ORACLE®
INSURANCE

Oracle® Documaker

Customizing iPPS

version 3.12

Part number: E16256-01

April 2010

ORACLE®

U.S. GOVERNMENT RIGHTS

THIRD PARTY SOFTWARE NOTICES

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

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

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

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

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

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

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

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

This product includes software developed by Jean-loup Gailly and Mark Adler. This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

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

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

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

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

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

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

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

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

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

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

This product includes software components distributed by the Cryptix Foundation.

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

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

Contents

Chapter 1, Customizing iPPS

2 Adding an Agency/Config

2 Business Scenario
2 Solution 1
8 Solution 2

17 Agency/Sub Agency Profiling

17 Business Scenario
17 Solution

20 Importing Agency and Sub Agency Data

20 Business Scenario
20 Solution

22 Configuring MSMQ

22 Business Scenario
22 Solution

23 Placing the User/Profile database onto an SQL Server

23 Business Scenario
23 Solution

24 Restricting Access to Lines of Businesses

24 Business Scenario
24 Solution

26 Adding Users to iPPS for Reports To Functionality

26 Business Problem
26 Solution

29 Publishing Quote Forms in Real-time

29 Business Scenario
29 Solution

30 Starting the Issue Process from a Rating Application

30 Business Scenario
30 Solution

31 Verifying Policy Numbers

31 Business Scenario

31 Solution

32 Applying Document Naming Conventions

32 Business Scenario

32 Solution

34 Providing Default Distribution Options to Agencies

34 Business Scenario

34 Solution

37 Inserting State Stamps, Signatures, and Logos

37 Business Scenario

37 Solution

38 Importing Archive Data into a New Policy

38 Business Scenario

38 Solution

Example DFD Files and SQL Scripts

40 Example APPIDX.DFD File

46 Example ARCHIVE.DFD File

48 Example WIP.DFD File

53 Example WIPDATA.DFD File

55 Example APPIDX SQL Script

56 Example ARCHIVE SQL Script

57 Example CATALOG SQL Script

58 Example QA_MRL SQL Script

60 Example WIP SQL Script

61 Example WIP_Archive_Combined SQL Script

64 Example WIPDATA SQL Script

65 Example UserDB SQL Script

Chapter 1

Customizing iPPS

This documents outlines ways you can customize iPPS to meet your unique business needs. You will find topics on...

- Adding an Agency/Config on page 2
- Agency/Sub Agency Profiling on page 17
- Importing Agency and Sub Agency Data on page 20
- Configuring MSMQ on page 22
- Placing the User/Profile database onto an SQL Server on page 23
- Restricting Access to Lines of Businesses on page 24
- Adding Users to iPPS for Reports To Functionality on page 26
- Publishing Quote Forms in Real-time on page 29
- Starting the Issue Process from a Rating Application on page 30
- Verifying Policy Numbers on page 31
- Applying Document Naming Conventions on page 32
- Providing Default Distribution Options to Agencies on page 34
- Inserting State Stamps, Signatures, and Logos on page 37
- Importing Archive Data into a New Policy on page 38

Adding an Agency/ Config

Business Scenario

You need to add a Agency Number/Config to the DAP.INI file so IDS knows about the new configuration.

- If you are using xBase for WIP and Archive, see Solution 1 below.
- If you are using ODBC for WIP and Archive, see Solution 2 on page 8.

Solution 1

Each config will have a master CONFIG.INI file and an agency-specific INI file for information specific to this agency. These examples include the WIP, archive, and userinfo paths. This example is based in part on the scenario discussed in Agency/Sub Agency Profiling on page 17 and the sample master resource library (MRL).

NOTE: This solution assumes you are using xBase WIP and archives. Be sure to create the agency-specific folder along with a WIP and archive folder so the DBF files can be stored in these folders.

```
;**** These configs are for a new iPPS install *******
< Config:100AIC >
    INIFile        = .\mstrres\Amergen\A_100\AIC_100.ini
    INIFile        = .\mstrres\Amergen\AIC.INI
< Config:200AIC >
    INIFile        = .\mstrres\Amergen\A_200\AIC_200.ini
    INIFile        = .\mstrres\Amergen\AIC.INI _
< Configurations >
    Config         = 100AIC
    Config         = 200AIC
```

Create a new directory to hold the agency-specific INI file and also the location of the WIP and archive. Specify the paths in the agency-specific INI file to the new directory for WIP and archive, for example, create a folder named A_100 and then additional folders named WIP and Archive.

Here is an example AIC_100.INI file:

```
< ARCRET >
    APPIDX     = .\mstrres\Amergen\A_100\arc\APPIDX
    ARCPath    = .\mstrres\Amergen\A_100\arc\
    CARFile    = ARCHIVE
    CARPath    = .\mstrres\Amergen\A_100\arc\
    Catalog    = .\mstrres\Amergen\A_100\arc\CATALOG
    TempIDX    = .\mstrres\Amergen\A_100\arc\TEMP
< UserInfo >
    Path       = .\mstrres\Amergen\A_100\
    File       = userinfo
< WIPData>
    Path       = .\mstrres\Amergen\A_100\WIP\
    File       = .\mstrres\Amergen\A_100\WIP\WIP
```

Here are the settings you would need for the WIP Edit plug-in:

```
;************* Plug-in Settings *****************
```

```
< File2Dpw >
; If you need specific information for the wipedit.ini file,
; then point to one here
```

Here is an example AIC_100.INI file:

```
< ARCRET >
    APPIDX     = .\mstrres\Amergen\A_200\arc\APPIDX
    ARCPath    = .\mstrres\Amergen\A_200\arc\
    CARFile    = ARCHIVE
    CARPath    = .\mstrres\Amergen\A_200\arc\
    Catalog    = .\mstrres\Amergen\A_200\arc\CATALOG
    TempIDX    = .\mstrres\Amergen\A_200\arc\TEMP
 < UserInfo >
    Path       = .\mstrres\Amergen\A_200\
    File       = userinfo
< WIPData >
    Path       = .\mstrres\Amergen\A_200\WIP\
    File       = .\mstrres\Amergen\A_200\WIP\WIP
```

Here are the settings you would need for the WIP Edit plug-in:

```
;************* Plug-in Settings *****************
< File2Dpw >
; If you need specific information for the wipedit.ini file,
; then point to one here
```

Here is an example of the rest of the path information that would be listed in the AIC.INI file.

```
< Archival >
    MODE           = AUTO
    ArchiveMem = No
< ARCRET >
    AppIdxDfd       = .\mstrres\Amergen\arc\AppIdx.Dfd
    ExactMatch     = No
    LBLimit        = 500
< CONFIG:AIC >
    BaseDef        =
    DALFile        =
    DefLib         = .\mstrres\Amergen\DEFLIB\
    FntFile        = FONTS.FNT
    FontLib        = .\mstrres\Amergen\DEFLIB\
    Form7x         =
    FormDef        = FORM.DAT
    FormFile       =
    FormLib        = .\mstrres\Amergen\FORMS\
    HelpLib        = HELP\
    LogoFile       =
    LbyLib         = .\mstrres\Amergen\deflib\
    TableLib       = .\mstrres\Amergen\TABLE\
    TransTrigger   = RECIP.DAT
    WIPPath        = .\mstrres\Amergen\WIP\
    XRFFile        = PPS
< CONFIGURATIONS >
    CONFIG         = AIC
< MasterResource >
    BaseDef        = [CONFIG:AIC] BaseDef =
    DALFile        = [CONFIG:AIC] DalFile =
    DefLib         = [CONFIG:AIC] DefLib =
    FntFile        = [CONFIG:AIC] FntFile =
    FontLib        = [CONFIG:AIC] FontLib =
    Form7x         = [CONFIG:AIC] Form7x =
    FormDef        = [CONFIG:AIC] FormDef =
    FormFile       = [CONFIG:AIC] FormFile =
    FormLib        = [CONFIG:AIC] FormLib =
    HelpLib        = [CONFIG:AIC] HelpLib =
    LogoFile       = [CONFIG:AIC] LogoFile =
    LbyLib         = [CONFIG:AIC] LbyLib =
    TableLib       = [CONFIG:AIC] TableLib =
    TransTrigger   = [CONFIG:AIC] TransTrigger =
    XRFFile        = [CONFIG:AIC] XrfFile =
< Control >
    XrfExt     = .fxr
    ImageEXT   = .fap
    DateFormat = 24%
    AutoInserts = Yes
< Debug ]
    DeleteFiles = Yes
< Data >
    ArcNaFile      = .\mstrres\Amergen\data\nafile.dat
    ArcPolFile     = .\mstrres\Amergen\data\polfile.dat
    TrnDfdFile  = .\mstrres\Amergen\deflib\TrnDfdFl.Dfd
    ArcTrnDataFile  = .\mstrres\Amergen\data\newtrn.dat
< WIPData >
    Key1 = KEY1
    Key2 = KEY2
    KeyID = KEYID
    DESC = DESC
    TranCode = TRANCODE
    StatusCode = STATUSCODE
    GUIDKey = GUIDKEY
    LocID = LOCID
    SubLocID = SUBLOCID
    Jurisdictn = JURISDICTN
    TRNName = TRNNAME
;Needed for Check Policy to work
< DocSetNames >
    GroupName1 = KEY1
    GroupName2 = KEY2
    GroupName3 = NULL
    FormName = NULL
    TransactionID = KEYID
    StatusCode = STATUSCODE
    TranCode = TRANCODE
    ArchiveKey = ARCKEY
    ArchiveDate = CREATETIME
< IPPValues >
    IPPExportRecips = True
< STATUS_CD >
    Archive        = AR
    Assign         = A
    BatchPrint     = B
    Combine        = CO
    Duplicate      = DU
    Printed        = P
    Quote          = Q
    Transmit       = T
    WIP            = W
; Logo to be used when proofing a PDF in iPPS
< WatermarkLogo >
    Logo           =.\mstrres\Amergen\deflib\Proof.LOG
    Top            = 4050
    Left           =5075
    Pages          =all
    Color          =0
```

Here are the settings you would need for the WIP Edit plug-in:

```
;************* Plug-in Settings *****************
;Items needed to be sent to the plug-in during initialization
;the wipedit.ini has plug-in specific settings
< File2Dpw >
    XRFToken = .\mstrres\Amergen\DEFLIB\PPS.fxr
    INIToken = .\mstrres\Amergen\wipedit.ini
< WIP2DPW >
;   Debug = Yes
    Menu = wipedit.res
< WIPLock >
    MatchUserid = Ignore
    UnMatchUserid = Ignore
< INI2XML >
    PutURL       = 127.0.0.1:80
    UserID       = ~GetAttach USERID
    Password     = ~GetAttach PASSWORD
< PrtType:DPW >
    PrintFunc   = DPWPrint
    Module      = DPWW32
;   Debug       = Yes
    FormSetID   = ~GetAttach FORMSETID
    RECNUM      = ~GetAttach RECNUM
    KeyID     = ~GetAttach KEYID
; Specifics for each print type including file naming convention
< PrtType:PDF >
    FileName = ~KEYID ~TRANCODE _ ~DALRUN guid.dal _PDF.pdf
< PrtType:PCL >
    DownloadFonts   = Yes,Enabled
    Module          = PCLOS2
    PrintFunc       = PCLPrint
    SendOverlays    = No,Enabled
    MultipleCopies  = Yes
    TemplateFields  = No
    AdjLeftMargin = Yes
    FileName = ~KEYID ~TRANCODE .pcl
< PrtType:XML >
    FileName = ~KEYID ~TRANCODE .xml
< PrtType:V2 >
    FileName = ~KEYID ~TRANCODE .v2
; Mapping from WIP Record to Archive record
```

```
< AfeWip2ArchiveRecord >
    KEY1          = KEY1
    KEY2          = KEY2
    KEYID         = KEYID
    RECTYPE       = RECTYPE
    CREATETIME    = MODIFYTIME,X
    ORIGUSER      = ORIGUSER
    CURRUSER      = CURRUSER
    MODIFYTIME    = MODIFYTIME,X
    FORMSETID     = FORMSETID
    TRANCODE      = TRANCODE
    STATUSCODE    = STATUSCODE
    FROMUSER      = FROMUSER
    FROMTIME      = FROMTIME
    TOUSER        = TOUSER
    TOTIME        = TOTIME
    DESC          = DESC
    INUSE         = INUSE
    ARCKEY        = ARCKEY
    APPDATA       = APPDATA
    RECNUM        = RECNUM
    TRNNAME       = TRNNAME
; Setup for complete print. This is to designate the Home Office
; PDF and auto export
< Complete >
    CompleteType = COMP1
    CompleteType = COMP2
< Complete:COMP1 >
    FileType = PDF
    FileName = ~KEYID ~TRANCODE.pdf
    FileExt = pdf
    FilePath = .\mstrres\Amergen\complete\
    Recipient = HOME OFFICE
< Complete:COMP2 >
    FileType = XML
    FileName = ~KEYID ~TRANCODE.xml
    FileExt = xml
    FilePath = .\mstrres\Amergen\complete\
    Recipient = ALLRECIPIENTS
```

Use the DSICOTB.exe in the docserv directory to test getting resources from Docupresentment. You can run the i_GetMRLResource request type to see if you can pull the company and lines of business. This will help you determine if your pathing is set up correctly.

NOTE: For more information on adding the agency number or returning the locationprofile.xml file, see Agency/Sub Agency Profiling on page 17.

Follow these steps:

1 Copy a <CONFIG> node into the global.xml (not the SYSTEM config), or use the global_template.xml. Change the Name attribute to the new Agency/Config. Set the Library attribute to the description from the doGetLocationProfileXML for that agency. Remember these values because you will need to add them to the user database if profiling is needed.

- Add or remove the Distribution options from the <IROUTE> nodes as needed.

  The removal of the <IROUTE> nodes prevents the agent from seeing possible distribution options such as XMLExport, V2, FieldsOnly, and so on. See the OW0006 agency as an example of removing these distribution options.

- Change the path to the Default Distribution Options XML for this agent.
- Modify the actual DefaultDistributionOptions.XML file with this agent's new defaults.

2 If the agency needs to have a subset of the company and line of business, add the Agency Number to the user database. See Agency/Sub Agency Profiling on page 17 for more information.

3 Add new users to the USERINFO.DBF file for the new agency. See Adding Users to iPPS for Reports To Functionality on page 26 for more information.

4 Make sure the location profile service provides a profile for the new agency. See Agency/Sub Agency Profiling on page 17 for more information.

If you have not automated this step, modify the following file to make sure your new agency will get a profile returned:

```
c:\Inetpub\wwwroot\_iPPS312_SERVICES\locationprofile.asp
```

5 If needed, make sure the service for importing agency-specific data returns data for the new agency. See Importing Agency and Sub Agency Data on page 20 for more information. If you have not automated this step, modify the following file to return the XML file with the data to import:

```
c:\Inetpub\wwwroot\_iPPS312_SERVICES\cusimportdata.asp
```

This data includes items that would be treated as default data.

6 Modify the Verify KeyId service to validate the policy numbers for the new agency. See Verifying Policy Numbers on page 31 for more information. If you have not automated this step, modify the following file to add additional validation:

```
c:\Inetpub\wwwroot\_iPPS312_SERVICES\cusverifykeyid.asp
```

7 Make sure your WIP and archive folders have been created underneath your agency-specific folder.

Solution 2

Each configuration will have a master config.ini file and an agency-specific INI file for information specific to the agency, such as the WIP and archive tables, and the userinfo table.

NOTE: This solution assumes you are using ODBC for WIP and archives.

Here is an example DAP.INI file:

```
< Config:OW0006 >
    INIFile         = \OracleWorld\OW0006\OW0006.ini
    INIFile         = \OracleWorld\fsiuser.ini
    INIFile         = \OracleWorld\fsisys.ini
< Config:OWA321 >
    INIFile         = \OracleWorld\OWA321\OWA321.ini
    INIFile         = \OracleWorld\fsiuser.ini
    INIFile         = \OracleWorld\fsisys.ini_
< Configurations >
    Config          = OW0006
    Config          = OWA321
```

Create a new directory in c:\OracleWorld\ to hold the agency-specific INI file and userinfo database files (such as c:\OracleWorld\OW0006\). You will need to change the items of the existing agency to the new agency.

Here is an example OW0006.INI file:

```
< ODBC_FileConvert >
    WIP            = OW0006_WIP
    WIPData        = OW0006_WIPDATA
    APPIDX         = OW0006_APPIDX
    Archive        = OW0006_ARCHIVE
    Catalog        = OW0006_CATALOG
< UserInfo >
    Path           = \OracleWorld\OW0006\
    File           = userinfo
;*** Complete print for OW copies *****
< Complete >
    CompleteType    = COMP1
    CompleteType    = COMP2
< Complete:COMP1 >
    FileType       = PDF
    FileName       = ~KEYID ~TRANCODE _~LOCID ~DALRUN date.dal .PDF
    FileExt        = pdf
    FilePath       = \\FTPSERVER\C$\IPPS\OW0006
    Recipient      = COMPANY
    PersistOutput  = Yes
< Complete:COMP2 >
    FileType       = XML
    FileName       = ~KEYID ~TRANCODE _~LOCID ~DALRUN date.dal .xml
    FileExt        = xml
    FilePath       = \\FTPSERVER\C$\IPPS\OW0006
    Recipient      = ALLRECIPIENTS
    PersistOutput  = Yes
```

Here is an example OWA321.INI file:

```
< ODBC_FileConvert >
    WIP            = OWA321_WIP
    WIPData        = OWA321_WIPDATA
    APPIDX         = OWA321_APPIDX
    Archive        = OWA321_ARCHIVE
    Catalog        = OWA321_CATALOG
< UserInfo >
    Path           = \OracleWorld\OWA321\
    File           = userinfo
< Complete:COMP1 >
    FileType       = PDF
    FileName       = ~KEYID ~TRANCODE _~LOCID ~DALRUN date.dal .PDF
    FileExt        = pdf
    FilePath       = \\FTPSERVER\C$\IPPS\OWA321
    Recipient      = COMPANY
    PersistOutput  = Yes
< Complete:COMP2 >
    FileType       = XML
    FileName       = ~KEYID ~TRANCODE _~LOCID ~DALRUN date.dal .xml
    FileExt        = xml
    FilePath       = \\FTPSERVER\C$\IPPS\OWA321
    Recipient      = ALLRECIPIENTS
    PersistOutput  = Yes
```

The rest of the path information is listed in the FSIUSER.INI file. Here is an example FSIUSER.INI file:

```
< AFEWIP2ArchiveRecord >
    APPData         = APPDATA
    ARCKey          = ARCKEY
    CreateTime      = MODIFYTIME,X
    CurrUser        = CURRUSER
    Desc            = DESC
    FormSetID       = FORMSETID
    FromTime        = FROMTIME
    FromUser        = FROMUSER
    InUse           = INUSE
    Key1            = KEY1
    Key2            = KEY2
    KeyID           = KEYID
    ModifyTime      = MODIFYTIME,X
    OrigUser        = ORIGUSER
    RecNum          = RECNUM
    RecType         = RECTYPE
    StatusCode      = STATUSCODE
    ToTime          = TOTIME
    ToUser          = TOUSER
    TranCode        = TRANCODE
    TrnName         = TRNNAME
< Archival >
    ArchiveMem      = Yes
    Mode            = AUTO
< ARCRET >
    AppIdxDfd      = \OracleWorld\deflib\AppIdx.Dfd
    CarFileDFD     =  \OracleWorld\deflib\Archive.Dfd
    APPIDX         = APPIDX
    CARFile        = ARCHIVE
    Catalog        = CATALOG
    TempIDX        = TEMP
    ExactMatch     = No
    LBLimit        = 500
< DBTable:ARCHIVE >
    DBHandler      = CD9B9CDB
    UniqueTag      = ARCKEY
< DBTable:APPIDX >
    DBHandler      = CD9B9CDB
  UniqueTag = ARCKEY
    DefaultTag     = CREATETIME
< DBTable:CATALOG >
    DBHandler      = CD9B9CDB
< Config:OW >
    ArcPath          = \OracleWorld\Arc\
    BaseDef          = OW
    BDFFile          = OW_QA
    BDFLib           = \OracleWorld\BDFLib\
    CARPath          = \OracleWorld\Arc\
    DALFile          = OW_QA
    DDTFile          = OW_QA
    DEFLib           = \OracleWorld\DEFLIB\
    DictionaryFile   = XDB.DBF
    EDLLib           = \OracleWorld\EDLLib\
    FieldBaseFile    = FDB.DBF
    FNTFile          = fonts
    FontLib          = \OracleWorld\DEFLIB\
    FORFile          = OW_QA
    FORLib           = \OracleWorld\FORLib\
    FormDef          = FORM.DAT
    FormFile         = OW_QA
    FormLib          = \OracleWorld\FORMS\
    FormsetTrigger   = SETRCPTB.INI
    GRPFile          = OW_QA
    GRPLib           = \OracleWorld\GRPLib\
    HelpLib          = \OracleWorld\HELP\
    LBYLib           = \OracleWorld\DEFLIB\
    LogoFile         = OW_QA
    LogoLib          = \OracleWorld\FORMS\
    TableLib         = \OracleWorld\TABLE\
;   WIPPath          = \OracleWorld\WIP\
    XDDFile          = OW_QA
    XRFFile          = fmgenos2
< Configurations >
    Config           = OW
< Control >
    AutoInserts      = Yes
    DateFormat       = 24%
    ImageEXT         = .fap
    XRFExt           = .fxr
< Data >
    AfgJobFile       =
    ArcNaFile        = \OracleWorld\data\nafile.dat
    ArcPolFile       = \OracleWorld\data\polfile.dat
    ArcTrnDataFile   = \OracleWorld\data\newtrn.dat
    DataPath         = .\DATA\
    SetRcpTb         =
    TranslationFile  = Translat.ini
    TrnDfdFile       = \OracleWorld\deflib\TrnDfdFl.Dfd
< DBHandler:CD62B30F >
    Class            = CB5
    File             = \OracleWorld\DEFLIB\MASTER.LBY
< DBHandler:CD9B9CDB >
    Class             = ODBC
    CreateIndex       = No
    CreateTable       = No
    Debug             = Yes
    PassWd            = 001600
    Server            = DocumakerDB
    UserID            = sa
< DBHandler:ODBC_PROD >
    Class             = ODBC
    CreateIndex       = No
    CreateTable       = No
    Debug             = No
    PassWd            = 001600
    Server            = DocumakerDB_PROD
    UserID            = sa
< DBTable:OW_QA >
    DBHandler         = CD9B9CDB
< DBTable:OW_QAC >
    DBHandler         = CD9B9CDB
    UniqueTag         = CATALOGID
< DBTable:OW_QAD >
    DBHandler         = CD9B9CDB
    UniqueTag         = ARCKEY+SEQ_NUM
< DBTable:OW_QAL >
    DBHandler         = CD9B9CDB
    UniqueTag         = DATE+TIME
< DBTable:OW_PROD >
    DBHandler         = ODBC_PROD
< DBTable:OW_PRODC >
    DBHandler         = ODBC_PROD
    UniqueTag         = CATALOGID
< DBTable:OW_PRODD >
    DBHandler         = ODBC_PROD
    UniqueTag         = ARCKEY+SEQ_NUM
< DBTable:OW_PRODL >
    DBHandler         = ODBC_PROD
    UniqueTag         = DATE+TIME
< DBTable:OW_DEV >
    DBHandler         = CD62B30F
< Debug >
    DeleteFiles       = No
< DocSetNames >
    ArchiveDate       = CREATETIME
    ArchiveKey      = ARCKEY
    FormName        = NULL
    GroupName1      = KEY1
    GroupName2      = KEY2
    GroupName3      = NULL
    StatusCode      = STATUSCODE
    TranCode        = TRANCODE
    TransactionID   = KEYID
< Environment >
    DO_LOGON        = Yes
    FSISYSINI       = FSISYS.INI
< ExportFields >
    End             = END OF FIELD
    Field           = AGYAD1
    Field           = AGYAD2
    Field           = AGYCTY
    Field           = AGYNAM
    Field           = AGYST
    Field           = AGYZIP
    Field           = FEEDESC1
    Field           = CSIGNEDLOC
    SingleLine      = no
    Start           = START OF FIELD
< ExportFormats >
    01              = ;TD;Standard Export;TRNW32->TRNExportV2;
    02              = ;FX;Field-only Export;TRNW32->TRNExportFields;
< FixOptions >
    UseXDBRule      = Yes
< Library:OW_DEV >
    DBTable         = OW_DEV
< Library:OW_QA >
    Catalog         = OW_QAC
    DBTable         = OW_QAD
    LBYLogFile      = OW_QAL
< Library:OW_PROD >
    Catalog         = OW_PRODC
    DBTable         = OW_PRODD
    LBYLogFile      = OW_PRODL
< LibraryManager >
    Class           = GA;Georgia resource
    Class           = TX;Texas resource
    Class           = MD;Maryland resource
    Library         = OW_DEV
    Library         = OW_QA
    Project         = P001;Project 001
    Project         = P002;Project 002
    Project         = P003;Project 003
< MasterResource >
    BaseDef         = <CONFIG:OW> BaseDef =
    BDFFile         = <CONFIG:OW> BDFFile =
    BDFLib          = <CONFIG:OW> BDFLib =
    CompLib         = <CONFIG:OW> CompLib =
    DalFile         = <CONFIG:OW> DalFile =
    DDTFile         = <CONFIG:OW> DDTFile =
    DDTLib          = <CONFIG:OW> DDTLib =
    DefLib          = <CONFIG:OW> DefLib =
    DictionaryFile  = <CONFIG:OW> DictionaryFile =
    EDLLib          = <CONFIG:OW> EDLLib =
    FieldBaseFile   = <CONFIG:OW> FieldBaseFile =
    FntFile         = <CONFIG:OW> FntFile =
    FontLib         = <CONFIG:OW> FontLib =
    FORFile         = <CONFIG:OW> FORFile =
    FORLib          = <CONFIG:OW> FORLib =
    Form7x          = <CONFIG:OW> Form7x =
    FormDef         = <CONFIG:OW> FormDef =
    FormFile        = <CONFIG:OW> FormFile =
    FormLib         = <CONFIG:OW> FormLib =
    FormSetTrigger  = <CONFIG:OW> FormsetTrigger =
    GRPFile         = <CONFIG:OW> GRPFile =
    GRPLib          = <CONFIG:OW> GRPLib =
    HelpLib         = <CONFIG:OW> HelpLib =
    LbyLib          = <CONFIG:OW> LbyLib =
    LogoFile        = <CONFIG:OW> LogoFile =
    LogoLib         = <CONFIG:OW> LogoLib =
    TableLib        = <CONFIG:OW> TableLib =
    TablePath       = <MASTERRESOURCE> TableLib =
    TransTrigger    = [CONFIG:AIC] TransTrigger =
    XDDFile         = <CONFIG:OW> XDDFile =
    XRFFile         = <CONFIG:OW> XrfFile =
< ODBC_FileConvert >
    OW_QA           = OW_QA
    OW_QAC          = OW_QAC
    OW_QAD          = OW_QAD
    OW_QAL          = OW_QAL
    OW_PROD         = OW_PROD
    OW_PRODC        = OW_PRODC
    OW_PRODD        = OW_PRODD
    OW_PRODL        = OW_PRODL
< RunMode >
    CompiledFAP     = No
< SignOn >
    UserID          = FORMAKER
< Status_CD >
    Archive         = AR
    Assign          = A
    BatchPrint      = B
    Combine         = CO
    Duplicate       = DU
    Printed         = P
    Quote           = Q
    Transmit        = T
    WIP             = W
< TABLES >
    Path            = [MASTERRESOURCE] TABLELIB=

;************* WIP Edit plug-in settings *****************
;Items to be sent to the WIP Edit plug-in during initialization
;the WIPEDIT.INI file has plug-in specific settings
< File2Dpw >
    XRFToken = \OracleWorld\DEFLIB\fmgenos2.fxr
    INIToken = \OracleWorld\wipedit.ini
< WIP2dpw >
```

```
;   Debug = Yes
    Menu = \OracleWorld\wipedit.res
< WIPLock >
    MatchUserID   = Ignore
    UnMatchUserID = Ignore
< INI2XML >
    PutURL       = 127.0.0.1:80
    HTTPUserID   = ipps
    HTTPPassword = ipps
< PrtType:DPW >
    PrintFunc   = DPWPrint
    Module      = DPWW32
    Debug       = Yes
    formSetID   = ~GetAttach FORMSETID
    RecNum      = ~GetAttach RECNUM
    KeyID       = ~GetAttach KEYID
< IPPValues >
    IPPExportRecips = True

;***iIPPS Print type settings ****
< PrtType:FIELDSONLY >
    FileName        = ~KEYID ~TRANCODE .fld
< PrtType:PCL >
    AdjLeftMargin   = Yes
    DownloadFonts   = Yes,Enabled
    FileName        = ~KEYID ~TRANCODE .pcl
    Module          = PCLOS2
    MultipleCopies  = Yes
    PrintFunc       = PCLPrint
    SendOverlays    = No,Enabled
    TemplateFields  = No
< PrtType:PDF >
    FileName        = ~KEYID ~TRANCODE _ ~DALRUN guid.dal .pdf
< PrtType:V2 >
    FileName        = ~KEYID ~TRANCODE .v2
< PrtType:XML >
    FileName        = ~KEYID ~TRANCODE .xml
< WatermarkLogo >
    Color           = 0
    Left            = 5075
    Logo            = \OracleWorld\deflib\Proof.LOG
    Pages           = all
    Top             = 4050
< DBTable:WIP >
    DBHandler       = CD9B9CDB
    UniqueTag       = DocTag
< DBTable:WIPData >
    DBHandler       = CD9B9CDB
    UniqueTag       = FormSetID
< WIPData >
    Desc            = DESC
    File            = WIP
    GUIDKey         = GUIDKEY
    Jurisdictn      = JURISDICTN
    Key1            = KEY1
    Key2            = KEY2
    KeyID           = KEYID
    LocID           = LOCID
    StatusCode      = STATUSCODE
    SubLocID        = SUBLOCID
    TranCode        = TRANCODE
    TrnName         = TRNNAME
    RecNum          = UNIQUE_ID
    DatabaseWIP     = Yes
    WIPDataDFD      = \OracleWorld\deflib\wipdata.dfd
    WIPDFDFile      = \OracleWorld\deflib\wip.dfd
    Path            = \OracleWorld\WIP\
< LSS_Inserts >
    LSS_Signature   = LSS_SIGNATURE_{%LOCID}
```

Follow these steps:

1. Copy a <CONFIG> node into the global.xml (not the SYSTEM config), or use the global_template.xml. Change the Name attribute to the new Agency/Config. Set the Library attribute to the description from the doGetLocationProfileXML for that agency. Remember these values because you will need to add them to the user database if profiling is needed.

    - Add or remove the Distribution options from the <IROUTE> nodes as needed.

      The removal of the <IROUTE> nodes prevents the agent from seeing possible distribution options such as XMLExport, V2, FieldsOnly, and so on. See the OW0006 agency as an example of removing these distribution options.

    - Change the path to the Default Distribution Options XML for this agent.

    - Modify the actual DefaultDistributionOptions.XML file with this agent's new defaults.

2. If the agency needs to have a subset of the company and line of business, add the Agency Number to the user database. See Agency/Sub Agency Profiling on page 17 for more information.

3. Add new users to the USERINFO.DBF file for the new agency. See Adding Users to iPPS for Reports To Functionality on page 26 for more information.

4. Make sure the location profile service provides a profile for the new agency. See Agency/Sub Agency Profiling on page 17 for more information.

   If you have not automated this step, modify the following file to make sure your new agency will get a profile returned:

   ```
   c:\Inetpub\wwwroot\_iPPS312_SERVICES\locationprofile.asp
   ```

5. If needed, make sure the service for importing agency-specific data returns data for the new agency. See Importing Agency and Sub Agency Data on page 20 for more information. If you have not automated this step, modify the following file to return the XML file with the data to import:

   ```
   c:\Inetpub\wwwroot\_iPPS312_SERVICES\cusimportdata.asp
   ```

   This data includes items that would be treated as default data.

6 Modify the Verify KeyId service to validate the policy numbers for the new agency. See Verifying Policy Numbers on page 31 for more information. If you have not automated this step, modify the following file to add additional validation:

```
c:\Inetpub\wwwroot\_iPPS312_SERVICES\cusverifykeyid.asp
```

7 Create the WIP and ARCHIVE tables in the SQLServer. Modify the WIP_Archive_Combined.SQL script to reflect the agency you are trying to add. Then run this script in the SQLServer Management Studio for the environment you are updating.

NOTE: See Example WIP_Archive_Combined SQL Script on page 61 for an example of this script.

Agency/Sub Agency Profiling

Business Scenario

You would like to define which agencies a user has access to. Each agency needs to be profiled for its valid sub agencies. In addition, each agency needs to be profiled so only valid states are available for selection by the agent.

Solution

The users and agencies you write for are typically maintained in the carriers's system, possibly in a database used to service the agent portal. iPPS is set up to use this information obtained via a REST service formatted according Schema. To use this functionality, a few setup requirements are required:

- You must add an entry to the global.xml file at the SYSTEM config level to tell the system where the URL is located for the profiling service. Here is an example:

```
<DOGETLOCATIONPROFILEURL>http://client.oracle.com/iPPS_Services/
locationprofile.asp</ DOGETLOCATIONPROFILEURL>
```

- This URL is the service provider for a doGetLocationProfileXML which tells the system which Agency Number and Description, Agency Sub Number and Description, and Tax State value and description are available.

Here is an example:

```
<doGetLocationProfileResponse xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
xsi:noNamespaceSchemaLocation="doGetLocationProfileResponse.xsd"
schemaVersion="1.0">

    <LocId id="100AIC" desc="Oracle Underwriters (100)"
mainlocationid="100">
        <SubLocId id="0" desc="Oracle Underwriters - Contract"/>
        <SubLocId id="1" desc="Oracle Underwriters - Brokerage"/>
        <SubLocId id="2" desc="Oracle Underwriters - Brokerage,
Orlando">
            <Jurisdiction id="AL" desc="Alabama"/>
            <Jurisdiction id="AR" desc="Arkansas"/>
        </SubLocId>
        <Jurisdiction id="AL" desc="Alabama"/>
        <Jurisdiction id="AR" desc="Arkansas"/>
        <Jurisdiction id="FL" desc="Florida"/>
    </LocId>
    <LocId id="200OIC" desc="Oracle Insurance House (200)"
mainlocationid="200">
        <SubLocId id="0" desc="Oracle Insurance House - Contract"/>
        <SubLocId id="1" desc="Oracle Insurance House - Program"/>
        <Jurisdiction id="DE" desc="Deleware"/>
        <Jurisdiction id="OH" desc="Ohio"/>
    </LocId>
</doGetLocationProfileResponse>
```

Here is another example:

```
<doGetLocationProfileResponse
xsi:noNamespaceSchemaLocation="doGetLocationProfileResponse.xsd"
schemaVersion="1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <LocId id="100AIC" desc=" Oracle Underwriters (100)">
        <SubLocId id="0" desc=" Oracle Underwriters - Contract"/>
        <SubLocId id="1" desc=" Oracle Underwriters - Brokerage"/>
        <Jurisdiction id="AL" desc="Alabama"></Jurisdiction>
    </LocId>
<LocId id="200AIC" desc=" Oracle Insurance House (200)">
        <SubLocId id="0" desc="Oracle Insurance House - Contract"/>
        <SubLocId id="1" desc="Oracle Insurance House - Program"/>
        <Jurisdiction id="KY" desc="Kentucky"></Jurisdiction>
    </LocId>
    <LocId id="300AIC" desc="Oracle Insurance Brokers (AIC)"
mainlocationid="300">
        <SubLocId id="0" desc=" Oracle Insurance Brokers - Contract"/
>           <SubLocId id="1" desc=" Oracle Insurance Brokers -
Brokerage"/>
        <Jurisdiction id="MS" desc="Mississippi"></Jurisdiction>
        <Jurisdiction id="TN" desc="Tennessee"></Jurisdiction>
    </LocId>
    <LocId id="300OIC" desc=" Oracle Insurance Brokers (OIC)"
mainlocationid="300">
        <SubLocId id="0" desc=" Oracle Insurance Brokers - Contract"/
>           <SubLocId id="2" desc=" Oracle Insurance Brokers -
Program"/>
        <Jurisdiction id="AZ" desc="Arizona"></Jurisdiction>
        <Jurisdiction id="NC" desc="North Carolina"></Jurisdiction>
        <Jurisdiction id="VA" desc="Virginia"></Jurisdiction>
    </LocId>
</doGetLocationProfileResponse>
```

The second example differs from the first example in that an optional mainlocationid attribute is included on the location (<LocId>).

In this example, agency 300 writes business for Amergen Insurance (AIC) and Oracle Insurance (OIC). There is an additional differentiating factor in that the states that are permissible differ for each of these companies.

To profile different states per company, you will need to list them as separate locations (<LociId>). In this example, location ID 300AIC represents the agency/config for the Amergen Company.

You will also notice that this particular agency/config for Amergen Insurance is limited to Mississippi and Tennessee. The location if 300AIC represents the agency/config for the Oracle company. It also has state limitations to Arizona, North Carolina, and Virginia. You have to list these as separate location IDs so you can restrict the states for each company.

These locations have the mainlocationid attribute set to 300 so that you can group/correlate all premiums written back to this main location. The value stored in the mainlocationid attribute will be the value stored in the WIP index and will populate on the forms in the package for this agency number.

The system will use this information and present it to the user. When the user accesses the system, the system provides a selection of libraries from which the user can choose. The listing is comprised of each <Locid id=”/> in the doGetLocationProfileXML.

In the examples above, the user will be presented with Oracle Underwriters (100) and Oracle Insurance House (200) as choices in the first example. In the second example, the user will be presented with Oracle Underwriters (100), Oracle Insurance House (200), Oracle Insurance Brokers (AIC), and Oracle Insurance Brokers (OIC).

Each location (<LocId> has access to all jurisdictions (<Jurisdiction>) listed as its children. Any jurisdiction list underneath the sub agency (<SubLocId>) is a restriction of that jurisdiction. In other words, the user will not have access to that jurisdiction listed as a choice.

The purpose of the jurisdiction listing is to control which states are available to the user. The jurisdiction listing populates two fields on the Company/Line of Business Selection window. The jurisdiction field, Tax State, is populated from this listing and is used to provide data for placing the appropriate state stamp on the form.

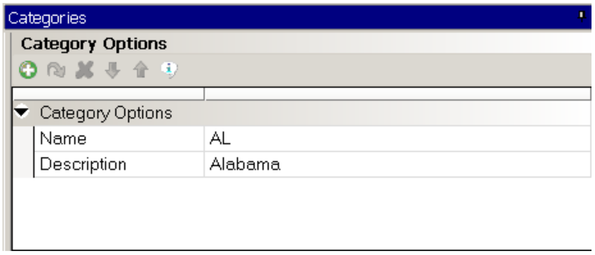
The other field, Category, is used as a forms list filter. The user can choose the company, line of business, and state, which will narrow the list of forms on the Forms Selection window. If a state category is not chosen, then no forms from that state will appear in the forms list.

To use this filter functionality, you must set up categories in the FORM.DAT file. Here is an example:

```
;Amergen;GENERAL LIABILITY;CG0108  11-1985;Alabama
Changes||CATEGORY=AL||;NM;;cg0108|A<ORIGINAL(1),RETAIL
AGENT(1),GENERAL AGENT(0),EXTRA COPY(0),HOME OFFICE(0)>;
```

Setting up categories in Studio

In Documaker Studio, open the Application Definition (BDF) file and click on the Categories tab. Use this window to define a name and description for each of the categories you need.



Category Options

Name	AL
Description	Alabama

In this example, create a state category for each state you need, then drag the state-specific forms into those categories on the forms list.

Importing Agency and Sub Agency Data

Business Scenario

You would like to import data onto policies specific to the agency, sub agency, or common for all agencies. Examples of this type of data include agency addresses, sub agency addresses, operator initials, counter-signed location, and so on.

Typically, this data is stored in the carrier's system. The issue is how to get this information out of the system without having to duplicate where this information is stored.

Solution

iPPS can use a URL-based service to import any global data into the system. iPPS will make a call to a particular URL defined in the global.xml file at the SYSTEM config level. Here is an example:

```
<DOIMPORTDATAURL>http://client.oracle.com/iPPS_Services/
cusimportdata.asp</DOIMPORTDATAURL>
```

When populating the response XML, remember the NAME attribute of the FIELD element *must* match the field name on the form or FAP file. Global data is only accepted at this time. The endpoint URL can expect to find the following data in the post:

Data	Description
authuser	The value for the current user.
key1	The value for the company.
key2	The value for the Line of Business.
keyid	The value for the policy number.
trancode	The value for the transaction code.
config	The value of the current configuration.
locid	The value for the agency code.
sublocid	The value for the agency sub code.
jurisdiction	The value for the tax state.

This represents a sample response from the service:

```
<?xml version="1.0" encoding="UTF-8"?>
<DOCUMENT>
    <DOCSET>
         <FIELD NAME="AGENCY NAME">Oracle Insurance Company</FIELD>
          <FIELD NAME="AGENCY ADDRESS 1">415 Main Street</FIELD>
          <FIELD NAME="AGENCY ADDRESS 2">Suite 200</FIELD>
          <FIELD NAME="AGENCY ADDRESS 3">Atlanta, GA 333330</FIELD>
         <FIELD NAME="COMPANY NAME">Amergen Insurance Agency</FIELD>
          <FIELD NAME="AGENT PHONE NBR #001">229-410-4100</FIELD>
          <FIELD NAME="AGENT PHONE NBR #002">230-410-4100</FIELD>
          <FIELD NAME="COUNTERSIGNED">Atlanta, GA</FIELD>
```

```
        <FIELD NAME="OPINITIALS">JS</FIELD>
    </DOCSET>
</DOCUMENT>
```

Configuring MSMQ

Business Scenario

You need to configure the system to use Microsoft Message Queue.

Solution

Refer to the Using WebSphere MQ topic in Chapter 2 of the Internet Document Server Guide for detailed instructions on using Microsoft Message Queue (MSMQ).

Placing the User/Profile database onto an SQL Server

Business Scenario

You need to put the User/Profile database onto an SQL Server.

Solution

First create the User/Profile tables in the ODBC compliant database. Refer to the Example UserDB SQL Script on page 65 for and example of how to create the tables

Once you have created the tables, modify the global.xml file to point to the new tables. Then add connection information to iUserDB to point to SQL:

```
<TABLE NAME="iUserDB">

<!--<DSN USERID="" PASSWORD="" CLASS="">driver={Microsoft Access
Driver
(*.mdb)};dbq=c:\inetpub\wwwroot\_iPPS312\user\USER.mdb;uid=Admin</
DSN>-->


            <DSN USERID="" PASSWORD=""
CLASS="">DSN=SQLDB_DEV;Uid=sa;Pwd=sqlexpress;</DSN>

</TABLE>
```

Restricting Access to Lines of Businesses

Business Scenario

You would like to restrict a line of business (or program) from certain agencies. Since you now have one FORM.DAT file, agency profiling becomes important in iPPS. You may want to deploy a specialty program to certain agencies and restrict other agencies from being able to choose that line of business.

Solution

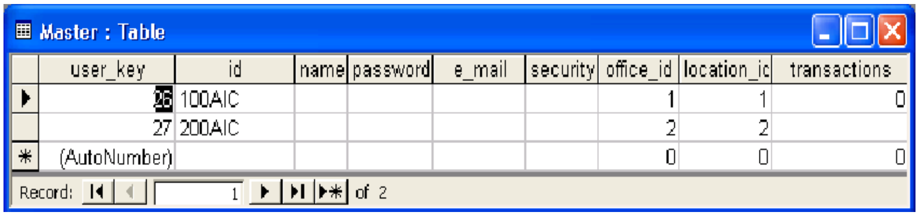
If the agency needs to have a subset of the company and line of business offerings, then you need to add the agency number to the user database.

- Set the user ID in the master table to be equal to the Config/Agency added to the GLOBAL.XML file. Here is an example:

  ```
  <CONFIG NAME="100AIC" LIBRARY="Oracle Underwriters (100)">
  ```

  User_key, ID, office_ID, and location_ID are required fields.

- Assign an office_ID in the master table that correlates to the office_ID in the lib_config table.



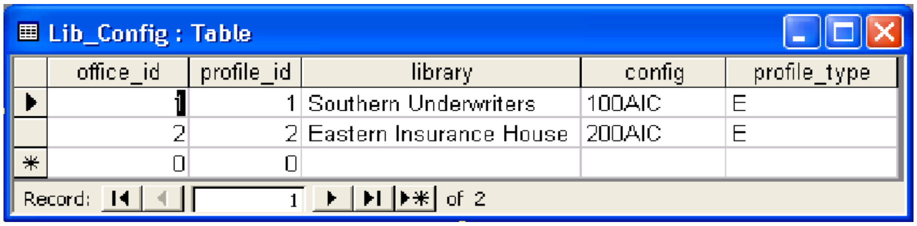
	user_key	id	name	password	e_mail	security	office_id	location_ic	transactions
▶	26	100AIC					1	1	0
	27	200AIC					2	2	
*	(AutoNumber)						0	0	0

- Assign a profile_id to the office_id in the lib_config table. Also enter the library and config values as they appear the global.xml for the LIBRARY and NAME attributes on the <CONFIG> node respectively. Here is an example:

  ```
  <CONFIG NAME="100AIC" LIBRARY="Oracle Underwriters (100)">
  ```

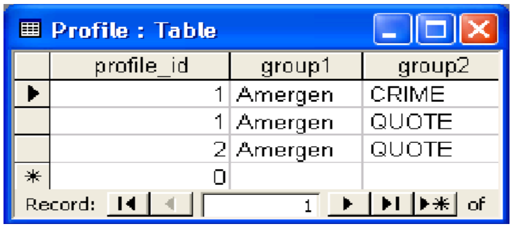
- Set the profile_type to **E** or **I** in lib_config table.

Enter	To give the agency access to
E	All company and line of business defined in the form.dat file except for the ones listed in the profile table for that profile_id.
I	Only the company and line of business listed in the profile table for that profile_id.



	office_id	profile_id	library	config	profile_type
▶	1	1	Southern Underwriters	100AIC	E
	2	2	Eastern Insurance House	200AIC	E
*	0	0			

- Set the Group1 (Company) and Group2 (Line of Business) as defined in the form.dat file according to the profile_type specified.



	profile_id	group1	group2
▶	1	Amergen	CRIME
	1	Amergen	QUOTE
	2	Amergen	QUOTE
*	0		

Adding Users to iPPS for Reports To Functionality

Business Problem

You would like to add users to the agency's user database, or you would like for your agency administrator to be able to add users.

Solution

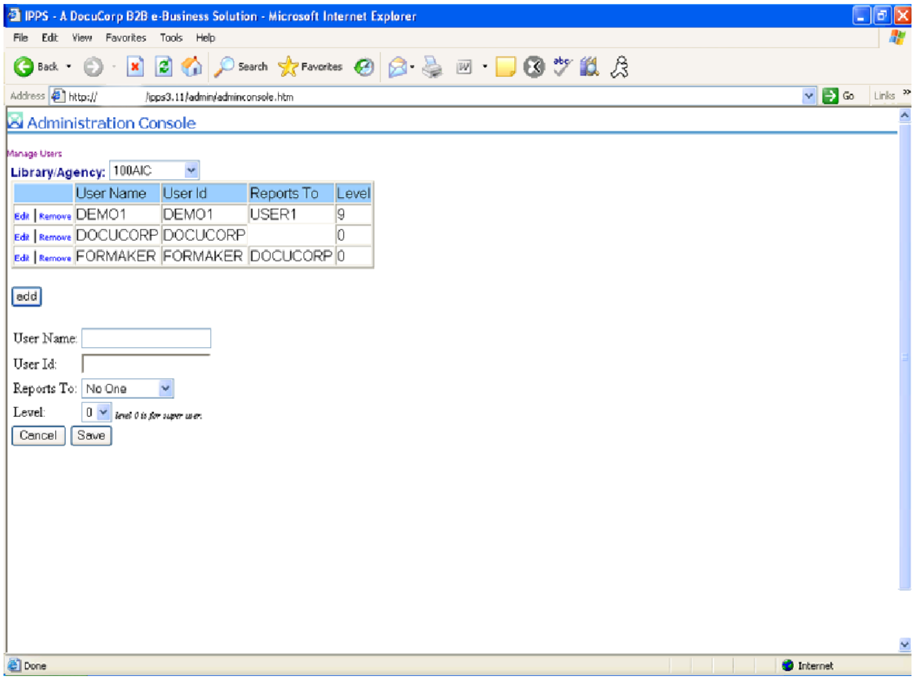
There are two approaches you can take:

iPPS has an admin console for adding users per Agency/Config. To access this console, got to a URL similar to this one:

```
http://myserver/iPPS311/admin/adminconsole.htm
```

Replace the *myserver* with the name of the server or IP address of the site hosting iPPS.

This simple access will allow for either the carrier or agency administrator to add, remove, and modify users in the userinfo database. iPPS will use the reports to structure the setup here when showing a user's WIP list.


IPPS - A DocuCorp B2B e-Business Solution - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http:// /ipps3.11/admin/adminconsole.htm
Go
Links
Done
Internet

Administration Console

Manage Users

Library/Agency: 100AIC

	User Name	User Id	Reports To	Level
Edit | Remove	DEMO1	DEMO1	USER1	9
Edit | Remove	DOCUCORP	DOCUCORP		0
Edit | Remove	FORMAKER	FORMAKER	DOCUCORP	0

add

User Name:

User Id:

Reports To: No One

Level: 0 level 0 is for super user.

Cancel Save

A second approach to adding, removing, modifying users is to use the RESTful services to perform the listed actions. This approach is helpful if you are already managing the MGA users at your agent portal. If this is the case, you can add the Reports To and Level parameters to your user management screen.

Whenever the logic on the portal site is used to add new users to their system, these RESTful requests can be run at the same time to place these new users into iPPS.

Adding users

Parameter	Description
LibraryId	The Agency/Config configured for the userinfo database.
UserId	The user ID you want to add.
Username	The name of the user being added.
Level	The user security level is a number between zero (0) and 9. Zero (0) is for superusers.
Reportsto	The user ID to whom this new user will report.
Action	"add"

Use this URL:

```
http://yourserver/iPPS311/doModifyUser.asp
```

Here is an example:

```
http://[yourserver]/iPPS311/
doModifyUser.asp?LibraryId=100AIC&UserId=DEMO1&username=DEMO1&level
=1&Reportsto=ORACLE&action=add
```

Deleting users

Parameter	Description:
LibraryId	The Agency/Config configured for the userinfo database.
UserId	The user ID you want to delete.
Action	"delete"

Use this URL:

```
http://yourserver/iPPS311/doModifyUser.asp
```

Here is an example

```
http://[yourserver]/iPPS311/
doModifyUser.asp?LibraryId=100AIC&UserId=DEMO1&action=delete
```

Updating users

Parameter	Description
LibraryId	The Agency/Config configured for the userinfo database.
UserId	The user ID you want to update.
Username	The user name to be updated.
Level	The user security level is a number between zero (0) and 9. Zero (0) is for superusers.
Reportsto	The user ID to whom this user will report.
Action	"update"

Use this URL

```
http://yourserver/iPPS311/doModifyUser.asp
```

Here is an example:

```
http://[yourserver]/iPPS311/
doModifyUser.asp?LibraryId=100AIC&UserId=DEMO1&username=DEMO1&level
=1&Reportsto=USER1&action=add
```

NOTE: The USERID value *cannot* be updated with this method. USERID is the unique identifier for the user. If you need to modify the USERID value, please delete the user first and re-add that person.

Publishing Quote Forms in Real-time

Business Scenario

You want to use Oracle Insurance to produce a quote form from your rating application. Some carriers may host an online rating application as an added benefit to their agents. Since an agent is providing enough data to get a quote, iPPS can propagate this data onto the quote form using Oracle Insurance's EWPS.

EWPS allows the quote to be produced in real-time by calling the doPublish web service and providing it with an import file which contains the specific form set information and data.

Solution

You will need to produce an Oracle Insurance standard XML file or an Oracle Insurance Standard Out object for use as the import file. You can also use the doGetTemplateListData web service to get a ComposeData object that can be populated with data.

NOTE: See Introduction to Enterprise Web Processing Services for more information.

Starting the Issue Process from a Rating Application

Business Scenario

Once the quote as been bound, an underwriter initiates the issuance process. Typically, the assigned underwriter produces the quote and will know when the time is right to issue the policy.

Solution

This is accomplished by using the doCreateFolder web service to place the documents needing to be issued into a folder for iPPS to process. The doCreateFolder web service can be called from a rating or other smart application data transformation tool that has the ability to map the needed forms for issuance into a Oracle Insurance standard XML file, a Standard Out object, or a ComposeData object.

The object or XML file is then submitted to the doCreateFolder web service which returns a FolderId. The parameters needed to create the folder are the Document Properties including Policy Number and Owner and others.

NOTE: See Introduction to Enterprise Web Processing Services for more information.

There are two ways to launch into iPPS to issue the policy.

If the person who starts the issuance process is not the one who will actually issue the policy, then use the Owner property of the doCreateFolder web service to define who will issue the policy. That person will log onto iPPS and view the work-in-process queue for transactions that need to be issued. The user can start working on an item and complete it within iPPS.

If the person starting the issuance process is the one who will complete the issue process, then they can be launched into iPPS immediately from the rating or smart application.

This can be accomplished by invoking the following URL with some standardized parameters:

```
http://[yourserver]/ipps/
doCompose_V2.asp?LibraryId=AIC&SourceType=FOLDERID&FolderId=25&Auth
User=ORACLE
```

Parameter	Description
LibraryId	AIC is the Agency/Configuration that contains the work-in-process queue.
SourceType	FOLDERID designates that you want to work on an item in the folder.
FolderId	25 is the unique identifier to the item that was returned from the doCreateFolder web service.
AuthUser	ORACLE is the user assigned to the folder item.

NOTE: A carrier may assign a block of policy numbers or develop an auto-policy number generating service so valid policy numbers can be provided to the rating application for delivery in the ImportFile for doCreateFolder.

Verifying Policy Numbers

Business Scenario

You would like to verify the policy numbers agents are entering into iPPS during the issue process. Enforcing valid policy numbers during the issue process saves time and effort when later uploading completed policies into imaging and coding systems. Some carriers may provide a block of numbers to each agency to be used. Others may require certain prefixes to be used.

Solution

iPPS lets you use a URL service to verify the KeyID/policy number entered into the system. iPPS will make a call to a particular URL defined in the global.xml at the SYSTEM config level. Here is an example:

```
<DOVERIFYKEYIDURL>http://client.oracle.com/iPPS_Services/
cusverifykeyid.asp</ DOVERIFYKEYIDURL >
```

When populating the response XML, remember that the DetailedMessage is a message displayed to the user if the result is Failure. The DetailedMessage element is not required if the result is Success. The endpoint URL can expect to find the following data in the post:

Data	Description
authuser	The value for the current user.
key1	The value for the Company.
key2	The value for the Line of Business.
keyid	The value for policy number.
trancode	The value for the transaction code.
config	The value of the current configuration.
locid	The value for the agency code.
sublocid	The value for the agency sub code.
jurisdiction	The value for the tax state.

This represents a sample response from this service:

```
<?xml version="1.0" encoding="UTF-8"?>
<doVerifyKeyIdResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="doVerifyKeyIdResponse.xsd"
schemaVersion="1.0">
        <Result>Success</Result>
        <DetailedMessage>Policy Number must start with AIC</
DetailedMessage>
</doVerifyKeyIdResponse>
```

This service can validate formats for the policy number and verify that the policy number entered is within a valid range for a particular agency or sub agency. This service will need to be written by someone in the carrier’s IT staff.

Applying Document Naming Conventions

Business Scenario

You would like to make sure the home office PDF and supporting XML/V2 output files are named consistently. This is especially important if the PDF and XML output files will be imported into an imaging system and the index information needs to be in the name of the files.

Solution

The print files produced are named by Docupresentment, not by iPPS. See the discussion of the DPRPint rule in Using the Documaker Bridge for more information.

In summary, you can add these options to the CONFIG.INI file for Docupresentment to specify a naming convention:

```
< PrtType:PDF >
   FileName = ~KEYID ~TRANCODE _ ~DALRUN time.dal .pdf
< PrtType:PCL >
   FileName = ~KEYID ~TRANCODE _ ~DALRUN time.dal .pcl
< PrtType:XML >
   FileName = ~KEYID ~TRANCODE _ ~DALRUN time.dal .xml
< PrtType:V2 >
   FileName = ~KEYID ~TRANCODE _ ~DALRUN time.dal .v2
```

NOTE: You must make sure enough information is provided to produce uniquely named print files. When specifying the names for PDF output, you may need to write a DAL script that will generate a unique string using the UniqueString DAL function. This may be required because users can produce multiple PDF documents and without a unique string, they will all be named the same.

Additionally, you can use the same conventions to produce uniquely name files during the COMPLETE process. After the distribution choices picked by the user are processed, iPPS will send one final request to Docupresentment to processing any final complete processing. This functionality is beneficial whenever you want to establish that an XML or V2 file is produced every time a policy is completed.

You can also set up the system to produce the Home Office copy as PDF every time a policy is completed. Both of the files can then be used by the carrier's system for updating their coding system or imaging system. See the discussion of automatically printing upon completion in Using the Documaker Bridge for more information

Here is an example setup for the complete process from [config.ini]:

```
;Setup for complete print. This is to designate the Home Office PDF
and auto export
< Complete >
   CompleteType = COMP1
   CompleteType = COMP2
< Complete:COMP1 >
   FileType = PDF
   FileName ~KEYID ~TRANCODE _ ~DALRUN time.dal.pdf
   FileExt = pdf
   FilePath = d:\AIC\complete\
   Recipient = HOME OFFICE
```

```
< Complete:COMP2 >
   FileType = XML
   FileName = ~KEYID ~TRANCODE _ ~DALRUN time.dal.xml
   FileExt = xml
   FilePath = d:\AIC\complete\
   Recipient = ALLRECIPIENTS
```

NOTE: If the FilePath location is set to the same location used by an existing upload process the agents use, then the upload process may occur real-time when the user completes their policy. This eliminates the need for the agent to upload the Home Office PDF, and XML/V2 has they do in PPS.

Providing Default Distribution Options to Agencies

Business Scenario

You would like to create a more customized user experience for each of your agencies. In PPS and Documaker Workstation, each agency could set up their MDR output however they wanted. In iPPS, the carrier will have to set up, or have the agency send their setup, to allow for different default distribution options.

Solution

iPPS can handle the distribution/processing options needed for the business during the complete process. iPPS can optionally use a DistributionOptions.xml file for setting the default choices per configuration.

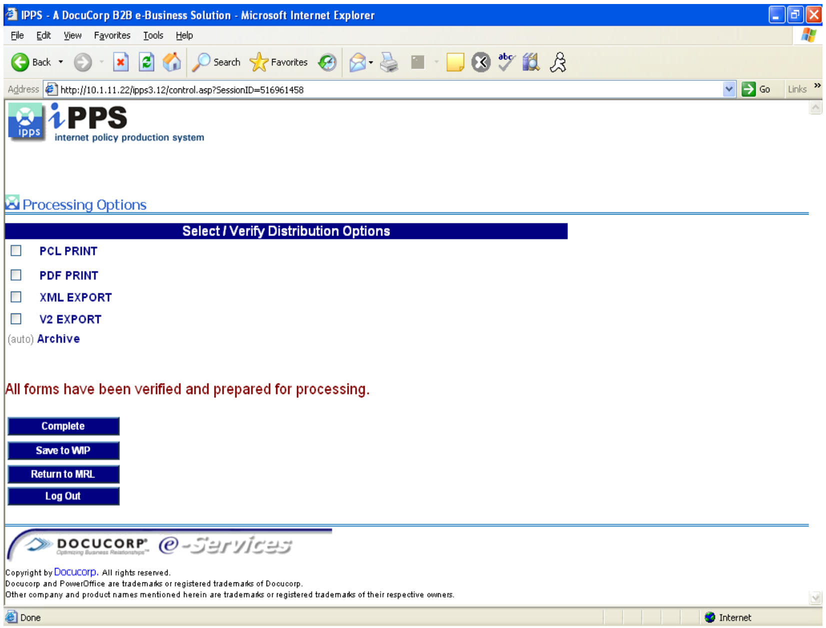
The system looks for this file in the path specified on the DEFAULTSRC attribute of the <PROCESSINGOPTIONS> node in the global.xml at each config level. To guarantee accurate permissions to load this file, the identity specified in Component Services or the Web Application Pool (if a library application), will need read access to this location. If this XML file is not present, no selections are defaulted.

iPPS assumes that immediate and batch print are for PCL print only. This allows for dynamic addition of recipients for PDF production. You specify the distribution options available in the system using the GLOBAL.XML file. An example is shown below.

The DESC attribute is the display value next to the checkbox. The NAME attribute determines what type of print is available. The value of INTERNETPROOF needs to remain but is ignored as a distribution option.

```
<PROCESSINGOPTIONS>
<IROUTE NAME=”PCLPRINT” DESC=”PCL PRINT” PROCESS=”FALSE” RESULTS=”” 
RECIPIENT=”ALL” EXPORTYPE=”2170”>
    </IROUTE>
    <IROUTE NAME=”PDFPRINT” DESC=”PDF PRINT” PROCESS=”FALSE” 
RESULTS=”” RECIPIENT=”ALL” EXPORTYPE=”2180”>
    </IROUTE>
    <IROUTE NAME=”XMLEXPORT” DESC=”XML EXPORT” PROCESS=”FALSE” 
RESULTS=”” RECIPIENT=”ALL” EXPORTYPE=”2195”>
    </IROUTE>
    <IROUTE NAME=”V2EXPORT” DESC=”V2 EXPORT” PROCESS=”FALSE” 
RESULTS=”” EXPORTYPE=”2190”>
    </IROUTE>
</PROCESSINGOPTIONS>
```

Based on these settings the user will be presented with a page similar to this one:


IPPS - A DocuCorp B2B e-Business Solution - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address http://10.1.11.22/ipps3.12/control.asp?SessionID=516961458
Go
Links
iPPS
internet policy production system
Complete
Save to WIP
Return to MRL
Log Out
DOCUCORP Optimizing Business Relationships e-Services
Done
Internet

Processing Options

Select / Verify Distribution Options

☐ PCL PRINT

☐ PDF PRINT

☐ XML EXPORT

☐ V2 EXPORT

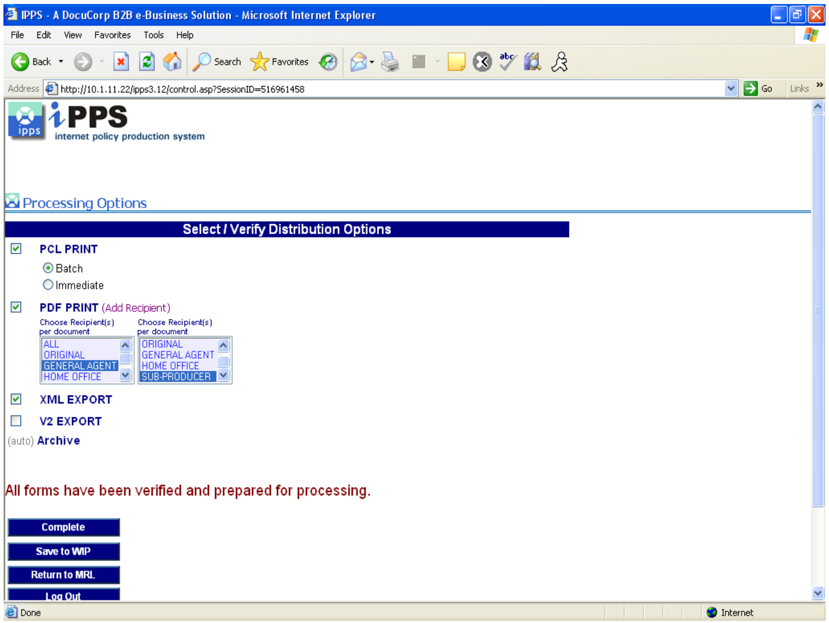
(auto) Archive

All forms have been verified and prepared for processing.

Copyright by Docucorp. All rights reserved.
Docucorp and PowerOffice are trademarks or registered trademarks of Docucorp.
Other company and product names mentioned herein are trademarks or registered trademarks of their respective owners.

If the system finds the DistributionOptions.XML file, it applies some default selections. Here is an example:

```
<?xml version="1.0" encoding="UTF-8"?>
<DistributionOptions>
    <Channel>
        <PublishType>PDFPRINT</PublishType>
        <DistributionType>IMMEDIATE</DistributionType>
        <Recipient name="GENERAL AGENT"/>
    </Channel>
    <Channel>
        <PublishType>PDFPRINT</PublishType>
        <DistributionType>IMMEDIATE</DistributionType>
        <Recipient name="SUB-PRODUCER"/>
    </Channel>
    <Channel>
        <PublishType>PCLPRINT</PublishType>
        <DistributionType>SCHEDULED</DistributionType>
        <Recipient name="ALLRECIPIENTS"/>
    </Channel>
    <Channel>
        <PublishType>XMLEXPORT</PublishType>
        <DistributionType>IMMEDIATE</DistributionType>
        <Recipient name="ALLRECIPIENTS"/>
    </Channel>
</DistributionOptions>
```



IPPS - A DocuCorp B2B e-Business Solution - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
iPPS
internet policy production system
Done
Internet

http://10.1.11.22/ipps3.12/control.asp?SessionID=516961458

Processing Options

Select / Verify Distribution Options

☑ PCL PRINT

● Batch

○ Immediate

☑ PDF PRINT (Add Recipient)

Choose Recipient(s) per document: ALL, ORIGINAL, GENERAL AGENT (selected), HOME OFFICE

Choose Recipient(s) per document: ORIGINAL, GENERAL AGENT, HOME OFFICE, SUB-PRODUCER (selected)

☑ XML EXPORT

☐ V2 EXPORT

(auto) Archive

All forms have been verified and prepared for processing.

Inserting State Stamps, Signatures, and Logos

Business Scenario

You would like to have the required state stamp to appear on the Dec Page without have to write additional DAL scripts with table lookups. You may also want to apply a signature to a form (Dec Page) based on the Company, Agency, Sub Agency, and so on.

Solution

For more information on inserting state stamps and signatures, refer to the Documaker Workstation Administration Guide. Docupresentment will try to apply this functionality by default, and therefore make it available to iPPS if the FAP files are set up correctly.

Importing Archive Data into a New Policy

Business Scenario

You would like use archived data to issue a renewal policy. You want the required forms to be pre-selected and use the data from a previously archived transaction.

Solution

This feature mimics the functionality of the TRNImpDatFromArchive feature in PPS with a few differences. The Documaker Workstation Administration Guide states this is the setting you need to import from archive:

```
< ImportFormats >
    02 =;IA;Import From Archive;TRNW32->TRNImpDatFromArchive;
```

For iPPS, this is the global.xml setting:

```
<PPS>
    <IMPORTFROMARCHIVE>
        <DATAONLY>TRUE</DATAONLY>
    </IMPORTFROMARCHIVE>
</PPS>
```

This information can be set at the global(system) config or each specific agency/config level.

iPPS will import the form set from the archived transaction. Once on the Forms Selection window, the pre-selected forms will be the required forms for the current form.dat file. If the user picks forms that existed in the archived form set, the system tries to match the form- and section-level data.

This way you could potentially get more than just global data. If you want the form set and archive index values to come across as the SelectForms and TransferRecord options in PPS do, then do not set the DATAONLY setting as specified above. The default archive retrieval of iPPS will select the forms and transfer the archive record by default. This lets the archived form set's data be used with a new transaction.

Appendix A

Example DFD Files and SQL Scripts

This appendix lists example DFD and SQL files referenced in the Business Scenarios in Chapter 1. Included are the following:

- Example APPIDX.DFD File on page 40
- Example ARCHIVE.DFD File on page 46
- Example WIP.DFD File on page 48
- Example WIPDATA.DFD File on page 53
- Example APPIDX SQL Script on page 55
- Example ARCHIVE SQL Script on page 56
- Example CATALOG SQL Script on page 57
- Example QA_MRL SQL Script on page 58
- Example WIP SQL Script on page 60
- Example WIP_Archive_Combined SQL Script on page 61
- Example WIPDATA SQL Script on page 64
- Example UserDB SQL Script on page 65

Example APPIDX.DFD File

```
; This is DFD file for application archive index file
; same as the standard WIP file.
[FIELDS]
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
FIELDNAME=RUNDATE
FIELDNAME=INVFLAG
FIELDNAME=CLAIMFL
FIELDNAME=TRNNAME

[FIELD:KEY1]
EXT_TYPE       = CHAR_ARRAY
EXT_LENGTH     = 30
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     30
INT_PRECISION=  0
KEY=Y
REQUIRED=Y

[FIELD:KEY2]
EXT_TYPE       = CHAR_ARRAY
EXT_LENGTH     = 30
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     30
INT_PRECISION=  0
KEY=Y
REQUIRED=Y

[FIELD:KEYID]
EXT_TYPE       = CHAR_ARRAY
EXT_LENGTH     = 20
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     20
INT_PRECISION=  0
```

```
KEY=Y
REQUIRED=Y

[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=Y

[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=Y

[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=Y

[FIELD:CURRUSER]
EXT_TYPE      = CHAR_ARRAY
EXT_LENGTH    = 8
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     8
INT_PRECISION=  0
KEY=Y
REQUIRED=Y

[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=Y

[FIELD:FORMSETID]
EXT_TYPE      = CHAR_ARRAY
```

```
EXT_LENGTH    = 8
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     8
INT_PRECISION=  0
KEY=N
REQUIRED=Y

[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=Y

[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=Y

[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=Y

[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=Y

[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=Y

[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=Y

[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=Y

[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=Y

[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:APPDATA]
EXT_TYPE      = CHAR_ARRAY
EXT_LENGTH    = 150
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     150
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=Y

[FIELD:RUNDATE]
EXT_TYPE      = NOT_PRESENT
EXT_LENGTH    = 0
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     8
INT_PRECISION=  0
KEY=N
REQUIRED=Y

[FIELD:INVFLAG]
EXT_TYPE      = NOT_PRESENT
EXT_LENGTH    = 0
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     1
INT_PRECISION=  0
KEY=N
REQUIRED=N

[FIELD:CLAIMFL]
EXT_TYPE      = NOT_PRESENT
EXT_LENGTH    = 0
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     1
INT_PRECISION=  0
KEY=N
REQUIRED=N

[FIELD:TRNNAME]
EXT_TYPE      = CHAR_ARRAY
EXT_LENGTH    = 62
EXT_PRECISION=  0
INT_TYPE=       CHAR_ARRAY
INT_LENGTH=     62
INT_PRECISION=  0
KEY=N
REQUIRED=N

[KEYS]
KEYNAME=KEY1
KEYNAME=KEY2
KEYNAME=KEYID
KEYNAME=NAMETAG

[KEY:KEY1]
```

```
EXPRESSION=KEY1
FIELDLIST=KEY1

[KEY:KEY2]
EXPRESSION=KEY2
FIELDLIST=KEY2

[KEY:KEYID]
EXPRESSION=KEYID
FIELDLIST=KEYID

[KEY:NAMETAG]
EXPRESSION=TRNNAME
FIELDLIST=TRNNAME
```

Example ARCHIVE.DFD File

```
< FIELDS >
   FIELDNAME = ARCKEY
   FIELDNAME = SEQ_NUM
   FIELDNAME = CONT_FLAG
   FIELDNAME = TOTAL_SIZE
   FIELDNAME = CARDATA

< FIELD:ARCKEY >
   INT_TYPE = CHAR_ARRAY
   INT_LENGTH = 18
   EXT_TYPE = CHAR_ARRAY
   EXT_LENGTH = 18
   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 = N
   REQUIRED = N

< FIELD:CARDATA >
   INT_TYPE = BLOB
   INT_LENGTH = 252
   EXT_TYPE = BLOB
   EXT_LENGTH = 252
   KEY = N
   REQUIRED = N

< KEYS >
  KEYNAME = ARCKEY
  KEYNAME = SEQ_NUM
  KEYNAME = CAR_KEY

< KEY:ARCKEY >
  EXPRESSION = ARCKEY+SEQ_NUM
  FIELDLIST = ARCKEY,SEQ_NUM
```

```
< KEY:SEQ_NUM >
  EXPRESSION = SEQ_NUM
  FIELDLIST = SEQ_NUM

< KEY:CAR_KEY >
  EXPRESSION = ARCKEY
  FIELDLIST = ARCKEY
```

EXAMPLE WIP.DFD FILE

```
< FIELDS >
    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   = GUIDKEY
    FIELDNAME   = TRNNAME
    FIELDNAME   = LOCID
    FIELDNAME   = SUBLOCID
    FIELDNAME   = JURISDICTN
    FIELDNAME   = QUEUID
    FIELDNAME   = RECNUM
< FIELD:APPDATA >
    EXT_LENGTH = 150
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 150
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:ARCKEY >
    EXT_LENGTH = 18
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 18
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:CREATETIME >
    EXT_LENGTH = 8
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 8
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:CURRUSER >
    EXT_LENGTH = 64
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 64
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:DESC >
    EXT_LENGTH = 30
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 30
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:FORMSETID >
    EXT_LENGTH = 32
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 32
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:FROMTIME >
    EXT_LENGTH = 8
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 8
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:FROMUSER >
    EXT_LENGTH = 64
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 64
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:GUIDKEY >
    EXT_LENGTH = 32
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 32
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:INUSE >
    EXT_LENGTH = 1
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 1
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:JURISDICTN >
    EXT_LENGTH = 30
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 30
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:UNIQUE_ID >
    INT_TYPE = CHAR_ARRAY
    INT_LENGTH = 32
    EXT_TYPE = CHAR_ARRAY
    EXT_LENGTH = 32
    KEY = N
    REQUIRED = N
< FIELD:KEY1 >
    EXT_LENGTH = 62
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 62
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:KEY2 >
    EXT_LENGTH = 62
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 62
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:KEYID >
    EXT_LENGTH = 20
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 20
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:LOCID >
    EXT_LENGTH = 30
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 30
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:MODIFYTIME >
    EXT_LENGTH = 8
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 8
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:ORIGUSER >
    EXT_LENGTH = 64
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 64
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:RECNUM >
    EXT_LENGTH = 0
    EXT_TYPE   = NOT_PRESENT
    INT_LENGTH = 4
    INT_TYPE   = LONG
    KEY        = N
    REQUIRED   = N
< FIELD:RECTYPE >
    EXT_LENGTH = 3
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 3
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:STATUSCODE >
    EXT_LENGTH = 2
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 2
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:SUBLOCID >
    EXT_LENGTH = 30
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 30
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:TOTIME >
    EXT_LENGTH = 8
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 8
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:TOUSER >
    EXT_LENGTH = 64
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 64
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:TRANCODE >
    EXT_LENGTH = 2
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 2
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:TRNNAME >
    EXT_LENGTH = 62
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 62
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N
< FIELD:QUEUEID >
    EXT_LENGTH = 32
    EXT_TYPE   = CHAR_ARRAY
    INT_LENGTH = 32
    INT_TYPE   = CHAR_ARRAY
    KEY        = N
    REQUIRED   = N

< KEY:UNIQUE_ID >
    EXPRESSION = UNIQUE_ID
    FIELDLIST = UNIQUE_ID
< KEY:DOCTAG >
    DESCENDING = 0
    EXPRESSION = KEY1+KEY2+KEYID+RECTYPE
    FIELDLIST  = KEY1,KEY2,KEYID,RECTYPE
    FILTER     =
    UNIQUE     = 0
< KEY:KEY2TAG >
    DESCENDING = 0
    EXPRESSION = KEY2
    FIELDLIST  = KEY2
    FILTER     =
    UNIQUE     = 0
< KEY:KEYIDTAG >
    DESCENDING = 0
    EXPRESSION = KEYID
    FIELDLIST  = KEYID
    FILTER     =
    UNIQUE     = 0
< KEY:USERTAG >
    DESCENDING = 0
    EXPRESSION = CURRUSER
    FIELDLIST  = CURRUSER
    FILTER     =
    UNIQUE     = 0
< KEYS >
    KEYNAME    = UNIQUE_ID
    KEYNAME    = DOCTAG
    KEYNAME    = KEY2TAG
    KEYNAME    = KEYIDTAG
    KEYNAME    = USERTAG
```

Example WIPDATA.DFD File

```
< FIELDS >
FIELDNAME = FORMSETID
FIELDNAME = SEQ_NUM
FIELDNAME = CONT_FLAG
FIELDNAME = TOTAL_SIZE
FIELDNAME = CARDATA

< FIELD:FORMSETID >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 32
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 32
KEY = Yes
REQUIRED = Yes

< FIELD:SEQ_NUM >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 5
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 5
KEY = Yes
REQUIRED = Yes

< FIELD:CONT_FLAG >
INT_TYPE = CHAR_ARRAY
INT_LENGTH = 1
EXT_TYPE = CHAR_ARRAY
EXT_LENGTH = 1
KEY = Yes
REQUIRED = Yes

< FIELD:TOTAL_SIZE >
INT_Type = LONG
INT_Length = 4
EXT_Type = LONG
EXT_Length = 4
Key = No
Required = No

< FIELD:CARDATA >
INT_Type = BLOB
INT_Length = 252
EXT_Type = BLOB
EXT_Length = 252
Key = No
Required = No

< Keys >
KeyName = FORMSETID
KeyName = FORMSETIDSEQ
KeyName = SEQ_NUM

< KEY:FORMSETID >
Expression = FORMSETID
FieldList = FORMSETID
```

```
< KEY:FORMSETIDSEQ >
Expression = FORMSETID+SEQ_NUM
FieldList = FORMSETID,SEQ_NUM

< KEY:SEQ_NUM >
Expression = SEQ_NUM
FieldList = SEQ_NUM
```

EXAMPLE APPIDX SQL SCRIPT

Here is an example of the APPIDX.SQL script:

```
USE [master]
GO
/****** Object:  Table [dbo].[OW0006_APPIDX]    Script Date: 10/22/2009
12:31:02 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_APPIDX](
    [KEY1] [char](30) NOT NULL,
    [KEY2] [char](30) NOT NULL,
    [KEYID] [char](20) NOT NULL,
    [RECTYPE] [char](3) NOT NULL,
    [CREATETIME] [char](8) NOT NULL,
    [ORIGUSER] [char](8) NOT NULL,
    [CURRUSER] [char](8) NOT NULL,
    [MODIFYTIME] [char](8) NOT NULL,
    [FORMSETID] [char](8) NOT NULL,
    [TRANCODE] [char](2) NOT NULL,
    [STATUSCODE] [char](2) NOT NULL,
    [FROMUSER] [char](8) NOT NULL,
    [FROMTIME] [char](8) NOT NULL,
    [TOUSER] [char](8) NOT NULL,
    [TOTIME] [char](8) NOT NULL,
    [DESC] [char](30) NOT NULL,
    [INUSE] [char](1) NOT NULL,
    [ARCKEY] [char](18) NULL,
    [APPDATA] [char](150) NULL,
    [TRNNAME] [char](62) NULL
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Example ARCHIVE SQL Script

Here is an example of the ARCHIVE.SQL script:

```
CREATE TABLE [OW0006_ARCHIVE] (
    [ARCKEY] [char] (18) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [SEQ_NUM] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [CONT_FLAG] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [TOTAL_SIZE] [int] NULL,
    [CARDATA] [image] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
CREATE INDEX [OW0006_ARCHIVEARCKEY] ON [OW0006_ARCHIVE]([ARCKEY], [SEQ_NUM])
ON [PRIMARY]
GO
CREATE INDEX [OW0006_ARCHIVESEQ_NUM] ON [OW0006_ARCHIVE]([SEQ_NUM]) ON
[PRIMARY]
GO
CREATE INDEX [OW0006_ARCHIVECAR_KEY] ON [OW0006_ARCHIVE]([ARCKEY]) ON
[PRIMARY]
GO
```

Example CATALOG SQL Script

Here is an example of the CATALOG.SQL script:

```
USE [master]
GO
/****** Object:  Table [dbo].[OW0006_CATALOG]    Script Date: 10/22/2009
12:32:08 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_CATALOG](
    [CATALOGID] [char](10) NULL,
    [CARFILE] [char](8) NULL,
    [MEDIAID] [char](11) NULL,
    [STATUS] [char](1) NULL
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Example QA_MRL SQL Script

Here is an example of the QA_MRL.SQL script:

```
CREATE TABLE [WW_PROD] (
    [FILETYPE] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [FILESTYP] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [FILENAME] [char] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [RESOURCE] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [DESCRIPT] [char] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [EFFECTIV] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [MODIFYTM] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [FILEINDX] [char] (8) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [RECSTAT] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [VERSION] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [REVISION] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [USERID] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [USRLEVL] [char] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [PASSWD] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [UNIQUE_ID] [char] (26) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [ARCKEY] [char] (18) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [MODE] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [STATUS] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [CLASS] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [PROJECT] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODFILEINDX] ON [WW_PROD]([FILETYPE], [FILESTYP],
[FILENAME], [VERSION], [REVISION]) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODUNIQUE_ID] ON [WW_PROD]([UNIQUE_ID]) ON [PRIMARY]
GO
CREATE TABLE [WW_PRODD] (
    [ARCKEY] [char] (18) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [SEQ_NUM] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [CONT_FLAG] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [TOTAL_SIZE] [int] NULL,
    [CARDATA] [image] NULL
) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODDARCKEY] ON [WW_PRODD]([ARCKEY], [SEQ_NUM]) ON
[PRIMARY]
GO
CREATE INDEX [WW_PRODDSEQ_NUM] ON [WW_PRODD]([SEQ_NUM]) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODDCAR_KEY] ON [WW_PRODD]([ARCKEY]) ON [PRIMARY]
GO
CREATE TABLE [WW_PRODC] (
    [CATALOGID] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [CARFILE] [char] (8) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [MEDIAID] [char] (11) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [LASTNUM] [char] (8) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [STATUS] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODCCATALOGKEY] ON [WW_PRODC]([CATALOGID]) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODCCARFILEKEY] ON [WW_PRODC]([CARFILE]) ON [PRIMARY]
```

```
GO
CREATE INDEX [WW_PRODCLASTNUMKEY] ON [WW_PRODC]([LASTNUM]) ON [PRIMARY]
GO
CREATE TABLE [WW_PRODL] (
    [DATE] [char] (8) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [TIME] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [LIBNAME] [char] (129) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [ACTION] [char] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [FILENAME] [char] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [FILETYPE] [char] (3) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [VERSION] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [REVISION] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [EFFECTIV] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [MODE] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [STATUS] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [CLASS] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [PROJECT] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [USERID] [char] (64) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [PROCESS] [char] (20) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODLLOGTAG] ON [WW_PRODL]([DATE], [TIME]) ON [PRIMARY]
GO
CREATE INDEX [WW_PRODLUNIQTAG] ON [WW_PRODL]([DATE], [TIME], [LIBNAME],
[ACTION], [FILENAME], [FILETYPE], [VERSION], [REVISION], [EFFECTIV]) ON
[PRIMARY]
GO
```

Example WIP SQL Script

Here is an example of the WIP.SQL script:

```
USE [master]
GO
/****** Object:  Table [dbo].[OW0006_WIP]    Script Date: 10/22/2009 12:33:25
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_WIP](
    [UNIQUE_ID] [char](32) NULL,
    [KEY1] [char](62) NULL,
    [KEY2] [char](62) NULL,
    [KEYID] [char](20) NULL,
    [RECTYPE] [char](3) NULL,
    [CREATETIME] [char](8) NULL,
    [ORIGUSER] [char](64) NULL,
    [CURRUSER] [char](64) NULL,
    [MODIFYTIME] [char](8) NULL,
    [FORMSETID] [char](32) NULL,
    [TRANCODE] [char](2) NULL,
    [STATUSCODE] [char](2) NULL,
    [FROMUSER] [char](64) NULL,
    [FROMTIME] [char](8) NULL,
    [TOUSER] [char](64) NULL,
    [TOTIME] [char](8) NULL,
    [DESC] [char](30) NULL,
    [INUSE] [char](1) NULL,
    [ARCKEY] [char](18) NULL,
    [APPDATA] [char](150) NULL,
    [GUIDKEY] [char](32) NULL,
    [TRNNAME] [char](62) NULL,
    [LOCID] [char](30) NULL,
    [SUBLOCID] [char](30) NULL,
    [JURISDICTN] [char](30) NULL
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Example WIP_Archive_Combined SQL Script

Here is an example of the WIP_ARCHIVED_COMBINED.SQL script:

Note: This script is a combination of the Example WIP SQL Script on page 60 and Example ARCHIVE SQL Script on page 56. Combining these scripts into a single script can make it easier to do updates.

```
USE [master]
GO
/****** Object:  Table [dbo].[OW0006_APPIDX]    Script Date: 10/22/2009
12:31:02 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_APPIDX](
    [KEY1] [char](30) NOT NULL,
    [KEY2] [char](30) NOT NULL,
    [KEYID] [char](20) NOT NULL,
    [RECTYPE] [char](3) NOT NULL,
    [CREATETIME] [char](8) NOT NULL,
    [ORIGUSER] [char](8) NOT NULL,
    [CURRUSER] [char](8) NOT NULL,
    [MODIFYTIME] [char](8) NOT NULL,
    [FORMSETID] [char](8) NOT NULL,
    [TRANCODE] [char](2) NOT NULL,
    [STATUSCODE] [char](2) NOT NULL,
    [FROMUSER] [char](8) NOT NULL,
    [FROMTIME] [char](8) NOT NULL,
    [TOUSER] [char](8) NOT NULL,
    [TOTIME] [char](8) NOT NULL,
    [DESC] [char](30) NOT NULL,
    [INUSE] [char](1) NOT NULL,
    [ARCKEY] [char](18) NULL,
    [APPDATA] [char](150) NULL,
    [TRNNAME] [char](62) NULL
) ON [PRIMARY]

GO

CREATE TABLE [OW0006_ARCHIVE] (
    [ARCKEY] [char] (18) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [SEQ_NUM] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [CONT_FLAG] [char] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
    [TOTAL_SIZE] [int] NULL,
    [CARDATA] [image] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
CREATE INDEX [OW0006_ARCHIVEARCKEY] ON [OW0006_ARCHIVE]([ARCKEY], [SEQ_NUM])
ON [PRIMARY]
GO
CREATE INDEX [OW0006_ARCHIVESEQ_NUM] ON [OW0006_ARCHIVE]([SEQ_NUM]) ON
[PRIMARY]
```

```
GO
CREATE INDEX [OW0006_ARCHIVECAR_KEY] ON [OW0006_ARCHIVE]([ARCKEY]) ON
[PRIMARY]
GO

/****** Object:  Table [dbo].[OW0006_CATALOG]    Script Date: 10/22/2009
12:32:08 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_CATALOG](
    [CATALOGID] [char](10) NULL,
    [CARFILE] [char](8) NULL,
    [MEDIAID] [char](11) NULL,
    [STATUS] [char](1) NULL
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[OW0006_WIP]   Script Date: 10/22/2009 12:33:25
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_WIP](
    [UNIQUE_ID] [char](32) NULL,
    [KEY1] [char](62) NULL,
    [KEY2] [char](62) NULL,
    [KEYID] [char](20) NULL,
    [RECTYPE] [char](3) NULL,
    [CREATETIME] [char](8) NULL,
    [ORIGUSER] [char](64) NULL,
    [CURRUSER] [char](64) NULL,
    [MODIFYTIME] [char](8) NULL,
    [FORMSETID] [char](32) NULL,
    [TRANCODE] [char](2) NULL,
    [STATUSCODE] [char](2) NULL,
    [FROMUSER] [char](64) NULL,
    [FROMTIME] [char](8) NULL,
    [TOUSER] [char](64) NULL,
    [TOTIME] [char](8) NULL,
    [DESC] [char](30) NULL,
    [INUSE] [char](1) NULL,
    [ARCKEY] [char](18) NULL,
    [APPDATA] [char](150) NULL,
    [GUIDKEY] [char](32) NULL,
    [TRNNAME] [char](62) NULL,
    [LOCID] [char](30) NULL,
    [SUBLOCID] [char](30) NULL,
    [JURISDICTN] [char](30) NULL
) ON [PRIMARY]
```

```
GO
/****** Object:  Table [dbo].[OW0006_WIPDATA]    Script Date: 10/22/2009
12:33:37 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_WIPDATA](
    [FORMSETID] [char](32) NOT NULL,
    [SEQ_NUM] [char](5) NOT NULL,
    [CONT_FLAG] [char](1) NOT NULL,
    [TOTAL_SIZE] [int] NULL,
    [CARDATA] [image] NULL
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Example WIPDATA SQL Script

Here is an example WIPDATA.SQL script:

```
USE [master]
GO
/****** Object:  Table [dbo].[OW0006_WIPDATA]    Script Date: 10/22/2009
12:33:37 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OW0006_WIPDATA](
    [FORMSETID] [char](32) NOT NULL,
    [SEQ_NUM] [char](5) NOT NULL,
    [CONT_FLAG] [char](1) NOT NULL,
    [TOTAL_SIZE] [int] NULL,
    [CARDATA] [image] NULL
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

Example UserDB SQL Script

```
CREATE TABLE iAuto (user_key int NULL,
                 CARRIER varchar(50) NULL,
                 COMPANY varchar(50) NULL,
                 DATE1 varchar(100) NULL,
                 LOB varchar(50) NULL,
                 POLICY_NBR varchar(50) NULL)

CREATE TABLE iDefault (user_key int NULL)

CREATE TABLE iSession (session_id varchar(50) NULL,
                       state_id varbinary(MAX) NULL)

CREATE TABLE iSystem (user_key int NULL,
          Endorse_Code varchar(2) NULL,
          Renew_Code varchar(2) NULL)

CREATE TABLE Lib_Config (office_id int NULL,
                         profile_id int NULL,
                         library varchar(50) NULL,
                         config varchar(50) NULL,
                         profile_type varchar(1) NULL)

CREATE TABLE Location (location_id int NULL, location_name varchar(50) NULL)

CREATE TABLE Master (user_key int NOT NULL,
                     id varchar(20) NULL,
                     name varchar(50) NULL,
                     password varchar(8) NULL,
                     e_mail varchar(50) NULL,
                     security varchar(50) NULL,
                     office_id int NULL,
                     location_id int NULL,
                     transactions int NULL)

CREATE TABLE Office (office_id int NULL, office_name varchar(50) NULL)

CREATE TABLE Profile (profile_id int NULL, group1 varchar(50) NULL,
group2 varchar(50) NULL)
```