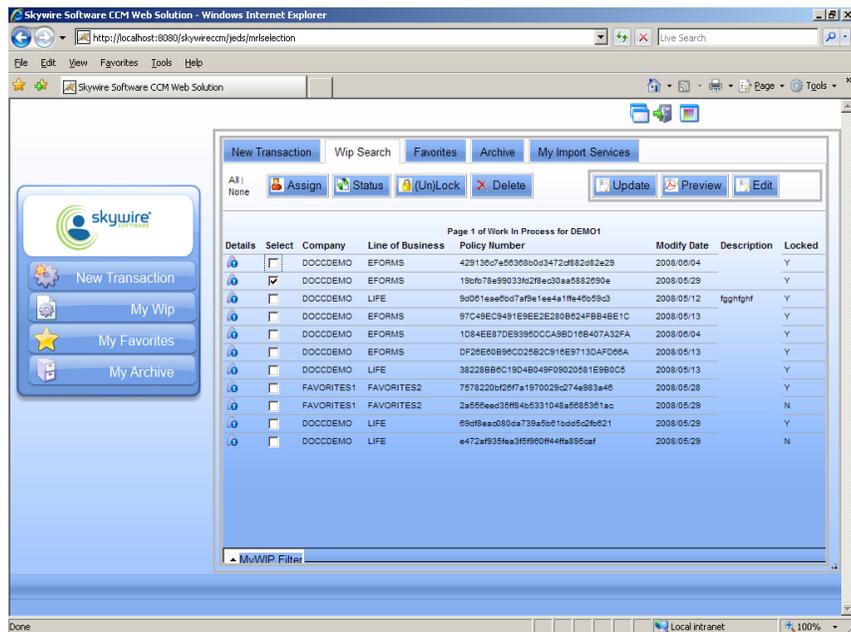
Here is a list of the modes and their associated views:

```

- <PPS>
...
...
- <!--
    Valid WIPMODES ....
        SERVICE_ENTRY_NEW_TRANSACTION
        SERVICE_ENTRY_FORMS_Selection
        SERVICE_ENTRY_WORK_ON_FORM
        SERVICE_ENTRY_WORK_ON_FORM_ENTRY_FORM
    -->
    <WIPMODE>SERVICE_ENTRY_NEW_TRANSACTION</WIPMODE>
...
...
</PPS>
    
```

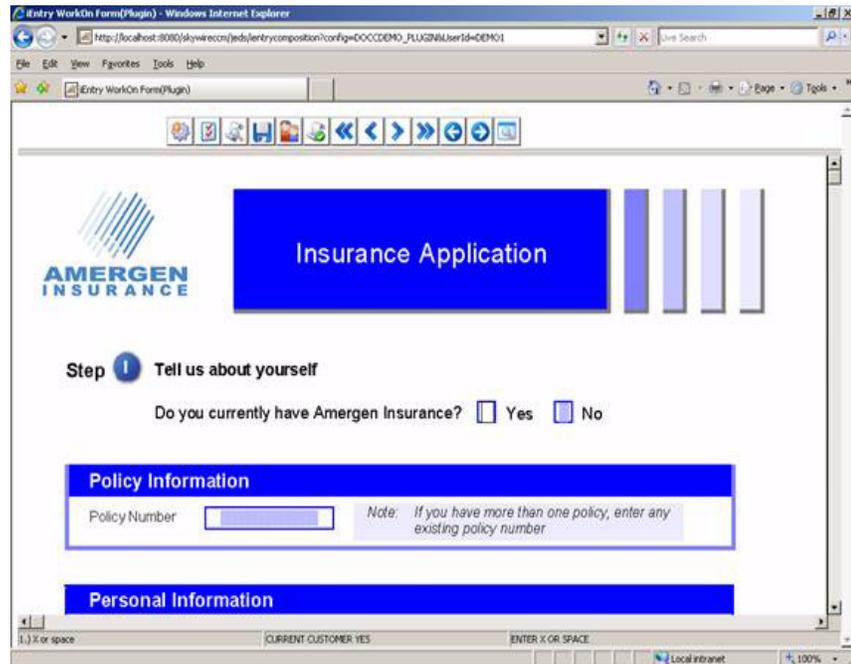
WIPMODES

SERVICE\_ENTRY\_NEW\_TRANSACTION AND SERVICE\_ENTRY\_FORMS\_Selection



WIPMODES (PLUGIN  
PROTOCOL)

SERVICE\_ENTRY\_WORK\_ON\_FORM AND  
SERVICE\_ENTRY\_WORK\_ON\_FORM\_ENTRY\_FORM



### Predefined View Action Attribute of the Screens.xml File

This value tells the application to show a certain view based on the *action* attribute value. The *action* can be any view defined in the screens.xml file. First the parameters for the current action are executed, and then the action value is evaluated and executed.

2296  
iDocumaker

### CREATING A LIST OF FAVORITES

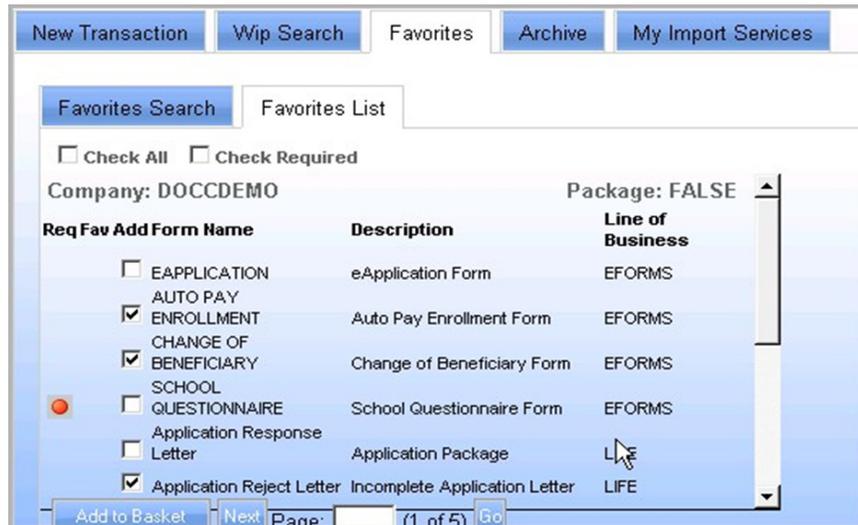
Now you can create a list of your favorite, or most often used, forms for display on iDocumaker's Form Selection window. This makes it easier, for instance, to select the forms you need when creating new transactions.

To create a favorites list, you must have...

- IDS 2.2 or higher
- Shared Objects version 11.3, patch 05

Follow these steps to create a favorites list:

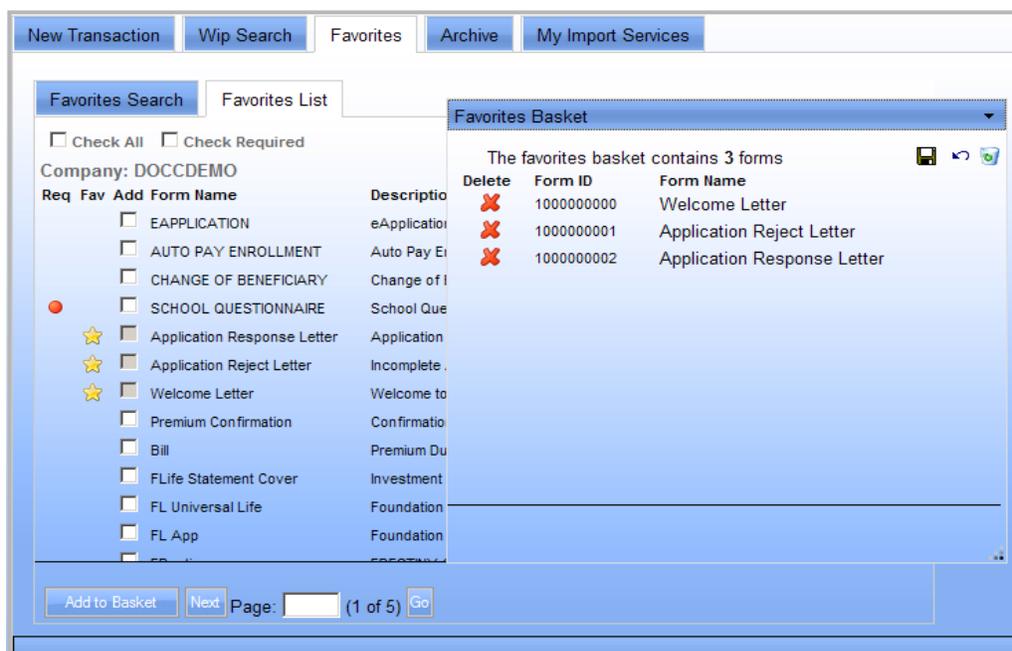
- 1 Start iDocumaker and go to the Forms Selection window. Then click the Sign In button.



- Click the Favorites tab. This takes you to the Favorites Forms Selection page.

**NOTE:** If there is no Favorites tab, IDS is not configured for favorites. If you have not created a favorites list, the favorites basket is empty. A gold star means the form is already in favorites.

- From the search results, choose the forms you want to add to the list. When you finish, click Add to Basket to append the forms in a favorites list. The Favorites Basket shows the forms you selected.



If you make changes to the favorites list, be sure to click the Save icon.

---

**NOTE:** When you create a favorites list, the system stores the list of forms in the profile.xml file which is stored in the docserv\MRL Name\UserID\ directory. The favorites list cannot contain duplicate forms.

---

**INI options** In the IPPS\_JAVA\_REQUEST.INI file, make sure you have this control group and options:

```
< ReqType:UpdateFavorite >
  Function = atcw32->ATCLogTransaction
  Function = atcw32->ATCLoadAttachment
  Function = dprw32->DPRSetConfig
  Function = atcw32->ATCUnloadAttachment
  Function = dprw32->DPRSaveUserXML
```

In the MRL INI file, add this control group and options:

```
< Favorites >
  Enabled = Yes
  Name1 = favorites
  Name2 = favorites
```

Your entries in the Name1 and Name2 fields are used by Docupresentation to generate a Favorites control group and to represent the names of the group's Key1 and Key2 values.

After you create the Favorites control group, the profile.xml file will list the forms you placed in the Favorites basket. This list is returned to iDocumaker when a request is made to IDS to get the forms for the groups for a config. The Key1 name (company) for the favorites group will be the value of the Name1 entry in the Favorites control group in the config.ini file. The Key2 name (line of business) will be Name2 value.

## Understanding the System

This feature expands your ability to get MRL information from the IDS via XML. Here is an overview of how client applications use the i\_GetMRLResource request type to get a list of group and forms that an MRL supports:

- The client application requests a list of groups by running the i\_GetMRLResource request without submitting any XML. A list of groups is returned to the client in an XML attachment called DOCUMENTSTREAM.

If you have enabled the favorites feature, one of the groups is identified as the favorites group.

- The client application or the user selects one or more of these groups to get a forms list. The forms list is obtained by sending the list of groups desired in the XML attachment XMLIMPORT to the i\_GetMRLResource rule.

The i\_GetMRLResource request type returns an XML attachment that contains forms, forms descriptions, and recipient data for all of the requested groups including the favorites group — provided the favorite feature is enabled and one of the submitted groups contains the attribute FAVORITES=TRUE.

---

2340  
iDocumaker

## SUPPORT FOR LDAP AUTHENTICATION

Now you can authenticate users stored in LDAP groups.

To enable LDAP authentication checking, the iDocumaker/LDAP system administrator must make changes in the <context> node of the context.xml file. These changes let Tomcat 6.0 know where the LDAP server is located.

Here is an example which shows the correct setting for an LDAP server on a local machine:

```
<Context antiJARLocking="true" antiResourceLocking="true"
reloadable="true">
<Realm className="org.apache.catalina.realm.JNDIRealm" debug="99"
connectionName="uid=demo,ou=users,ou=docucorp,dc=example,dc=com"
connectionPassword="demo" connectionURL="ldap://localhost:389"
roleBase="ou=groups,ou=docucorp,dc=example,dc=com" roleName="cn"
roleSearch="(uniqueMember={0})" roleSubtree="true"
userSearch="(uid={0})" userPassword="userPassword"
userPattern="uid={0},ou=users,ou=docucorp,dc=example,dc=com" />
</Context>
```

It is important that the LDAP server administrator:

- Add the user IDs to the LDAP groups in the LDAP server configuration
- Modify the context.xml file located within the deployment of iDocumaker to have the correct URL path to the LDAP server so Tomcat knows where the server is that will handle the authentication

Keep in mind...

- Users must be defined in the LDAP group
- LDAP must be accessible to iDocumaker
- There is no support for communication with third-party applications on top of LDAP
- Only the Tomcat application server is supported. Other application servers are not supported

2341  
iDocumaker

## CONTROLLING MRL ACCESS BY USERS

Now you can control access to your MRLs (master resource libraries) on a per user basis. To do so, you have to set up each user in the user database. See the documentation for the database you use for more information.

The base iDocumaker application ships with an in-memory, read-only user database. This user database is populated from the DBLOADER.XML file, which defines the library access based on user. You also have to modify the SCREENS.XML file.

---

**NOTE:** Users who do not exist in the database have access to all configurations.

---

### Modifying the DBLOADER.XML file

iDocumaker uses the DBLOADER.XML file to populate the default in-memory user database. Here are the excerpts from the DBLOADER.XML file that you must modify. The changes you should make are displayed in **red**.

```
<table name="Lib_Config">
  <columns>
    <column name="office_id" type="4"></column>
    <column name="profile_id" type="4"></column>
    <column name="library" type="12"></column>
    <column name="config" type="12"></column>
    <column name="profile_type" type="12"></column>
  </columns>
  <rows>
    <row>
      <value>2</value>
      <value>1</value>
      <value>Amergen Insurance</value>
      <value>AMERGEN</value>
      <value>I</value>
    </row>
    <row>
      <value>1</value>
      <value>2</value>
      <value>DOCCDEMO_PLUGIN</value>
      <value>DOCCDEMO_PLUGIN</value>
      <value>I</value>
    </row>
  </rows>
</table>
```

```
<table name="Location">
  <columns>
    <column name="location_id" type="4"></column>
    <column name="location_name" type="12"></column>
  </columns>
  <rows>
    <row>
      <value>1</value>
      <value>Docucorp</value>
    </row>
    <row>
      <value>2</value>
      <value>Skywire</value>
    </row>
  </rows>
</table>
```

```
<table name="Master">
  <columns>
    <column name="user_key" type="4"></column>
    <column name="id" type="12"></column>
    <column name="name" type="12"></column>
    <column name="password" type="12"></column>
    <column name="e_mail" type="12"></column>
```

```

        <column name="security" type="12"></column>
        <column name="office_id" type="4"></column>
        <column name="location_id" type="4"></column>
        <column name="transactions" type="4"></column>
    </columns>
        <row>
            <value>2</value>
            <value>DEMO2</value>
            <value>Demo 2</value>
            <value>DEMO2</value>
            <value></value>
            <value></value>
            <value>2</value>
            <value>2</value>
            <value>null</value>
        </row>
    </rows>
</table>

<table name="Profile">
    <columns>
        <column name="profile_id" type="4"></column>
        <column name="group1" type="12"></column>
        <column name="group2" type="12"></column>
    </columns>
    <rows>
        <row>
            <value>1</value>
            <value>INSURANCE PACKAGE</value>
            <value>COMMERCIAL</value>
        </row>
    </rows>
</table>

```

### Modifying the SCREENS.XML file

You also have to modify the screens.xml file to implement this feature. Comment out the lines in **red** in the screens.xml file.

```

<!--      <parameter name="title" global="true" value="Master
Resource Library Selection" direct="true"/>  -->
<parameter name="body" global="true" value="/ccm/velocity-myportal-
mrlselection.jsp?PROFILEUSERBYCONFIG=TRUE" direct="false"/>
<!--      <parameter name="body" global="true" value="/ccm/velocity-
myportal-mrlselection.jsp" direct="false"/>  -->

```

### 2350 iDocumaker

## USING METADATA IN FORM SEARCHES

iDocumaker now lets you search for forms based on metadata elements. These elements are based on form attributes set up via Documaker Studio. The form attributes include:

- Form ID
- Form description
- Category

- Any additional form metadata items set up via Studio

---

**NOTE:** This does not include including the state and effective dates combination.

---

## Changing the GLOBAL.XML File

Use the <Service Name>="CUSTOMEFILTER" node at the end of the <Services> node to activate Advanced Filtering for use with metadata searching. Here is an example:

```
<SERVICE NAME="CUSTOMEFILTER" TYPE="iFilter" POSTTYPE="STANDARD"
FILTERTYPE="1"

FILTERCLASS="com.DocuCorp.jeds.eServices.ientry.filter.ExtendedFilter"
FILTERFILELOCATION="" DISPLAY="TRUE" SEARCHABLE="NO"></SERVICE>
```

## Modifying the Velocity Template

To activate the Advance Filter window where you will select filter options for metadata searching, you must modify the velocity template for new transactions. The velocity template for new transactions is named SERVICE\_ENTRY\_NEW\_TRANSACTION.vm.

This file is in HTML format and is located, in this example, under Tomcat 6.0/webapps/skywireccm/vtemplates/doccdemo\_plugin. It would be under your web application under a similar directory structure.

Filter types To modify the velocity template for your metadata filtering needs, you need to understand the various filter types.

Type	Description
Type 1	Used for selecting a single item. This type implies you want results to equal the item selected. For example, assume you had a metadata item set up on all your forms for <i>Doc Type</i> and one of the values was <i>application</i> . You could use this type filter to select all the forms that are a <i>Doc Type</i> of <i>application</i> .
Type 2	The same as Type 1 but you can use this type with custom coding for specific needs.
Type 3	A Type 1 filter with comparison operator and an <i>ignores case</i> selection. The comparison operator can contain any values such as <i>equals</i> , <i>contains</i> , or <i>starts with</i> . It can also contain <i>ends with</i> or <i>exists</i> to indicate a match occurred if the form has the INFO node or form attribute, no matter what the value may be. You can even select the Does Not Exist option to indicate a match occurred because the form has no INFO node or form attribute set up for that metadata. This type can also include a check box to indicate if the search is case sensitive. For example, suppose you want to find all forms with a metadata item of <i>Doc Type</i> that <i>contain</i> a value of <i>DEC</i> . Even if the actual value of the metadata item was <i>DECLARATION</i> , since the value contains <i>DEC</i> , the first three letters, the form would be returned.

Type	Description
Type 4	<p>Used for free form entry into a text field along with comparison operators and an <i>ignore case</i> selection.</p> <p>The comparison operator and the Ignore Case check box work the same as in Type 3.</p> <p>You might use this type if you had forms in your MRL that had a metadata item of <i>Doc Type</i> with possible values of <i>Beneficiary Application</i> and <i>Change in Beneficiary</i>. Using this type, you could type <i>beneficiary</i> in the text field, select <i>contains</i> for your comparison operator, and click the Ignore Case check box. The system would return all forms that had either <i>Beneficiary Application</i> or <i>Change in Beneficiary</i>. In fact, it would return all forms that have the word <i>beneficiary</i> in the metadata item of <i>Doc Type</i>, regardless of where <i>beneficiary</i> appeared and the case it was in.</p>
Type 5	<p>This type is similar to Type 1 except it lets you select multiple values from a list of all possible values. For example, suppose you had a metadata item of <i>Category</i> on your forms. The possible categories are <i>eForms</i>, <i>Correspondence</i>, <i>Policy</i>, <i>Endorsements</i>, and <i>Other</i>. If you wanted to search for all the forms categorized as <i>Correspondence</i> or <i>Endorsements</i>, you could just select these two categories from the list and click the Get Forms button.</p>
Type 6	<p>This type is similar to Type 2 except it lets you select multiple values from a list of all possible values.</p>
Type 7	<p>This type is similar to Type 3 except it lets you select multiple values from a list of all possible values.</p>

## Setting Up Filters

Once you have decided which filters you need for filtering your metadata, you can modify the velocity template file, `SERVICE_ENTRY_NEW_TRANSACTION.vm`, to set up these filters. Insert the type filters you want under the `<!--Extended Filter -->` node. In keeping with examples already used, below are examples of using filters for metadata items of “Category” and “Doc Type” for filter types 1, 3, 4, 5, and 7:

```

<!-- Extended Filter -->
#set($inuse = $extendedFilter.get("INUSE"))
#if($inuse == "TRUE")
  <div dojoType="dijit.TitlePane" title="Advanced Filter"
open="false">
    <form name="newTranFilterForm" id="newTranFilterForm">
      <br />
      <table id="newTranExtFilter">
        <tr class="filterRow" type="0" name="Clear Button">
          <td colspan="5" align="right">

            <input class="ccmndButton" name="ClearFilters"
id="ClearFilters"

              topic="com_ccm_MyNewTransaction_clearFilters"
              value="Clear Filter Data" type="button">

          </td>
        </tr>
        <tr class="filterRow" type="1" name="Category">
          <td class="dataLabel" name="Type 1 - Category">Type 1 -
Category</td>
          <td class="filterListItem">

```

















**Table 1:**

Code Page 1252 (ANSI Code Page)

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

---

**NOTE:** Also, do not include these (shaded) characters in Studio as you enter the form attributes. If you do, those characters will not be recognized by iDocumaker during a metadata-based form search.

---

**2395**  
iDocumaker

## **ADDITIONAL PACKAGE POLICY SUPPORT**

Now you can choose forms from multiple lines of business (Key2) on the Form Selection window. This makes iDocumaker more similar to Documaker Workstation in package policy support.

You can choose forms from multiple lines of business if the resource library contains a Key2 with a Master DEC form identified.

---

**NOTE:** This feature does not implement the DEC page form list capabilities of Documaker Workstation.

---

