OnDemand Newsletter

News and Tips about IBM Content Manager OnDemand

4th Quarter 2012

In This Issue

News

About this Newsletter 1
OnDemand Version 9.0 Available 1
ECM Forum at IBM IOD 2
ODUG Conference 2

Tips – Cross Platforms

Using the Report Wizard to Create User Defined PDF 3
Updating Indexes from the OnDemand Windows Client 4
Configuring ERM and CFS-CMOD Using the Report Wizard 5

Tips – Multiplatforms

PDF Indexer Float Trigger Support in V9 ... 6
Configuring Multiple TSM Servers 11

Tips – z/OS

OnDemand for z/OS Quick Hits 14
OS/390 Indexer Usage Notes: The INDEXSTYLE Parameter - Part 4 of 4 15

Tips – IBM i

OnDemand for i Quick Hits 21
Enhanced Spooled File Selection Using the 9.0 Administrator Client 22
Enhanced Spooled File Selection at V7.1 . 23
Graphical Indexer Support for the Find Once Option 24
Graphical Indexer Support for the Use Load Default or Propagation Option 25

Additional Information

News

About this Newsletter

This newsletter is designed to keep you better informed about IBM® Content Manager OnDemand on all platforms. The newsletter will be published quarterly.

Previous editions of this newsletter can be found on the OnDemand support web sites under the 'Featured links' heading. They are also available on the OnDemand Users Group web site under the heading 'Presentations, Newsletters, and such'.

Correspondence related to this newsletter should be directed to odnews@us.ibm.com.

OnDemand Version 9.0 Available

OnDemand version 9.0 is now available for Multiplatforms and z/OS®. Version 9.0 includes these enhancements and more:

- Full text search - An optional chargeable feature that enables content that is not easily identified by regular indices to be easily searched and retrieved.
- The Windows™ client can now update metadata fields.
- Exporting of search hit lists and documents directly to .csv file format
- OS/390® indexer now supported on AIX®
- Database timestamp support eliminates internal OnDemand date limitations
- Documents can now be indexed with up to 128 metadata fields

- Graphical annotation support in the ODWEK line data viewer applet
- Support for IBM Content Navigator
- Integration with IBM System Dashboard for Enterprise Content Management
- Integration with IBM FileNet System Monitor (separately available product)
- Compatibility with Content Management Interoperability Services (CMIS)

Enhancements to OnDemand for z/OS version 9.0 include those listed above plus the following:

- New display commands
  - F ARSSOCKD,D,STATUS - Displays information similar to arssockd -p
  - F ARSSOCKD,D,CONFIG - Displays the resolved configuration from the ars.ini and ars.cfg files
  - F ARSSOCKD,D,ENVAR - Displays the environment variables
  - F ARSSOCKD,D,ICSF - Displays the state of ICSF relative to the server
- Password phrase support for passwords longer than 8 characters
- ARSVIEW enhancements
- Shutdown and startup enhancements with regard to DB2
- z/OS Installation Wizard enhancements

More information about ODMP version 9 is available in IBM United States Software Announcement 212-273.

More information about OnDemand for z/OS version 9 is available in IBM United States Software Announcement 212-293.

ECM Forum at IBM IOD

The IBM Information On Demand (IOD) conference is coming up soon. Take a look at where Enterprise Content Management (ECM) will be showcased throughout the conference:

- **Six technical tracks**: Advanced Case Management, Content Analytics, Social Content Management, Document Imaging and Capture, Information Lifecycle Governance, and a cross-portfolio track.
- **30 client and 15 business partner speakers** who will share their experience and best practices about the ECM portfolio.
- **133 technical sessions**: From introductory sessions to deep-dive and roadmap sessions, we offer it all!
- **4 Birds of a Feather sessions**: Network with colleagues in compelling discussions that address technical and business issues important to you.
- **30 Ask the Experts sessions**: Join our experts for informal discussions to address your specific needs and questions.
- **ECM Central**: Attend innovative programming highlighting the latest ECM solutions, network with members of the ECM community, and learn about our client reference programs in the ECM hub.
- **ECM Expo Booth**: See demos from each of the ECM offering teams and talk to the experts -- all in one place!

More information is available on the Information On Demand conference web site.

ODUG Conference

The annual OnDemand User Group (ODUG) conference will be held on Thursday, October 25th, following the IBM IOD conference.

The ODUG agenda includes an IBM product update, discussion of enhancements including full text search and IBM Content Navigator, and a brief ODUG business meeting.

Tips – Cross Platforms

Using the Report Wizard to Create User Defined PDF

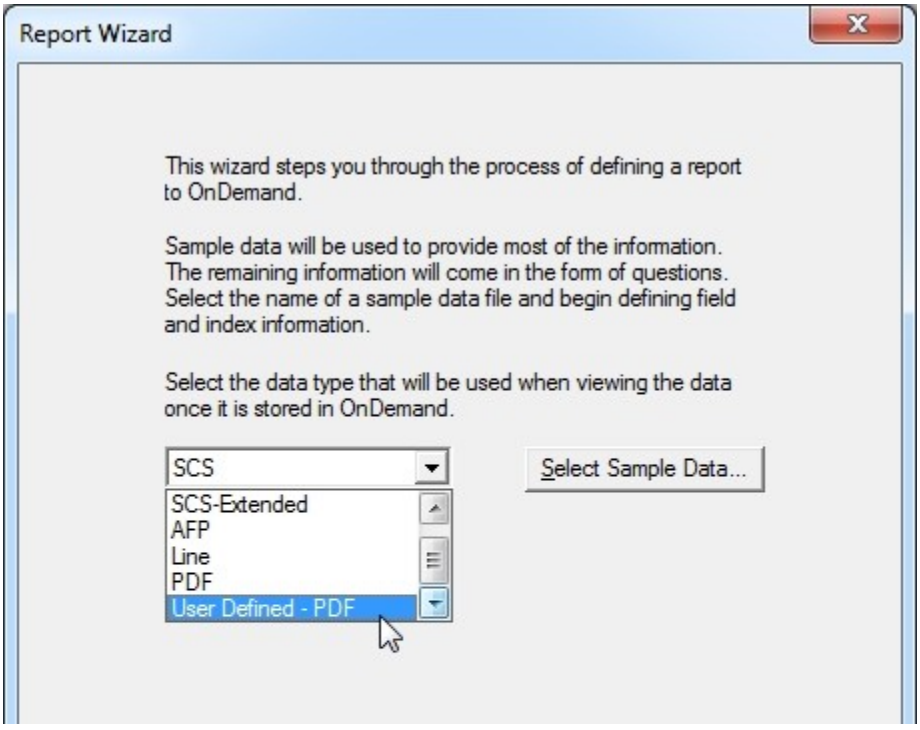
This tip applies to the OnDemand version 9.0 Administrator Client when used with either server version 9.0 or server version 8.5.

More and more customers are creating applications with a data type of User Defined and a file extension of PDF. This allows the end user to use the Adobe Acrobat Reader to view documents. (A data type of PDF requires the full Adobe Acrobat product for document viewing.)

The Report Wizard has been enhanced to provide the option to create applications with a data type of User Defined and a file extension of PDF. Previously, you could not use the Report Wizard to create the report unless you first specified PDF as the data type, created the definitions and then removed and re-added the application with a data type of User Defined.

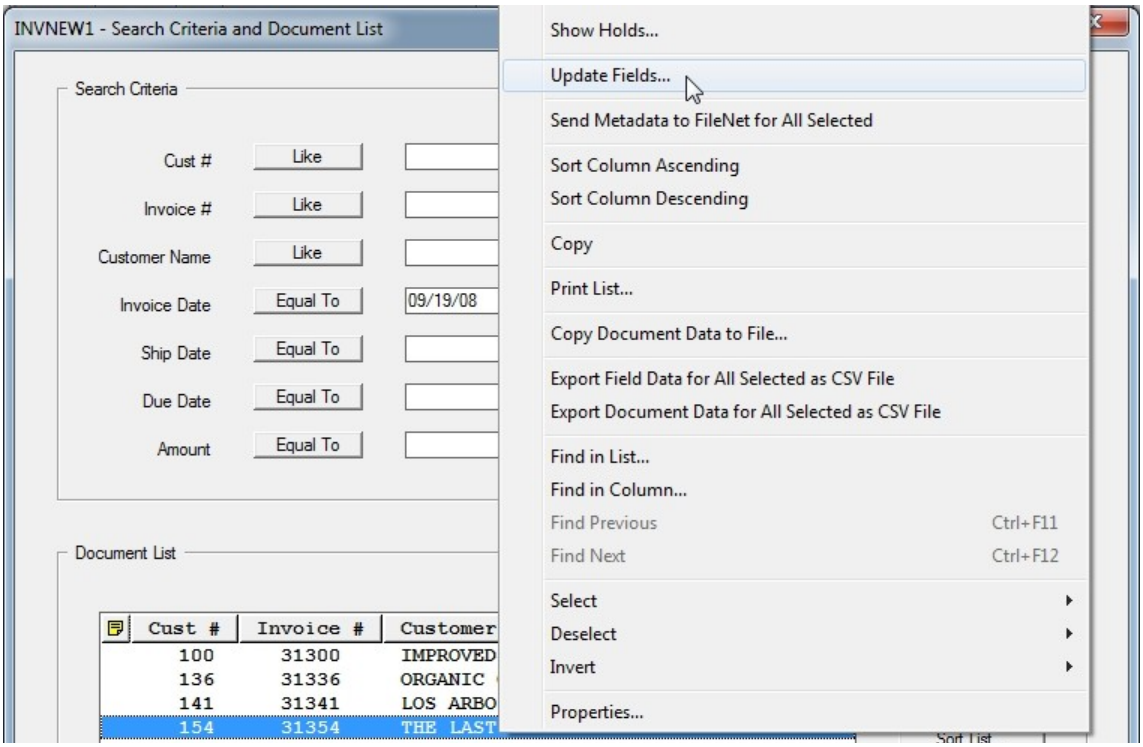
Now you can select 'User Defined - PDF' from the list of data types. Once the data type is selected, the remaining steps are the same as for a data type of PDF.

Note that the full Adobe Acrobat product is still required to mark up PDF documents in the Report Wizard.

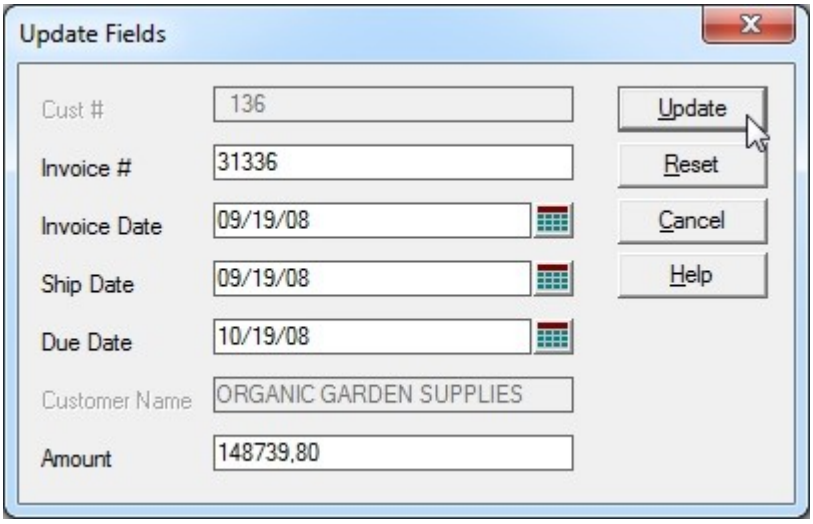


Updating Indexes from the OnDemand Windows Client

This tip applies to the OnDemand version 9.0 Windows Client when used with either server version 9.0 or server version 8.5. By right-clicking a hit in the document list, a new option to "Update Fields" is available if the user has document update permission to the Application Group.



This option will launch a new Update Fields dialog that will allow the user to specify new values for fields associated with this document. Values that are not updateable, either because of the type (ie: Application Alias fields cannot be updated) or the "Updateable" Application Group Field setting, are greyed out and new values cannot be specified.

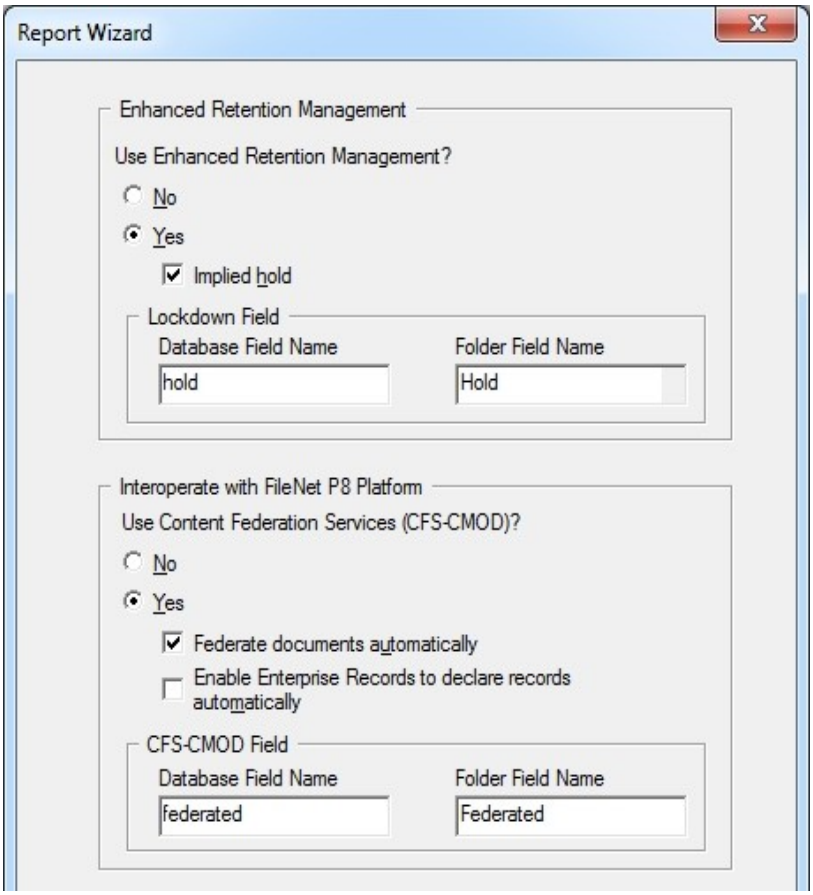


Configuring ERM and CFS-CMOD Using the Report Wizard

This tip applies to the OnDemand version 9.0 Administrator Client when used with either server version 9.0 or server version 8.5.

The Report Wizard has been enhanced to provide the option to create application groups that use Enhanced Retention Management (ERM) and/or Content Federation Services (CFS-CMOD). Previously, if the Report Wizard was used to create the definitions, ERM and CFS-CMOD had to be configured manually in a separate step. Also, once an application group was created, implied hold could not be set for an existing application group. This meant OnDemand administrators could not use the Report Wizard to create a report that used ERM with implied holds. That limitation is now removed.

A new page has been added to the Report Wizard. It contains the same options that are located on the Database Information dialog accessible from the Application Group, General tab. In addition, there are other controls for database fields and folder fields that are used by ERM and CFS-CMOD. The Report Wizard will create the fields with the appropriate data types and attributes. The names of the database and folder fields specified for ERM and CFS-CMOD must be unique because they are also part of the application group.



Tips – Multiplatforms

PDF Indexer Float Trigger Support in V9

OnDemand version 9 includes support for PDF float triggers. This article discusses how they work, how to use them, and how to avoid mistakes in their use. It assumes that you have a basic understanding of indexing concepts.

With the release of version 9, both ACIF and the PDF indexer support two types of triggers: group and float. It is important to understand group triggers in order to understand the difference between group and float triggers.

For this discussion, the documents that are created by the indexing process are called groups.

Note that Graphical indexer support for PDF indexer float triggers will be available in the OnDemand Administrator Client version 9.0.0.1.

PDF Group Trigger

A group trigger locates indexing fields that occur once within a group. For example, in the simplest case, you might have a group trigger consisting of the text "Name" and a field which will contain the actual name extracted from the document. The definitions for the trigger and field parameters are as follows:

```
TRIGGER1=UL(0.57,3.13),LR(4.18,3.61),*,'Name'
FIELD1=UL(0.54,3.70),LR(3.97,4.31),0,(TRIGGER=1,BASE=0)
```

When this trigger is found, the PDF indexer collects the name from the document. Then the PDF indexer begins searching for the next occurrence of the trigger. In order to understand how the groups are created, you must understand where the indexer starts searching for the next occurrence of the trigger. On the next line, the next page, or somewhere else? The answer is that the PDF indexer starts searching on the page following the page where it found the field.

Here are some examples of where the search begins again:

(1) Trigger found on page 1, and the field found on page 1: indexer starts searching for the next occurrence of “Name” on page 2.

(2) Trigger found on page 1, and the field found on page 5: indexer starts searching for the next occurrence of “Name” on page 6. In this case the field would be defined as

```
FIELD1=UL(0.54,3.70),LR(3.97,4.31),4,(TRIGGER=1,BASE=0)
```

The number of pages in the group will be determined by when the trigger is found again and the field value changes. For example, if the trigger is found again on page 2, but the Name is the same as page 1, both pages will be part of the same group, and the indexer will start searching for the trigger on page 3.

In case (1), if the trigger is found again on page 2, and the Name has changed, the first group will contain one page. In case (2), if the trigger is found again on page 6, and the Name has changed,

the first group will contain five pages (pages 1 – 5). If the trigger is not found on page 6, but on page 10, then the first group will contain nine pages (pages 1 – 9). If the trigger is not found again within the document, the first group will contain all of the pages in the file.

If multiple fields are defined, the group changes if any of the field values change.

Since the PDF indexer stops searching on the current page once it finds the trigger, if the trigger were to reoccur on the page it would be ignored, even if it occurred within the bounding box of the trigger definition.

How do group triggers work when there are multiple triggers? The answer is that, after the first trigger is found, the PDF indexer searches for the remaining triggers. When they are all found, the fields are collected. If one of the triggers is not found, the PDF indexer starts the process again; it starts searching for the first trigger on the next page after the location where trigger 1 was originally found.

The complete answer to the question of where does the PDF indexer start searching for the next occurrence of trigger 1 is as follows:

PDF indexer starts searching on the page after the last page where it found a field or trigger for the current group. This rule prevents groups from overlapping and therefore prevents document pages from being stored more than once.

For example, if the first trigger is on page 1, and the second trigger is on page 2, and the fields are on page 1, the PDF indexer will start the search for trigger 1 on page 3.

You might think of the triggers and fields like the tentacles of an octopus that reach out and gather the pages into the group.

Summary of important rules of group triggers:

1. A group trigger can only be found once on a page.
2. All of the group triggers must be found before any fields are collected.
3. When the value of a field which is based on a group trigger changes, a new group is started.

None of these rules apply to float triggers.

PDF float Trigger

A float trigger can be found multiple times on a page, or not at all.

Float triggers operate independently of each other.

Fields based on float triggers do not determine group boundaries.

The need for float triggers came about from trying to index data where the trigger information might or might not occur, or might occur more than once on a page.

Here is an example of a document where a float trigger is needed. In the following statement, the text “Checking Account Balance” or “Savings Account Balance” might or might not occur,

depending on whether the accounts exist. If they do exist, you would like to collect the balance amounts to use as fields.

```
Name  John Smith

Account 9999

Checking Account Balance    500.00
Savings Account Balance     300.00

Total                                    800.00
```

The group trigger will be "Name", the float triggers will be "Checking" and "Savings".

As with group triggers, the bounding box for a float trigger must encompass the area of the page where the float trigger could occur.

Here are the relevant indexing parameters:

```
TRIGGER1=UL(0.14,0.42),LR(0.70,0.67),*,'Name',(TYPE=GROUP)
TRIGGER2=UL(0.17,1.17),LR(0.93,1.43),*,'Checking',(TYPE=FLOAT)
TRIGGER3=UL(0.14,1.31),LR(0.84,1.64),*,'Savings',(TYPE=FLOAT)
FIELD1=UL(0.64,0.40),LR(1.57,0.68),0,(TRIGGER=1,BASE=0)
FIELD2=UL(0.78,0.79),LR(1.29,1.03),0,(TRIGGER=1,BASE=0)
FIELD3=UL(2.07,1.13),LR(2.71,1.40),0,
   (TRIGGER=2,BASE=0,DEFAULT='0.00')
FIELD4=UL(2.07,1.35),LR(2.74,1.60),0,
   (TRIGGER=3,BASE=0,DEFAULT='0.00')
INDEX1='Name',FIELD1,(TYPE=GROUP)
INDEX2='Account',FIELD2,(TYPE=GROUP)
INDEX3='Checking',FIELD3,(TYPE=GROUP)
INDEX4='Savings',FIELD4,(TYPE=GROUP)
```

Notes:

1. All indexes are TYPE=GROUP because they all might appear on the Search Results screen.

2. The float trigger syntax requires an asterisk for the page value.

Unlike group triggers, float triggers operate independently of other float triggers. For example, it is not necessary for the indexer to locate "Checking" and then "Savings" in order to collect the fields that are based on them. If "Checking" is found, the field based on it will be collected, likewise for "Savings".

The index values collected from the float triggers will appear in the Search Results in the same row that contains the index values for the group to which it belongs. The groups are determined

by the group triggers and fields.

For this example the indexes will appear on the Search Results screen as follows:

```
Name          Account      Checking      Savings
John Smith    9999         500.00        300.00
```

If the checking account did not exist then the default value would be used and the Search Results screen would appear as follows:

```
Name          Account      Checking      Savings
John Smith    9999         0.00          300.00
```

If the checking account occurred more than once within the group then the Search Results would appear as shown below. (In this example, the customer has two checking accounts.)

```
Name          Account      Checking      Savings
John Smith    9999         500.00        300.00
John Smith    9999         200.00        300.00
```

Note the following about the last example:

- Both of the rows reference the same document that is stored in OnDemand.
- One row is created for every distinct value which is based on a float trigger.
- Values which have not changed will be repeated in every row.

The following rules apply to float triggers.

1. Since the group trigger and field determine the group boundaries, at least one group trigger and one field based on it are required.
2. A default value is required in the FIELD definition.

   The trigger might not occur, therefore the field might not exist. If the field does not exist, then the default value will be used.
3. Fields based on float triggers cannot determine group boundaries.

   With fields based on group triggers, the group will be determined by when the field value changes. Since the float trigger and field might occur more than once on a page, and the field value might change several times on a page, this would cause the group to start or begin in the middle of a page. Since a group cannot consist of only part of a page, fields based on float triggers cannot determine group boundaries.
4. Fields based on float triggers cannot be combined with other fields in an Index.

For example, with fields based on group triggers, you can do the following:

```
INDEX1='Account',FIELD1,FIELD2
```

Since a field based on a float trigger might not exist, or might occur several times within the group, they cannot be combined with other fields.

Base=0 and Base=Trigger

It is important to understand the difference between Base=0 and Base=Trigger with regard to float triggers.

In the float trigger example in the previous section, FIELD3 is defined as follows:

```
FIELD3=UL(2.07,1.13),LR(2.71,1.40),0,
   (TRIGGER=2,BASE=0,DEFAULT='0.00')
```

BASE=0 means that the coordinates of the field refer to an absolute location on the page. Therefore, if Trigger 2 were to occur multiple times within the bounding box on a single page, the same field would be collected multiple times.

But perhaps you would like to collect several fields that occur at the same offset from a float trigger which occurs multiple times on a page. To accomplish this, you can use BASE=TRIGGER with a float trigger. BASE=TRIGGER means that the field coordinates refer, not to an absolute location, but to a location at an offset from the trigger. Here is an example:

```
Name  John Smith

  Checking Account Number   465789-01
   Savings Account Number   465789-02
```

The group trigger will be "Name", the float trigger will be "Number". The bounding box for the float trigger will encompass both occurrences of "Number". The field based on the float trigger will initially be set to encompass the "465789-01" and will be defined as BASE=TRIGGER. Therefore, when the first occurrence of "Number" is found, the field "465789-01" will be collected. When the second occurrence of "Number" is found, the field "465789-02" will be collected.

Here are the relevant indexing parameters:

```
TRIGGER1=UL(0.14,0.39),LR(0.70,0.68),*,'Name',(TYPE=GROUP)
TRIGGER2=UL(1.40,0.65),LR(2.15,1.40),*,'Number',(TYPE=FLOAT)
FIELD1=UL(0.64,0.39),LR(1.63,0.72),0,(TRIGGER=1,BASE=0)
FIELD2=UL(0.59,0.00),LR(1.54,0.26),0,
   (TRIGGER=2,BASE=TRIGGER,DEFAULT='N/A')
INDEX1='Name',FIELD1,(TYPE=GROUP)
INDEX2='Account',FIELD2,(TYPE=GROUP)
```

The indexes will appear on the Search Results screen as follows:

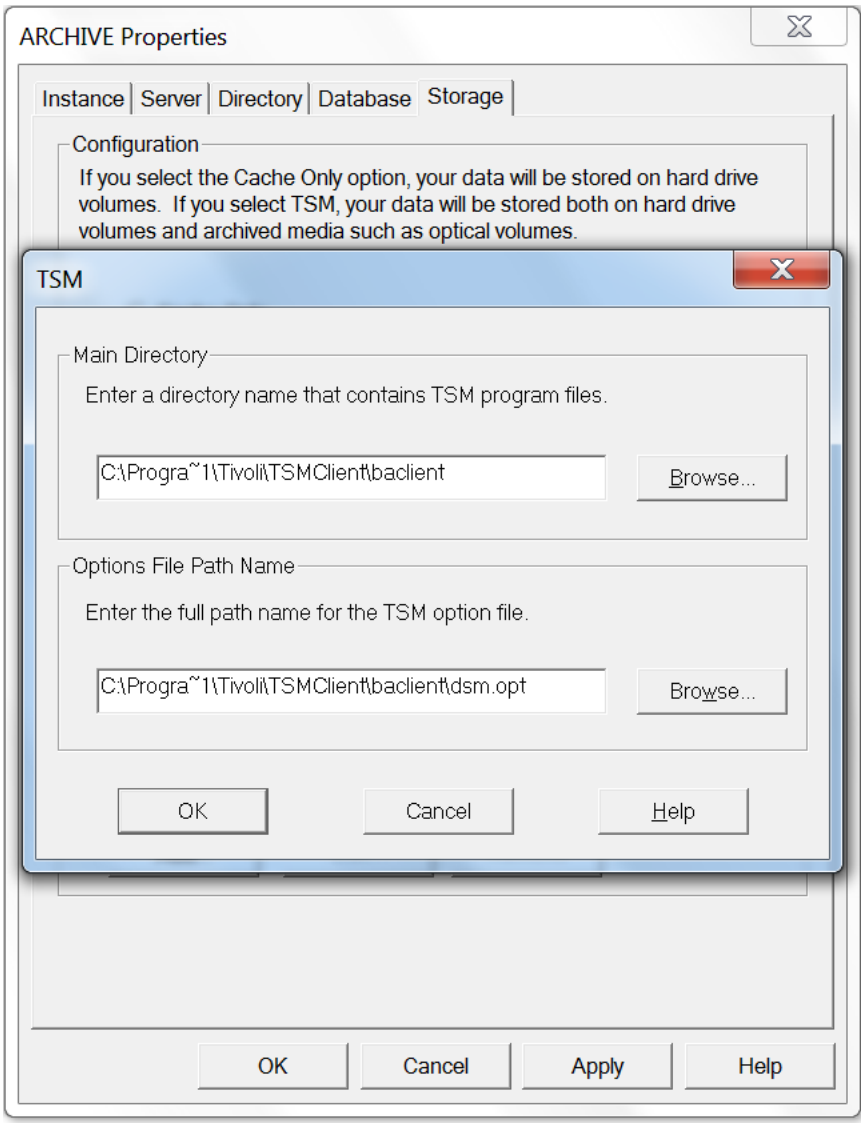
```
Name               Account
John Smith         465789-01
John Smith         465789-02
```

As in the previous example, both of these entries refer to the same document.

Configuring Multiple TSM Servers

Beginning with server version 8.5, a storage set can be configured to use multiple TSM servers.

After setting up the first TSM server in OnDemand, a few more steps are required to use multiple TSM servers. One or more new options files need to be created and added to the directory where options files are located for TSM. The screen capture below shows the TSM dialog in the OnDemand Configurator where the information is located for a Windows server.



For UNIX servers, the directory (DSMI_DIR) and options file (DSMI_CONFIG) information is provided in the ars.cfg file located in the config directory. For example, the ars.cfg file contains the following information:

```
DSMI_DIR=/usr/tivoli/tsm/client/api/bin64
DSMI_CONFIG=/usr/tivoli/tsm/client/api/bin64/dsm.opt
```


Once the location for the options files has been determined, create a new options file (i.e. dsm2.opt) in the same directory. In this example the directory is C:\Progra~1\Tivoli\TSMClient\baclient for Windows and /usr/tivoli/tsm/client/api/bin64 for AIX. The new options file contains information to connect to the second TSM server. Make sure the new node that will be defined in the OnDemand Administrator client is registered in the second TSM server. In this example, the node name is TSM2.

Example options file on Windows:

dsm2.opt:

```
COMMMETHOD                          TCPIP
TCPPORT                             1500
TCPSERVERADDRESS                    objsrvr2.svl.ibm.com
COMPRESSION                         OFF
ENABLEARCHIVERETENTIONPROTECTION    YES
```

For UNIX servers, the dsm2.opt is created and the dsm.sys is updated to include the information for the additional TSM server:

dsm2.opt:

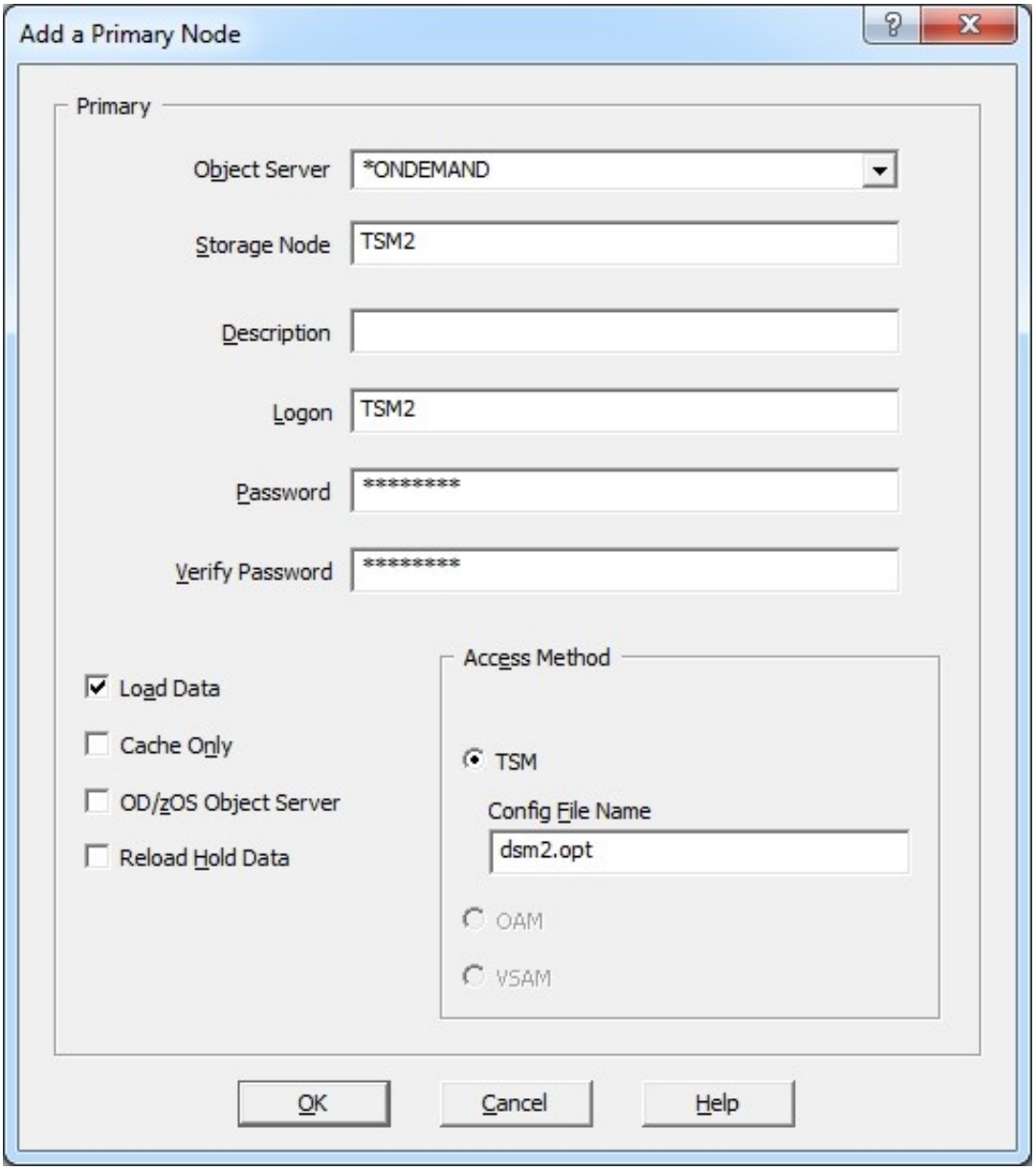
```
SERVERNAME                          TSM2
QUIET
```

dsm.sys:

```
SERVERNAME                          TSM1
COMMMETHOD                          TCPIP
TCPPORT                             1500
TCPSERVERADDRESS                    objsrvr1.svl.ibm.com
COMPRESSION                         OFF
ENABLEARCHIVERETENTIONPROTECTION    YES

SERVERNAME                          TSM2
COMMMETHOD                          TCPIP
TCPPORT                             1500
TCPSERVERADDRESS                    objsrvr2.svl.ibm.com
COMPRESSION                         OFF
ENABLEARCHIVERETENTIONPROTECTION    YES
```

Next, create a primary node to connect to the second TSM server using the OnDemand Administrator client and add the newly created options file name in the "Config File Name" field shown in the screen capture below:



The newly created storage node is now ready to save OnDemand documents.

Tips – z/OS

OnDemand for z/OS Quick Hits

Why Am I Receiving Message ICH408I After Upgrading to OnDemand v8.5?

If you receive messages similar to this:

```
ICH408I USER(ARSSERV2) GROUP(ODCMARS ) NAME(XXXXXXXX) 353
   353   CSFIQF CL(CSFSERV )
   353   INSUFFICIENT ACCESS AUTHORITY
   353   FROM * (G)
   353   ACCESS INTENT(READ) ACCESS ALLOWED(NONE)
```

The reason is:

One of the enhancements with OnDemand version 8.5 is the encryption of passwords with AES-128, both over the wire and in the OnDemand database. This gets done automatically if the Cryptographic Service Facility (CSF) is available and providing AES-128 service.

How does OnDemand find out if AES-128 is available? It calls CSFIQF - ICSF Query Facility callable service. It turns out that even though it is a query type service, you can lock it down via RACF. OnDemand uses CSFIQF and must have permission to it.

OnDemand will function without the permission but you will receive error messages from RACF, and OnDemand will use the pre 8.5 methods for the passwords over the wire and in the database.

What Browser Plugins Ship with OnDemand?

There are two browser plugins that ship with OnDemand:

1) The AFP Viewer for Advanced Function Presentation Data Stream (AFPDS) reports

2) The Image Viewer for TIFF, BMP, JPEG, and other image types

On z/OS, both are available in the /usr/lpp/ars/V8R5M0/www/plugins/ directory. The AFP Viewer is available as either an executable file, or as a zip file. Just FTP, execute and you are ready to view AFP or image data in your web browser.

```
afpplgin.zip
imgplgin.exe
afpplgin.exe
```

To determine what plugins you have installed:

- In Firefox, press Ctrl + Shift + A to display the Add-ons Manager
- In Internet Explorer, use the menu option Tools, then Manage Add-ons

OS/390 Indexer Usage Notes: The INDEXSTYLE Parameter - Part 4 of 4

This is the fourth of a four part series on the use of the INDEXSTYLE indexing parameter in the OS/390 indexer. The OS/390 indexer is available on z/OS (all versions) and Multiplatforms (version 9 only). This part covers the INDEXSTYLE=AFP usage. Part 1 covered the NODX usage. Part 2 covered the PDOC usage. Part 3 covered the PAGE usage. Refer to the Indexing Reference manual, Part 5 OS/390 indexer Reference, Chapter 29 OS/390 indexer Parameters, for more information on the INDEXSTYLE indexing parameter.

INDEXSTYLE=AFP

The AFP index style provides two ways of extracting index values from an AFP Data Stream (AFPDS).

- TLE records
- NOP records

The indexing instructions are defined differently for these two indexing techniques. The OS/390 indexer does not support indexing a single AFPDS file using both methods.

Note that in the following examples, parameters and values not relevant to understanding the INDEXSTYLE parameter are replaced with an ellipsis (…).

Application Group definition

There is nothing special about the Application Group definition. Define whatever fields are appropriate for your report. The OS/390 indexer does not support any Range Indexes with INDEXSTYLE=AFP. The indexes defined in the Application Group must relate to the values available on the TLE or NOP records, with the exception of an Application ID index. Default values can be specified in the Application definition on the Load Information tab.

The Application definition in the next section will provide examples of using both TLE and NOP records for indexing reports using the following Application Group definition.

```
APPLICATION GROUP NAME:  CHKSTAFP

*** GENERAL ***
  DESCRIPTION:             CHECKING ACCOUNT STATEMENTS
  ...

  *** APPLICATIONS ***
  APPLICATION NAME:  CHKSTAFP

*** FIELD INFORMATION ***
  NAME:  ACCOUNT_NUMBER
    SEGMENT:        No
    DATA TYPE:      String
    TYPE:           Index
    ...
```



```
NAME:  CUST_NAME
  SEGMENT:         No
  DATA TYPE:       String
  TYPE:            Index
  ...

NAME:  POSTING_DATE
  SEGMENT:         Yes
  DATA TYPE:       Date
  TYPE:            Index
  ...

NAME:  VERSION
  SEGMENT:         No
  DATA TYPE:       String
  TYPE:            Filter
  APPL ID FIELD:   Yes
  ...
```

Application definition

The way in which the indexing instructions are specified is different, depending on whether the index values are coming from TLE or NOP records.

Indexing Instructions for TLE Records

A TLE record consists of two parts:

Attribute Name

Attribute Value

The Attribute Name must match the corresponding name value in the INDEXn instruction of the indexing parameters. In the above Application Group definition, two of the fields will come from TLE records: ACCOUNT_NUMBER and CUST_NAME. The Attribute Names on the TLE records for these two fields are KEY1 and KEY2 respectively.

To get everything to match up, the name field on the INDEXn instructions of the indexing parameters will use the KEY1 and KEY2 values. These names are mapped to the Application Group field names through the Load Information tab of the Application definition.

Note that you could have set the Attribute Names on the TLE records to be ACCOUNT_NUMBER and CUST_NAME. Then the mappings done on the Load Information tab would not need to be set. But you sometimes have to take what a third party application gives you. Being able to map the TLE Attribute Name values to the Application Group field names gives you this flexibility.

Here is the Application definition for this report:

```
APPLICATION NAME:  CHKSTAFP

*** GENERAL ***
```



```
  APPLICATION GROUP NAME:  CHKSTAFP
  IDENTIFIER:              06
  DESCRIPTION:             CHECKING ACCOUNT STATEMENTS

*** INDEXER PARAMETERS ***
  INDEXER:  OS/390

TRIGGER1=*,1,X'5A',(TYPE=GROUP)  /* AFP x'5A' */
FIELD1=-0,1,9,(TRIGGER=1,BASE=0)
FIELD2=0,1,12,(TRIGGER=1,BASE=0)
INDEX1=X'D2C5E8F1',FIELD1,(TYPE=GROUP,BREAK=YES) /* ACCOUNT_NUMBER (KEY1) */
INDEX2=X'D2C5E8F2',FIELD2,(TYPE=GROUP,BREAK=NO)  /* CUST_NAME (KEY2) */
INDEXSTYLE=AFP

*** LOAD INFORMATION ***
  ...

  APPLICATION GROUP DB NAME:  ACCOUNT_NUMBER
  LOAD ID NAME:               KEY1
  ...

  APPLICATION GROUP DB NAME:  CUST_NAME
  LOAD ID NAME:               KEY2
  ...

  APPLICATION GROUP DB NAME:  POSTING_DATE
  LOAD ID NAME:
  DEFAULT VALUE:              t
  FORMAT:                     %m-%d-%y
  ...
```

In the INDEXER PARAMETERS section, the TRIGGER1 parameter is not used by the OS/390 indexer. It exists only to meet syntax checking requirements of the graphical indexer.

On the FIELD1 and FIELD2 parameters, only the length value is used by the OS/390 indexer. This tells the indexer how many positions to use from the Attribute Value portion of the TLE record.

The name values on the INDEX1 and INDEX2 parameters must match the values of the Attribute Name field of the TLE records. The FIELDn parameter of the INDEXn instruction must point back to the FIELDn instruction with the correct length value for each index. The BREAK parameter of the INDEXn instruction is not used.

In the Load Information section, the Load ID Name value corresponds to the INDEXn name value (and the TLE Attribute Name value). The Application Group DB Name value corresponds to the names defined in the Application Group definition.

In this example, the POSTING_DATE value does not come from the report data, but defaults to the capture date.

Indexing Instructions for NOP Records

The NOP records used for indexing have a fixed layout. Refer to the Indexing Reference manual, Part 5, for a complete definition of this layout. Note that NOP definitions from prior releases of OnDemand for OS/390 V2.1 and R/DARS V2.3 are still supported.

Two types of NOP records can be used. ODZOSSEG records identify the start of a new document and provide one set of indexing values for that document. Each document within the AFPDS must have one, and only one, ODZOSSEG NOP record.

Optionally, each document within the AFPDS can have one thru 99 ODZOSDIR NOP records. Each ODZOSDIR record provides an additional set of index values for the document that are located within the AFPDS.

The layout of the ODZODSEG and ODZOSDIR records is positional, supporting up to 32 index values of up to 256 characters each. The trick is to be sure each index value from the NOP record gets associated to the correct Application Group field.

To accomplish this, the indexing instructions point to the position within the NOP record. The following Application definition can be used with the previously defined Application Group definition to capture the indexes for an AFPDS using NOP records.

```
APPLICATION NAME:  CHKSTNOP

*** GENERAL ***
  APPLICATION GROUP NAME:  CHKSTAFP
  IDENTIFIER:              01
  DESCRIPTION:             AFP WITH ODZOSSEG NOPS

*** INDEXER PARAMETERS ***
  INDEXER:  OS/390

TRIGGER1=*,1,X'5A',(TYPE=GROUP)  /* AFP X'5A'  */
FIELD1=-0,1,9,(TRIGGER=1,BASE=0)
FIELD2=0,1,12,(TRIGGER=1,BASE=0)
INDEX1=X'F1',FIELD1,(TYPE=GROUP,BREAK=NO)  /* ACCOUNT_NUMBER */
INDEX2=X'F2',FIELD2,(TYPE=GROUP,BREAK=NO)  /* CUST_NAME */
INDEXSTYLE=AFP

*** LOAD INFORMATION ***
  ...

  APPLICATION GROUP DB NAME:  ACCOUNT_NUMBER
  LOAD ID NAME:               1
  ...

  APPLICATION GROUP DB NAME:  CUST_NAME
  LOAD ID NAME:               2
  ...
```



```
APPLICATION GROUP DB NAME:  POSTING_DATE
LOAD ID NAME:
DEFAULT VALUE:              t
FORMAT:                     %m/%d/%y
...
```

In the INDEXER PARAMETERS section, the TRIGGER1 parameter is not used by the OS/390 indexer. It exists only to meet syntax checking requirements of the graphical indexer.

On the FIELD1 and FIELD2 parameters, only the length value is used by the OS/390 indexer. This tells the indexer how many positions to use from the NOP record.

The name values on the INDEX1 and INDEX2 parameters must point to the position within the NOP record to be used for each INDEX. In this example, the INDEX1 instruction points to the first position of the NOP record by using the value X'F1' (the character 1). The INDEX2 instruction points to the second position of the NOP record by using the value X'F2' (the character 2). This format for mapping the INDEXn record to the correct positions within the NOP records is mandatory. The values of X'F1' through X'F3F2' are the only values recognized by the OS/390 indexer.

The FIELDn parameter of the INDEXn instruction must point back to the FIELDn instruction with the correct length value for each index. The BREAK parameter of the INDEXn instruction is not used.

In the Load Information section, the Load ID Name value corresponds to the INDEXn name value. The Application Group DB Name value corresponds to the names defined in the Application Group definition.

In this example, the POSTING_DATE value does not come from the NOP record, but defaults to the capture date.

A note about placing the NOP records in the AFPDS:

The assumption here is that the AFPDS does not contain BNG, ENG or TLE records. It does contain BPG and EPG records. The ODZOSSEG NOP record must be positioned immediately after the BPG record for the page that is to start the new document. Any ODZOSDIR NOP records that are used for this document must be positioned immediately after the ODZOSSEG NOP record.

Folder Definition

The following summarizes the definition of an AFP Folder:

```
FOLDER NAME:  CHKSTAFP

*** GENERAL  ***
  DESCRIPTION:                AFP Statements
  NOTE SEARCH:                Retrieve
  DISPLAY DOCUMENT LOCATION:  No

  APPLICATION GROUP NAME:  CHKSTAFP
```



```
    APPLICATION NAME:  CHKSTAFP

*** FIELD INFORMATION ***
    ID:  *PUBLIC

    NAME:              ACCOUNT NUMBER
    TYPE:              String
    MAPPING TYPE:      Single
    DESC:
    QUERY:             1
    HIT LIST:          2
    ...

    NAME:              CUSTOMER NAME
    TYPE:              String
    MAPPING TYPE:      Single
    DESC:
    QUERY:             2
    HIT LIST:          3
    ...

    NAME:              POSTING_DATE
    TYPE:              Date
    MAPPING TYPE:      Single
    DESC:
    QUERY:             3
    HIT LIST:          1
    ...

    NAME:              VERSION
    TYPE:              String
    MAPPING TYPE:      Single
    DESC:
    QUERY:             0
    HIT LIST:          0
    ...

*** FIELD MAPPING  ***
  FOLDER FIELD NAME :: APPLICATION GROUP NAME: DATABASE NAME

  ACCOUNT NUMBER       :: CHKSTAFP: ACCOUNT_NUMBER
  CUSTOMER NAME 20CHAR :: CHKSTAFP: CUST_NAME
  POSTING_DATE         :: CHKSTAFP: POSTING_DATE
  VERSION              :: CHKSTAFP: VERSION
```

This Folder definition is pretty simple. No special concerns are being identified. Both the CHKSTAFP and CHKSTNOP Applications are part of the same Application Group of CHKSTAFP. The Folder Fields are mapped to fields of this one Application Group.

Tips – IBM i

OnDemand for i Quick Hits

Are You Archiving From ODi to ODMP?

If you are archiving from an OnDemand for i (ODi) system to an OnDemand for Multiplatforms (ODMP) system, it is important that you update the system log on the ODMP system to support 2000 characters of message text. If you do not perform this step you might receive SQL truncation errors on the ODMP system, such as "DB Error [IBM] CLI Driver] CLI0109E String Data right truncation."

Directions for updating the system log application group are found in the readme.txt file for version 8.4.1.

How Should I Encode My XML Files When Using ARSXML With OnDemand for i?

We recommend that XML input files be created with a CCSID of 1208 or 819. These CCSIDs are compatible with the default xml encoding of UTF-8.

If the XML input file is created with an EBCDIC CCSID, for example 37, and the xml encoding is UTF-8, the ARSXML command might fail with error "CPDB609, File CCSID not valid". A circumvention for this error is to change the XML encoding specified in the file to iso-8859-1, for example:

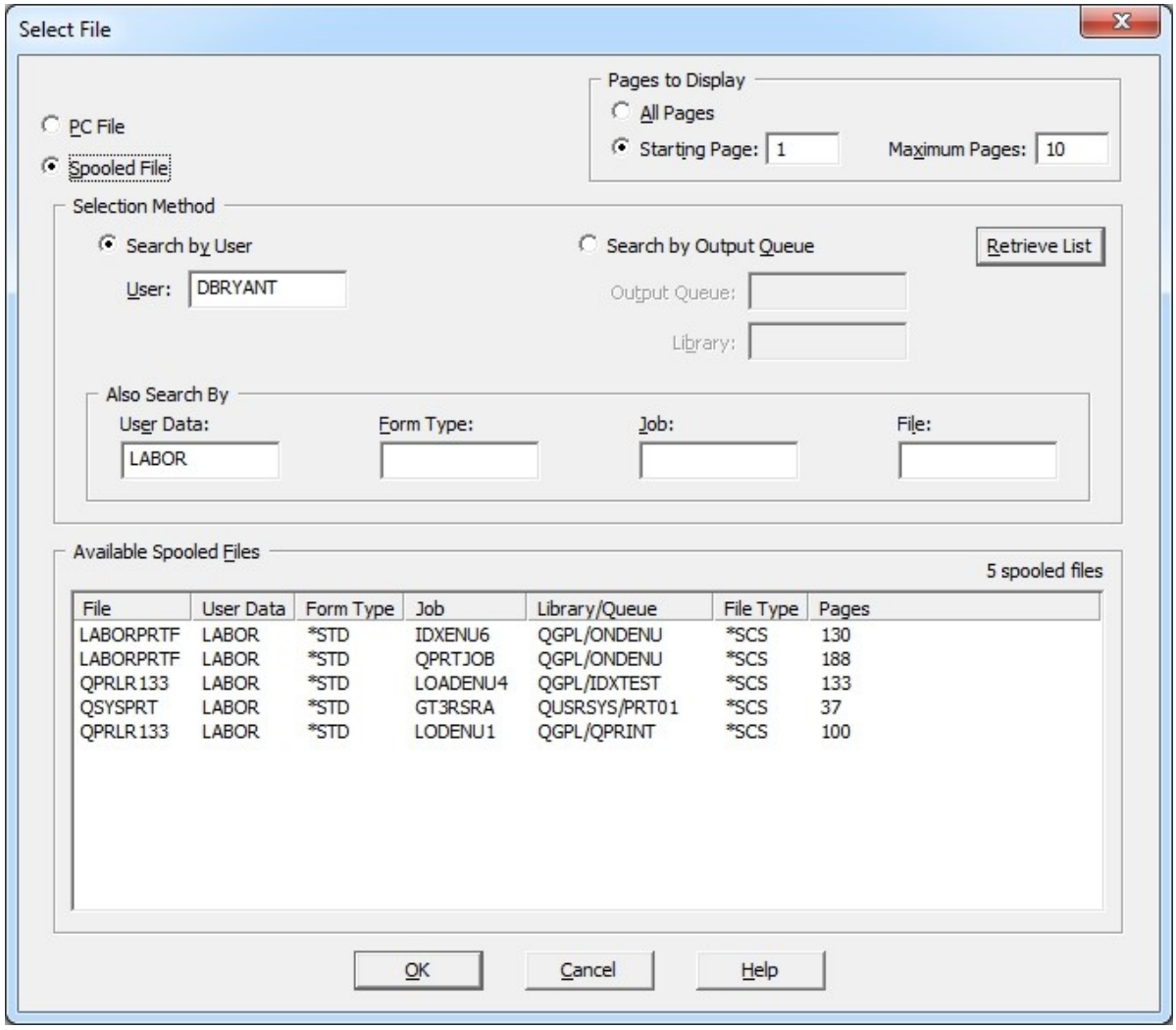
```
<?xml version="1.0" encoding="iso-8859-1"?>
```

This tip is adapted from support item 1607235.

Enhanced Spooled File Selection Using the 9.0 Administrator Client

When using the version 9.0 Administrator Client with the OnDemand for i version 8.5.0.5 server (available only at v7.1) there are additional options for selecting the spooled file to use as sample data in the graphical indexer.

You can now specify the User Data, Form Type, Job name or File name. These can be used separately or together to produce a smaller list of spooled files to select from. Note that User Data and Form Type are case sensitive, that wildcards are not supported, and that either a User or Output Queue must also be specified.



Enhanced Spooled File Selection at v7.1

You can now set the number of spooled files searched when retrieving a list of spooled files from the Select File panel of the OnDemand Administrator client.

This enhancement is available at v7.1, server version 8.5.0.5, when PTF SI47369 is installed. This enhancement works with either the version 8.5.0.5 or version 9.0 Administrator Client.

If the maximum number of spooled files to search is reached and the number of spooled files that match the search is less than the maximum number of spooled files searched, a message is displayed indicating "A total of nnnn spooled files were searched. Only the spooled files that matched the search criteria are listed." The maximum number of spooled files searched (nnnn) will either be 1000 or a user-defined value that is specified in data area QRLRGCNT in library QUSRRDARS. To specify a maximum other than 1000, the data area QRLRGCNT must be created in QUSRRDARS.

For example, to create the QRLRGCNT data area and set the maximum number of spooled files searched to 2000:

```
CRTDTAARA DTAARA(QUSRRDARS/QRLRGCNT) TYPE(*DEC) LEN(4 0) VALUE(2000)
```

If the value in the data area is 0, the maximum spooled files to search will be set to 1000. If the value in the data area is less than 100, the maximum spooled files to search will be set to 100. The maximum value allowed in the data area is 9999.

This tip is adapted from support item 1606409.

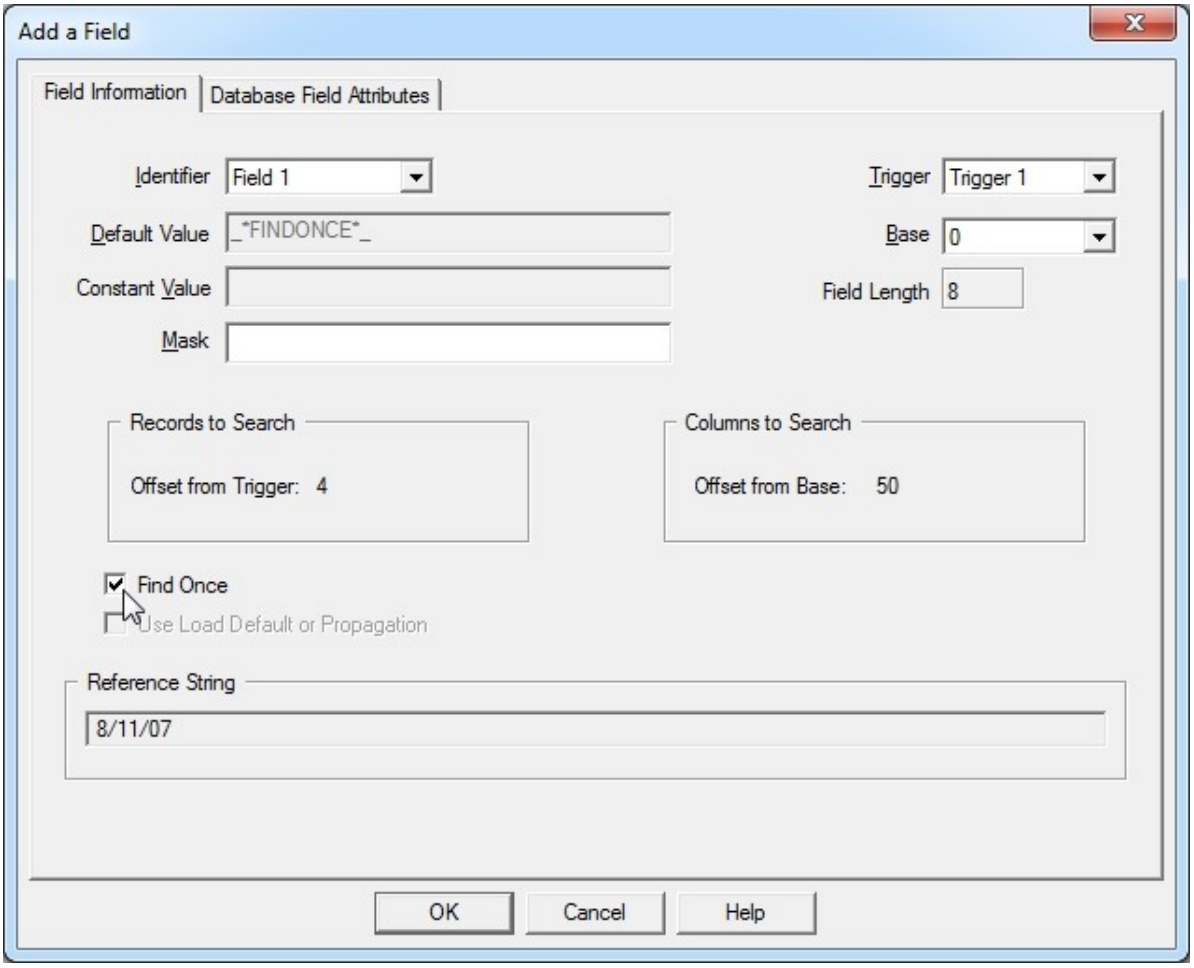
Graphical Indexer Support for the Find Once Option

The version 9.0 Administrator Client adds a capability to specify the Find Once option when used with an OnDemand for i version 8.5.0.5 server.

The OS/400 indexer provides a way to find a value in the first document and use that value in all of the other documents in the report. This is accomplished by using a default value of "_*FINDONCE*_". The indexer parameter looks like the following:

```
FIELD6=0,108,9,
   (TRIGGER=3,BASE=0,DEFAULT=X'6D5CC6C9D5C4D6D5C3C55C6D') /*
   _*FINDONCE*_ */
```

A change has been made to add a 'Find Once' check box when the indexer is OS/400 so that the default value can be set to _*FINDONCE*_ when using the graphical indexer.



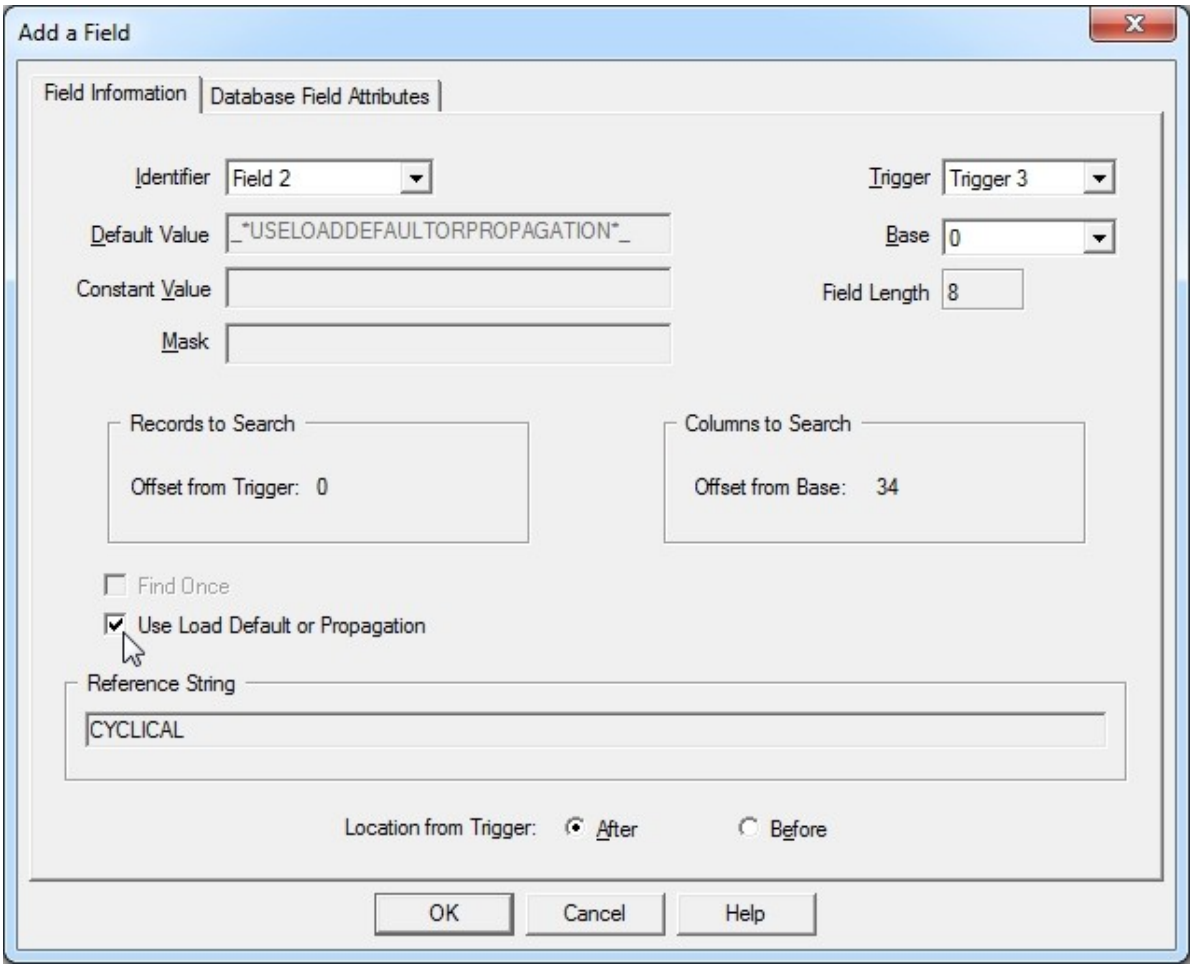
Graphical Indexer Support for the Use Load Default or Propagation Option

The version 9.0 Administrator Client adds a capability to specify the Use Load Default or Propagation option when used with an OnDemand for i version 8.5.0.5 server.

The OS/400 indexer provides a way to use the default value from the Load tab of the Application or to propagate the value from a previous document. This is accomplished by using a default value of "_*USELOADDEFAULTORPROPAGATION*_". The indexer parameter looks like the following:

```
FIELD1=0,40,10,
   (TRIGGER=1,BASE=0,DEFAULT=x'6D5CE4E2C5D3D6C1C4C4C5C6C1E4D3E3D6D9D7
   D9D6D7C1C7C1E3C9D6D55C6D') /* _*USELOADDEFAULTORPROPAGATION*_ */
```

A 'Use Load Default or Propagation' check box is now available when the indexer is OS/400 so that the default value can be set to _*USELOADDEFAULTORPROPAGATION*_ when using the graphical indexer.



Additional Information

Additional information about IBM Content Manager OnDemand can be found at the following web sites.

Information Centers

OnDemand Version 9.0 Information Center

OnDemand Version 8.5 Information Center

OnDemand Version 8.4.1 Information Center

Publication Libraries - Containing all PDF versions of the documentation

OnDemand for Multiplatforms Version 9.0 Publication Library

OnDemand for Multiplatforms Version 8.5 Publication Library

OnDemand for z/OS Version 9.0 Publication Library

OnDemand for z/OS Version 8.5 Publication Library

OnDemand for i Version 7.1 Publication Library

Product System Requirements

OnDemand for Multiplatforms Version 9.0 System Requirements

OnDemand for z/OS Version 9.0 System Requirements

OnDemand for i Version 7.1 System Requirements

More OnDemand Web Sites

OnDemand Product Overview

OnDemand Information Roadmap

Compatibility Matrix for the OnDemand Client and Servers

OnDemand User Group

The primary objective of the OnDemand User Group (ODUG) is to create an environment and network encouraging the exchange and development of information regarding Content Manager OnDemand and its associated products.

Copyright and Trademark Information