FAST SMART CONNECTOR FOR HUMMINGBIRD DM 5

version 4.0.1

MODULE GUIDE

Document Number: DS930, Document Revision: A, March 17, 2005

Copyright

Trademarks

Restricted Rights Legend

Contents

FAST Support

Website

Please visit us at:

`http://www.fastsearch.com/`

Contacting FAST

Fast Search & Transfer Inc.
Cutler Lake Corporate Center
117 Kendrick Street, Suite 100
Needham, MA 02492 USA
Tel: +1 (781) 304-2400 (8:30am - 5:30pm EST)
Fax: +1 (781) 304-2410

Technical Support and Licensing Procedures

E-mail: `fds-support@fastsearch.com`

Product Training

E-mail: fastuniversity@fastsearch.com

Sales

E-mail: `sales@fastsearch.com`

About this Guide

Purpose of this Guide

This guide describes the FAST Smart Connector for Hummingbird DM 5 and explains how to use it.

Audience

This guide provides information for all users of the FAST Smart Connector for Hummingbird DM 5.

Conventions

This guide uses the following textual conventions:

- Terminal output, contents of plaintext ASCII files will be represented using the following format:

  ```
  Answer yes to place the node in the known_hosts file.
  ```

- Terminal input from operators will be in the same but bold format:

  ```
  chmod 755 $HOME
  ```

- Input of some logic meaning will be enclosed in <> brackets:

  ```
  setup_<OS>.tar.gz
  ```

  where <OS> represents a specific operating system that must be entered.

- URLs, directory paths, commands, and the names of files, tags, and fields in paragraphs appear in the following format:

  The default home directory is the *C:\DataSearch* directory.

- User Interface page/window texts, buttons, and lists appear in the following format:

  Click **Next** and the **License Agreement** screen is displayed.

- *$FASTSEARCH* (UNIX) or *%FASTSEARCH%* (Windows) refer to an environment variable set to the directory where FAST Data Search is installed.

Chapter 1

Installing the Hummingbird Connector

Software Requirements

- Any Windows machine on which you install the connector needs to be fully patched using Windows Update.
- An installation of FAST Data Search 4.0 or later, not necessarily on the machine where the Hummingbird Connector is installed.
- A full version of the Oracle client (for Hummingbird DM installations using Oracle database) on the machine where the Hummingbird Connector is used. The client is used both for testing a connection and for retrieving data.

Assumptions

Configuration file is structured in such a way that names of all the items belonging to the same logical group are started with the same string. In this document we will refer to these items as "SECTION" of the configuration file. For example the following two items describe "processlibraries" item and "messagelevel" item of the "general" section of the configuration:

```
<parameter name="general_processlibraries"
  comment="list of libraries to process separated with '|'">
  <![CDATA[DM_Legal_Ora]]>
</parameter>
<parameter name="general_messagelevel"
  comment="Message level 0 - 8 less to more verbose">
  <![CDATA[8]]>
</parameter>
```

Note! All values of the "name=" attribute in the configuration file have to be entered in the lower case!

Delivery Contents

At this time, Hummingbird DM Connector delivery consists of one directory containing all the DLLs from the connector API and DSAPI kits used in the build, connector configuration file and a connector executable.

Installation

In this chapter, we make the following assumptions:

- **a** the connector name is DOCon4
- **b** the connector working directory is located in C:\DOConSetup\bin
- **c** the directory with API dlls is located in C:\DOConSetup\bin

1 Create Connector Working Directory, for example C:\DOConSetup\bin

2 Copy connector distribution directory into the working directory (C:\DOConSetup\bin).

3 Make API DLLs accessible from the PATH environment variable. This can be done by either of the following two methods:

- **a** Copy all DLLs of the C:\DOConSetup\bin directory to the Windows\system32 directory or any other directory that is in the search path for DLLs.
- **b** Add C:\DOConSetup\bin to the PATH environment variable using command:

```
set PATH=%PATH%;C:\DOConSetup\bin
```

4 The connector configuration file should be located in the connector working directory and is named C:\DOConSetup\bin\DOCon4.xml.

Setup

Make the following adjustments to the connector configuration file (C:\DOCon-Setup\bin\DOCon4.xml):

[general] Section

1 In the [general] section of the configuration file, set the processlibraries item to the name of the library you want this connector to work with. If you want the connector to work with several Hummingbird DM libraries, separate the library names with the '|'. The following line will set the connector to work with DM_Legal_Ora library:

```
<parameter name="general_processlibraries"
  comment="list of libraries to process separated with '|'">
  <![CDATA[DM_Legal_Ora]]>
</parameter>
```

For each library listed in the "processlibraries" item, there should be a separate [library] section in the configuration file describing the specific library. In our example, there should be a section called [dm_legal_ora].

2 In the [general] section of the configuration file, set the "messagelevel" item to any number between 0 and 8. This will set the verbose parameter of the connector. 0 will log no messages into the log file and 8 will make the maximum logging.

```
<parameter name="general_messagelevel" comment="Message level 0 - 8
less to more verbose"> <![CDATA[8]]> </parameter>
```

3 In the [general] section of the configuration file, set the "logconfig" item to the configuration file of the logger.

```
<parameter name="general_logconfig" comment="Whatever logger needs for
configuration of the LogConfig item">  <![CDATA[logconfig.xml:debug]]>
</parameter>
```

4 In the [general] section of the configuration file, set the "uniqueid" item to the unique ID of the Hummingbird DM installation. This is an arbitrary unique ID you give to the installation of the Hummingbird DM.

```
<parameter name="general_uniqueid" comment="Unique ID of the DOCSOpen
server">  <![CDATA[kareta_DOCS]]></parameter>
```

5 In the [general] section of the configuration file, set the "connectorid" item to the arbitrary unique name of the connector.

```
<parameter name="general_connectorid" comment="Connector ID for
logging">  <![CDATA[karetaConnector]]></parameter>
```

6 In the [general] section of the configuration file set the "errdocsfile" item to the name of the error file. This file will get the list of all document numbers that were not processed correctly during the run. If this name is not given a new file with the name

YYYYMMDDHHMM.bad will be created in the connector working directory every time the connector is started

```
<parameter name="general_errdocsfile" comment="Name of file for the
list of documents that were not processed. If this name is not given a
new file with the name YYYYMMDDHHMM.bad will be created in the same
directory as config every time the connector is started">
<![CDATA[ErrDocsFile.bad]]></parameter>
```

7 In the [general] section of the configuration file, set the "batchsize" item to the preferred DM API batch size. Large batch sizes may cause the connector to hang.

```
<parameter name="general_batchsize" comment="Size of DSAPI batches">
<![CDATA[20]]></parameter>
```

[library] Section

For our example, the name of the library section should be [dm_legal_ora]. If you use a different library name - replace string "dm_legal_ora" by your library name throughout the configuration file.

1 In the [dm_legal_ora] section of the configuration file, set the DocProcessList item to the name of the file with the list of documents that were not processed correctly in the previous run and that you want to be reprocessed during the current run. This file is created as ErrDocsFile (see item 7. in [Generic] configuration section).

```
DocProcessList=ProcDocsFile.proc
```

2 In the [dm_legal_ora] section of the configuration file, set the "connectstring" item to the first part of the connect string to the Hummingbird DM library database.

```
<parameter name="dm_legal_ora_connectstring" comment="Start of the
database connect string (included in the log)">
<![CDATA[Provider=MSDAORA;Data Source=TAN;User
Id=system;Password=]]></parameter>
```

The connection string should be an OLEDB connection string. Please refer to your OLEDB provider documentation for specific parameters. The connector has been tested with SQLOLEDB provider for MS SQL Server 2000 and with MSDAORA provider for Oracle 8i.

3 In the [dm_legal_ora] section of the configuration file, set the "connecttail" item to the password part of the connect string to the Hummingbird DM library database.

```
<parameter name="dm_legal_ora_connecttail" comment="Remainder of the
database connect string (not included in the log)">
<![CDATA[manager]]></parameter>
```

4 In the [dm_legal_ora] section of the configuration file, set the "fasthost" item to the host name or IP address of the FAST server:

```
<parameter name="dm_legal_ora_fasthost" comment="FAST host name or
IP">  <![CDATA[cosk]]></parameter>
```

5 In the [dm_legal_ora] section of the configuration file, set the "fastcollection" item to the name of the collection for this library:

```
<parameter name="dm_legal_ora_fastcollection" comment="FAST collection
name">  <![CDATA[OracleQA]]></parameter>
```

6 In the [dm_legal_ora] section of the configuration file, set the "fastport" item to the FAST port number:

```
<parameter name="dm_legal_ora_fastport" comment="#FAST port number">
<![CDATA[16099]]></parameter>
```

7 In the [dm_legal_ora] section of the configuration file, set the "serverindexname" to the name of the FAST field for the unique ID of the Hummingbird DM Server.

```
<parameter name="dm_legal_ora_serverindexname" comment="FAST index
name for the unique ID of the DOCSOpen server">
<![CDATA[servername]]></parameter>
```

8 In the [dm_legal_ora] section of the configuration file, set the "libindexname" item to the name of the FAST field for the library name.

```
<parameter name="dm_legal_ora_libindexname" comment="FAST index name
for the library name">  <![CDATA[databaseid]]></parameter>
```

9 In the [dm_legal_ora] section of the configuration file, set the "libname" item to the name you want to map this library to in the FAST repository.

```
<parameter name="dm_legal_ora_libname" comment="FAST index value for
the library">  <![CDATA[dm_legal_ora]]></parameter>
```

10 In the [dm_legal_ora] section of the configuration file, set the "hummingbirduser" item to the ID of the user that has read access to all documents to be indexed by FAST.

```
<parameter name="dm_legal_share_hummingbirduser" comment="Name of a
user that has read access to all Hummingbird documents to be indexed">
  <![CDATA[FASTUSER]]>
</parameter>
<parameter name="dm_legal_share_hummingbirdpassword" comment="Password
for the user specified in the hummingbirduser parameter">
  <![CDATA[FastPassword]]>
</parameter>
```

11 In the [dm_legal_ora] section of the configuration file, set the "hummingbirdpassword" item to the password of the user specified in the "hummingbirduser" item.

```
<parameter name="dm_legal_share_hummingbirdpassword" comment="Password
for the user specified in the hummingbirduser parameter">
  <![CDATA[FastPassword]]>
</parameter>
```

12 If you want to limit documents indexed through the FAST Data Search based on the user privileges, set the "includeuser" item in the [dm_legal_ora] section of the configuration file to the name of the user in the Hummingbird DM system with the proper privileges.

```
<parameter name="dm_legal_share_includeuser" comment="If using user
privileges to index data enter user name here">
<![CDATA[FASTUSER]]></parameter>
```

Note! In most cases this would be the same as the user ID specified in the "hummingbirduser" item.

Note! The user name is case sensitive when Hummingbird DM runs on the Oracle database.

13 In the [dm_legal_ora] section of the configuration file, set the "domainid" item to the security domain name for this library in the FAST repository. This should be "hum" in most cases.

```
<parameter name="dm_legal_ora_domainid" comment="Domain ID for
security submission">  <![CDATA[hum]]></parameter>
```

14 In the [dm_legal_ora] section of the configuration file, set the "urlbase" item to the base part of the URL of the DM Web server.

```
<parameter name="dm_legal_ora_urlbase" comment="Absolute url for DM
Web Server to retrieve a document">  <![CDATA[http://kareta/CyberDOCS/
Libraries/Default_Library/Common/getdocact.asp]]></parameter>
```

Running the Connector

Production version of the Connector is going to be always configured to run as Windows NT Command Line Application.

Command Line Mode

Full population of the FAST repository (item 1) has to be invoked at least once before starting incremental population (item 2).

1 1. To run the Connector in the command line mode for the full population of the FAST repository use the following command:

```
C:\DOConSetup\bin\DOCon4.exe -fullstart -nDOCon4 "-dC:\DOConSetup\bin"
```

Where:

-fullstart	flag for the full population of the FAST repository
-n	name of the configuration file (without the .xml extension) [Optional]. Default value is "DOConnector.xml".
-d	full path to the connector working directory

Note! NOTE: Before running connector in the full population mode clear the FAST Data Search collection to remove old data.

2 To run the Connector in the command line mode for the incremental population of the FAST repository use the following command:

```
C:\DOConSetup\bin\DOCon4.exe -nDOCon4 "-dC:\DOConSetup\bin"
```

Where:

-n	- name of the configuration file (without the .xml extension) [Optional]. Default value is "DOConnector.xml".
-d	- full path to the connector working directory

Troubleshooting

In this section we will describe main cases of the connector failure and log file entries for those cases.

Database connection issues

Main cause of database errors is wrong configuration of the connect string or wrong permissions for the database user.

Connect string is described in the Setup->[Library] section of this document. Complete description of the connect string syntax and options could be found in Microsoft ADO documentation.

The following entries in the log file are indication of the connect string error:

```
[2003-09-18 21:07:07] ERRORError number: 80040e4d Login failed for
user 'DOCSADM_VZVZVZVZV'.
[2003-09-18 21:07:07] ERRORError code 80040e4d - IDispatch error #3149
Source = Microsoft OLE DB Provider for SQL Server
Description = Login failed for user 'DOCSADM_VZVZVZVZV'.
[2003-09-18 21:07:07] FATALCould not initialize database connection in
WalkSQLLib!
```

This particular error is caused by the wrong user id/password combination for the SQL Server login. In case of other errors the exact description of the error will be logged in the similar way.

When Oracle client detects a connection error it creates a file called "sqlnet.log". Check if this file exists.

FAST Data Search connection issues

In the case of wrong configuration of the FAST Data Search connection the following error is logged to the LOG file:

```
[2003-09-18 20:57:17] FATALCould not initialize connector Api in
WalkSQLLib!
```

Please check configuration parameters for FAST Data Search (FASTHost; FASTCollection; FASTPort) and make sure that FAST Data Search is up and running and accessible from this machine!

Chapter 2

Configuring the Hummingbird Connector

Introduction

This document describes technologies used in the Hummingbird DM Connector for FAST Data Search Engine, architecture and user configurable items of the connector.

Installation and Setup of the connector are described in a separate document entitled "Hummingbird DM Connector Setup.doc"

Technologies

Hummingbird DM Connector for FAST Data Search Engine is developed in Microsoft Visual Studio .NET using C++ language.

Microsoft Active Data Objects (ADO) technology provides a bridge to the Hummingbird DM databases.

Hummingbird DM's native API is used by the Connector to extract documents only.

FAST Connector API Toolkit is utilized to submit data to FAST Data Search and to keep track of the changes in security tables for Hummingbird DM.

Production version of the connector is running as a Windows NT Command Line Application.

Functional Overview

Hummingbird DM Connector for FAST Data Search Engine has two modes of operation:

- Full Data Scan
- Incremental Update

Note! The Scheduler has to run Full Data Scan at least once after the installation and before starting incremental population.

Connector Application is started via command-line command with parameters.

Connector can be stopped during either full or incremental population. After the stop, connector will process only documents that were not processed during the previous run. Note that the Connector works in a 1-minute granularity. So at least in theory some documents might be processed twice - those documents that were changed during that minute.

When the Connector is started it executes the following steps:

1 Reads data from the configuration file.

- Configuration file should be located in the directory specified by the -d parameter in the command line and should have the name specified by the -n parameter with extension ".xml".

  For example if the service is started with the following parameters:

  ```
  -nDOConnectorSample "-dC:\DOConnectorDirectory"
  ```

  the connector will expect configuration file in the following location:
  C:\DOConnectorDirectory\DOConnectorSample.xml

2 Reads the list of Hummingbird DM libraries to process from the "processlibraries" item of the "general" section of the configuration file. Each library will be processed sequentially in the loop described in the steps 3 - 8.

3 In the case of Full Scan all temporary files created by previous runs or by connector API are removed.

4

a In the case of the Full Scan population library is initialized and the following query is executed to retrieve all document profiles from the library:

```
SELECT docnumber, max(convert(bigint, datepart(yyyy,
last_edit_date)) * 100000000 + datepart(mm, last_edit_date)*
1000000 + datepart(dd, last_edit_date)* 10000 + datepart(hh,
```

```
last_edit_time)* 100 + datepart(n, last_edit_time)) as LTFormatted
FROM [docsadm].[profile] WHERE status < 18 AND convert(bigint,
datepart(yyyy, last_edit_date)) * 100000000 + datepart(mm,
last_edit_date)* 1000000 + datepart(dd, last_edit_date)* 10000 +
datepart(hh, last_edit_time)* 100 + datepart(n, last_edit_time) >
190000000000 GROUP BY docnumber ORDER BY LTFormatted
```

If previous run was interrupted and full population was not finished this query is adjusted to extract only unprocessed document profiles.

Note! This query is in MS SQL Server syntax. For the Oracle syntax see the sample configuration file below.

b In the case of incremental population library is initialized and the following query is executed to retrieve all document profiles that were modified from the library:

```
SELECT docnumber, max(convert(bigint, datepart(yyyy, start_date)) *
100000000 + datepart(mm, start_date)* 1000000 + datepart(dd,
start_date)* 10000 + datepart(hh, start_time)* 100 + datepart(n,
start_time)) as LTFormatted FROM [docsadm].[activitylog] WHERE
activity_type in (0,1,2,8,9,10,12,14,20,26,27) AND convert(bigint,
datepart(yyyy, start_date)) * 100000000 + datepart(mm, start_date)*
1000000 + datepart(dd, start_date)* 10000 + datepart(hh,
start_time)* 100 + datepart(n, start_time) > LAST_PROCESSED_TIME
GROUP BY docnumber ORDER BY LTFormatted
```

where: LAST_PROCESSED_TIME is actual time of the editing of the last document processed in the previous run of the connector.

Note! This query is in MS SQL Server syntax. For the Oracle syntax see the sample configuration file below.

5. In the case of incremental population all documents that were removed since last population are removed from the FAST repository. Documents numbers of the documents that were removed are retrieved with the following query:

```
SELECT docnumber, max(convert(bigint, datepart(yyyy, start_date)) *
100000000 + datepart(mm, start_date)* 1000000 + datepart(dd,
start_date)* 10000 + datepart(hh, start_time)* 100 + datepart(n,
start_time)) as LTFormatted FROM [docsadm].[activitylog] WHERE
activity_type = 6 AND convert(bigint, datepart(yyyy, start_date)) *
100000000 + datepart(mm, start_date)* 1000000 + datepart(dd,
start_date)* 10000 + datepart(hh, start_time)* 100 + datepart(n,
start_time) > LAST_PROCESSED_TIME GROUP BY docnumber ORDER BY
LTFormatted
```

where: LAST_PROCESSED_TIME is actual time of the previous run of the connector.

Note! This query is in MS SQL Server syntax. For the Oracle syntax see the sample configuration file below.

6. Metadata, content and Access Control List (ACL) for each document retrieved in step 5 are submitted to FAST.

7. All documents for which security settings were changed since the last run are processed.

8. In this step library processing is completed. If there are more libraries to process, connector returns to step 3.

Configuration Guide

For description of the items of the configuration file that apply to the connector setup, please consult "Hummingbird DM Connector Setup.doc"

Extraction and mapping of the Hummingbird DM metadata

Hummingbird DM system is in many ways configurable by administrator, so one of the main goals for the connector is ability to adjust to the specific configuration of the Hummingbird DM repository.

Connector can be configured to extract any metadata from the Hummingbird DM database. Extracted metadata can be mapped to any FDS index fields.

To achieve configuration flexibility connector can define multiple SQL queries to retrieve metadata from the Hummingbird DM database and mapping of the retrieved data into the FAST index profile fields.

Each query is defined in the separate section of the configuration file. Such section has to define:

1 The SQL query itself.

2 The list of database fields retrieved by the query

3 The corresponding FAST index profile fields

During the run time the query defined in 1. will be appended with the document number. For example, query defined as:

```
SELECT docname FROM Profiles WHERE docnumber =
```

Will be executed during runtime for each document as:

```
SELECT docname FROM Profiles WHERE docnumber = n
```

Where: n - is the document number of the current document.

Sample configuration file defines several queries for most common Hummingbird DM customizations: legal, government and financial.

Default metadata

The default fields that are always extracted and their corresponding FDS index fields are:

Hummigbird DM metadata	FDS index field
Library	databaseid
Servername	servername
Version & subversion	version
Document URI	fastid
Document content	data

Where:

Library - is the content of the LibName item in the library section of the configuration file. This item is used to map library name in the FAST repository.

Servername - is the content of the UniqueID item in the "General" section of the configuration file. This item is used to map current Hummingbird DM installation in the FAST repository.

Version & subversion - is the concatenation of the version and subversion strings for the current document.

Document URI - has the following format:

```
http://userver2k/CyberDOCS/Libraries/Default_Library/Common/
getdocact.asp?DOuid=servername&lib=databaseid&doc=N
```

- this string includes all the information needed to uniquely identify the document.

Document content - is the content of the document.

Optional metadata

User can specify any additional metadata to be extracted from the Hummingbird DM database and inserted into the FAST repository. This could be accomplished by defining additional SQL query(s) and index field name(s) in the configuration profile.

To execute additional query user has to define the section name for this query in the "Queries" item of the library section of the configuration file. Several queries can be defined and they have to be separated by '|'. For example, to define two additional metadata queries user would add the following item to the library section of the configuration file:

```
<parameter name="dm_legal_ora_queries">
<!--
   List of query sections for DM metadata. Each bar-separated item is a
name of a section.
  Each section should contain three items - dofields, sql and fastfields.
  - dofields is a bar-separated list of DM metadata fields.
  - sql is the SQL statement for the extraction of these fields.
  - fastfields is a bar-separated list of the corresponding FAST fields
   -->

  <![CDATA[q1|q2]]>

</parameter>
```

For each query defined in the "queries" item user has to define a section in the configuration file. In our example user would have to define two sections **-** [q1] and [q2].

Each query definition section consists of three items:

1 SQL statement to extract data from the database.

2 List of names of the FAST profile to store extracted data

3 List of names in the SQL statement corresponding to the FAST profile names.

Document number will be appended to the SQL statement to form the actual query.

In our example user will have to add following sections:

1

```
<parameter name="q1_dofields" comment="List of field names in the DOCSOpen
query (see next item)">
  <![CDATA[docname|Descr|Author|app]]>
</parameter>
<parameter name="q1_sql" comment="SQL query">
  <![CDATA[
  select a.docname, b.description descr, c.full_name author,
  d.description app
  from docsadm.profile a, docsadm.documenttypes b, docsadm.people c,
docsadm.apps d
  where a.documenttype = b.system_id
  and a.author = c.system_id
  and a.application = d.system_id
  and a.docnumber =
  ]]>
</parameter>
<parameter name="q1_fastfields" comment="List of field names in the FAST
index delimited with '|'">
  <![CDATA[title|documenttype|author|application]]>
</parameter>
```

2

```
<parameter name="q2_dofields" comment="Legal database type with CLIENT/
MATTER tables">
  <![CDATA[matter|client]]>
</parameter>
<parameter name="q2_sql" comment="SQL query">
  <![CDATA[
  select b.Matter_name matter, c.client_name client
  from docsadm.profile a, docsadm.matter b, docsadm.client c
  where a.matter = b.system_id
  and b.client_id = c.system_id
  and a.docnumber =
  ]]>
</parameter>
<parameter name="q2_fastfields" comment="List of field names in the FAST
index delimited with '|'">
  <![CDATA[matter|client]]>
</parameter>
```

Query described in section [q1] will extract Document Name; Document Description; Document Author and Application Name and store them in the "docname", "document-type", "author" and "application" fields of the FAST index profile.

Query described in section [q2] will extract Matter Name and Client Name and store them in the "matter" and "client" fields of the FAST index profile.

Specifying Database Connection String

The database connection string is a concatenation of two items - connectstring and connecttail. The latter is not included in any logs and should be used to specify a password.

The connection string should be an OLEDB connection string. Please refer to your OLEDB provider documentation for specific parameters. The connector has been tested with SQLOLEDB provider for MS SQL Server 2000 and with MSDAORA provider for Oracle 8i.

Specifying User ID for access to Hummingbird

The connector needs a user ID and a password for a Hummingbird user that has read access to all documents to be indexed by FAST. The user ID is specified in the "hummingbirduser" item, while the password is specified in the "hummingbirdpassword" item.

The pool of documents indexed by FAST can be further limited by specifying the "includeuser" item described in the next section. However in most cases the user ID specified in "hummingbirduser" and "includeuser" items should be the same.

Including Documents Based on User Privileges

Administrator can specify that documents should be included for indexing based on the user privileges.

Including documents could be accomplished by adding the "includeuser" item to the configuration file.

If you want to limit documents indexed through the FAST Data Search based on the user privileges, set the "includeuser" item in the library section of the configuration file to the name of the user in the Hummingbird DM system with the proper privileges.

```
<parameter name="dm_legal_ora_includeuser" comment="If using user
privileges to index data enter user name here">
  <![CDATA[FASTUSER]]>
</parameter>
```

Note! In most cases the "includeuser" item should be the same as the "hummingbirduser" item.

Excluding Documents from Indexing

User can specify that some documents have to be excluded from indexing.

Excluding documents could be accomplished by defining additional SQL query(s) in the configuration profile.

To exclude documents from indexing user has to define the section name for each query in the "exclude" item of the library section of the configuration file. Several queries can be defined and they have to be separated by '|'. For example, to define two exclude queries user would add the following item to the library section of the configuration file:

```
<parameter name="dm_legal_ora_exclude">
<!--
  List of query sections for deletion from (or non inclusion into) the
index.
  Each bar-separated item is a name of a section.
  Each section should contain three items - dofields, sql and values.
  - dofields is a bar-separated list of DM metadata fields to check.
  - sql is the SQL statement for the extraction of these fields.
  - values is a bar-separated list of the corresponding values.
  Document is excluded if value of at least one of the "dofields" matches
  the corresponding value in "values".
-->

  <![CDATA[E3|E4]]>

</parameter>
```

For each query defined in the Exclude item user has to define a section in the configuration file. In our example user would have to define two sections - [e1] and [e2].

Each query definition section consists of three items:

1 SQL statement to extract data from the database.

2 Name of the field that is extracted by this query.

3 Value of the field described in 2. that will trigger exclusion from indexing.

Document number will be appended to the SQL statement to form the actual query.

In our example the user will have to add the following sections:

1

```
<parameter name="e1_dofields" comment=" ">
  <![CDATA[author]]>
</parameter>
<parameter name="e1_sql" comment=" ">
  <![CDATA[select c.full_name author from docsadm.[profile] a join
docsadm.people c on author = c.system_id where docnumber =  ]]>
</parameter>
<parameter name="e1_values" comment=" ">
  <![CDATA[JohnDow]]>
</parameter>
```

2

```
<parameter name="e2_dofields" comment=" ">
```

```
  <![CDATA[dn]]>
</parameter>
<parameter name="e2_sql" comment=" ">
  <![CDATA[select docnumber dn from docsadm.profile where docnumber = ]]>
</parameter>
<parameter name="e2_values" comment=" ">
  <![CDATA[5]]>
</parameter>
```

Query described in section [q1] will exclude all documents authored by "JohnDow" from indexing in FAST Data Search engine.

Query described in section [q2] will exclude just document #5 from indexing in the FAST Data Search engine.

Note! After you change exclude settings in the configuration file, you have to run the connector in the full data scan mode once.

Error and Status logging

Hummingbird DM connector logging is controlled by a "messagelevel" item of the "General" section of the configuration file.

Messagelevel could be set to any value from 0 to 8 - 0 being the least number of logging messages and 8 being the most. High messagelevel values should be used sparingly for debugging and investigation of specific issues.

Logging is controlled by standard FAST logging configuration set in the item "logconfig" of the "general" section of the configuration file.

```
<parameter name="general_logconfig" comment="Whatever logger needs for
configuration of the LogConfig item">
  <![CDATA[logconfig.xml:debug]]>
</parameter>
```

Database Queries

In this section we will describe database queries used in the Hummingbird DM connector.

All queries in this section are in MS SQL Server syntax. For the Oracle syntax see the sample configuration file below.

Full Data Scan Query

Document profiles for all documents in the Hummingbird DM database are stored in the Profiles table. To retrieve all document numbers from the database for the first time connector uses the following query:

```
SELECT docnumber, max(convert(bigint, datepart(yyyy, last_edit_date)) *
100000000 + datepart(mm, last_edit_date)* 1000000 + datepart(dd,
last_edit_date)* 10000 + datepart(hh, last_edit_time)* 100 + datepart(n,
last_edit_time)) as LTFormatted

FROM [docsadm].[profile]

WHERE status < 18 AND convert(bigint, datepart(yyyy, last_edit_date)) *
100000000 + datepart(mm, last_edit_date)* 1000000 + datepart(dd,
last_edit_date)* 10000 + datepart(hh, last_edit_time)* 100 + datepart(n,
last_edit_time) > LAST_PROCESSED_DATE GROUP BY docnumber

ORDER BY LTFormatted
```

Where: LAST_PROCESSED_DATE is date and time of the last processed document at the time of the start of the connector service.

Date and time for the events are stored in separate fields in the Hummingbird DM database. To get around this issue we use the following expression to format the DATE_TIME items into the human readable format that is also easy for compares:

```
convert(bigint, datepart(yyyy, last_edit_date)) * 100000000 + datepart(mm,
last_edit_date)* 1000000 + datepart(dd, last_edit_date)* 10000 +
datepart(hh, last_edit_time)* 100 + datepart(n, last_edit_time) as
LTFormatted
```

LTFormatted in this case will be returned in the form YYYYMMDDHHMM

Where:

YYYY	is 4 digit year
MM	is 2 digit month
DD	is 2 digit day of the month
HH	is 2 digit hour
MM	is 2 digit mnute

Incremental Update Query

Every event on the existing document in the Hummingbird DM system is logged into the "ActivityLog" table. This allows connector to use ActivityLog table as source for the information for the Incremental Update.

The following query is used to retrieve all document numbers that were modified since the LAST_PROCESSED_DATE (see previous section for description).

```
SELECT docnumber, max(convert(bigint, datepart(yyyy, start_date)) *
100000000 + datepart(mm, start_date)* 1000000 + datepart(dd, start_date)*
10000 + datepart(hh, start_time)* 100 + datepart(n, start_time)) as
LTFormatted

FROM [docsadm].[activitylog]

WHERE activity_type in (0,1,2,8,9,10,12,14,20,26,27) AND convert(bigint,
datepart(yyyy, start_date)) * 100000000 + datepart(mm, start_date)*
1000000 + datepart(dd, start_date)* 10000 + datepart(hh, start_time)* 100
+ datepart(n, start_time) > LAST_PROCESSED_DATE

GROUP BY docnumber
ORDER BY LTFormatted
```

Where: LAST_PROCESSED_DATE - see description in the previous section

LTFormatted - see description in the previous section

Activity_type - type of the activity performed on the document. This list could be edited by administrator to adjust to the specific needs. Description of the activity_type values follows:

0 - user defined;

1 - create;

2 - edit;

6 - delete content;

8 - new version;

9 - delete version;

10 - import;

12 - check in;

14 - receive;

20 - edit profile;

26 - restore;

27 - annotate.

Getting the List of the Documents Removed From the Database

There are two ways to remove documents from the database. Only one is recorded in the ActivityLog table: remove content. The activity_type for this event is 6 and the query that retrieves the document numbers affected is:

```
SELECT docnumber, max(convert(bigint, datepart(yyyy, start_date)) *
100000000 + datepart(mm, start_date)* 1000000 + datepart(dd, start_date)*
10000 + datepart(hh, start_time)* 100 + datepart(n, start_time)) as
LTFormatted

FROM [docsadm].[activitylog]

WHERE activity_type = 6 AND convert(bigint, datepart(yyyy, start_date)) *
100000000 + datepart(mm, start_date)* 1000000 + datepart(dd, start_date)*
10000 + datepart(hh, start_time)* 100 + datepart(n, start_time) >
LAST_PROCESSED_DATE

GROUP BY docnumber
ORDER BY LTFormatted
```

Second type of delete documents from Hummingbird DM databases is handled through the connector API and is based on the algorithm described in section "Tracking Changes in Security Table".

Check If Document Content Needs Reindexing

Not all selected documents need reindexing during the incremental update. For example, there are documents where only the profile metadata was changed since last update. For these documents content will not be reindexed.

Connector will retrieve the last edit date and time for the document in question and compare it to the date and time of the previous run.

```
SELECT convert(bigint, datepart(yyyy, last_edit_date)) * 100000000 +
datepart(mm, last_edit_date)* 1000000 + datepart(dd, last_edit_date)*
10000 + datepart(hh, last_edit_time)* 100 + datepart(n, last_edit_time) as
getter

FROM [docsadm].[profile]

WHERE docnumber = N
```

Where: N - is the number of the document being processed.

Check If Document Exists in the Database

To check if the document exists in the Hummingbird DM database the connector uses the following query.

```
SELECT docnumber FROM [docsadm].[profile] WHERE docnumber = N
```

Selection of the ID for the Current Version of the Document

The following query selects the version ID for the latest version of the document.

```
      SELECT MAX(version_id) LastVersion
      FROM docsadm.versions v
      WHERE  docnumber =
```

Latest Version Information Retrieval

Version information is stored in the "Versions" table of the Hummingbird DM database. This information is split between two fields - "Version" and "Subversion". For the purpose of document extraction we treat concatenation of the two digit latest version with one character latest subversion as latest document version.

To retrieve the latest version of the document we use the following query:

```
SELECT substring(max(convert(char(3), v.version + 100) + v.subversion), 2,
3) getter
FROM docsadm.[versions] v
WHERE v.docnumber = N
```

Where: N - is document number for which we are getting the latest version.

In this query the statement:

```
substring(max(convert(char(3), v.version + 100) + v.subversion), 2, 3)
```

will extract the concatenation of two digit version number and one character subversion of the latest version of the document number N.

Document ACL list retrieval

Access rights list for each document is stored in the "Security" table one record per user or group per document. ACL lists are split into 4 categories:

1 List of users that have read access to the document.

2 List of groups that have read access to the document.

3 List of users that do not have read access to the document.

4 List of groups that do not have read access to the document.

Since "view document" permission is represented by value 4 and "edit document" permission is represented by value 16 any user or group that has any of these permissions is allowed to view the document and any user or group that does not have any of these permissions will not be allowed to view the document.

In the queries 1. and 2. we join on the people table to retrieve only users with appropriate permissions and in the queries 3. and 4. we join on the groups table to retrieve only groups with appropriate permissions.

1 The query to get all user IDs with read access to the document number N

```
SELECT personorgroup v1
FROM Security s
JOIN People p on p.system_id = s.personorgroup
JOIN Profile pr on thing = pr.system_id
WHERE (accessrights & 20) > 0  and pr.docnumber = N
```

2 The query to get all user IDs with read access to the document number N denied.

```
SELECT personorgroup v1
FROM Security s
JOIN People p on p.system_id = s.personorgroup
JOIN Profile pr on thing = pr.system_id
WHERE (accessrights & 20) = 0  and pr.docnumber = N
```

3 The query to get all group IDs with read access to the doc.

```
SELECT personorgroup v1
FROM Security s
JOIN Groups p on p.system_id = s.personorgroup
JOIN Profile pr on thing = pr.system_id
WHERE (accessrights & 20) > 0  and pr.docnumber = N
```

4 The query to get all group IDs with read access to the doc denied.

```
SELECT personorgroup v1
FROM Security s
JOIN Groups p on p.system_id = s.personorgroup
JOIN Profile pr on thing = pr.system_id
WHERE (accessrights & 20) = 0  and pr.docnumber = N
```

The results of these four queries are combined to populate the FAST "docacl" field.

Tracking Changes in Security Table

Connector API is responsible for tracking of the changes in the Security table of the Hummingbird DM database. To provide connector API with information on document security Hummingbird DM connector uses the following query:

```
SELECT docnumber, personorgroup, accessrights
FROM [docsadm].[security] s
JOIN [docsadm].[profile] p on s.thing = p.system_id
```

Connector API compares current submission with previous one and notifies connector of the documents with changed security values. After receiving notification, connector processes the documents from this list.

Connector API also notifies connector of the records that were removed from the Security table. For documents that were removed altogether connector processes removal from FAST Data Search and for documents that are still available, connector updates security information.

Sample Configuration File

```
<?xml version="1.0"?>
<AccessorConfig>
  <Accessor id="dbextract">
```

<-- general -->

```
    <parameter name="general_processlibraries" comment="list of libraries
to process separated with '|'">
      <![CDATA[DM_Legal_Ora|DM_Legal_Share]]>
    </parameter>
    <parameter name="general_messagelevel" comment="Message level 0 - 8
less to more verbose">
      <![CDATA[8]]>
    </parameter>
    <parameter name="general_logconfig" comment="Whatever logger needs for
configuration of the LogConfig item">
      <![CDATA[logconfig.xml:debug]]>
    </parameter>
    <parameter name="general_netlogdest" comment="Whatever logger needs
for configuration of the NetLogDest item">
      <![CDATA[]]>
    </parameter>
    <parameter name="general_uniqueid" comment="Unique ID of the DOCSOpen
server">
      <![CDATA[kareta_DOCS]]>
    </parameter>
    <parameter name="general_connectorid" comment="Connector ID for
logging">
      <![CDATA[karetaConnector]]>
    </parameter>
    <parameter name="general_errdocsfile" comment="Name of file for the
list of documents that were not processed. If this name is not given a new
file with the name YYYYMMDDHHMM.bad will be created in the same directory
as config every time the connector is started">
      <![CDATA[ErrDocsFile.bad]]>
    </parameter>
    <parameter name="general_batchsize" comment="Size of DSAPI batches">
      <![CDATA[20]]>
    </parameter>
```

<-- SQL Server DB -->

```
    <parameter name="dm_legal_share_domainid" comment="Domain ID for
security submission">
      <![CDATA[hum]]>
    </parameter>
    <parameter name="dm_legal_share_hummingbirduser" comment="Name of a
user that has read access to all Hummingbird documents to be indexed">
      <![CDATA[FASTUSER]]>
    </parameter>
   <parameter name="dm_legal_share_hummingbirdpassword" comment="Password
for the user specified in the hummingbirduser parameter">
      <![CDATA[FastPassword]]>
    </parameter>
    <parameter name="dm_legal_share_includeuser" comment="If using user
privileges to index data enter user name here">
      <![CDATA[FASTUSER]]>
    </parameter>
    <parameter name="dm_legal_share_docprocesslist" comment="Name of file
with the list of documents that were not processed in the previous run and
have to be processed now. Format of each record is LIBRARYNAME|DOCNUMBER
This file is created as ErrDocsFile (see prev item)">
      <![CDATA[dm_legal_share_ProcDocsFile.proc]]>
    </parameter>
    <parameter name="dm_legal_share_connectstring" comment="Start of the
database connect string (included in the log)">
      <![CDATA[Provider=sqloledb;Data Source=kareta;Initial
Catalog=DM_Legal_Share;User Id=FASTDOConnector;Password=]]>
    </parameter>
   <parameter name="dm_legal_share_connecttail" comment="Remainder of the
database connect string (not included in the log)">
      <![CDATA[password]]>
    </parameter>
    <parameter name="dm_legal_share_fasthost" comment="FAST host name or
IP">
      <![CDATA[cosk]]>
    </parameter>
    <parameter name="dm_legal_share_fastcollection" comment="FAST
collection name">
      <![CDATA[DMConTest]]>
    </parameter>
    <parameter name="dm_legal_share_fastport" comment="#FAST port number">
      <![CDATA[16099]]>
    </parameter>
    <parameter name="dm_legal_share_serverindexname" comment="FAST index
name for the unique ID of the DOCSOpen server">
      <![CDATA[servername]]>
    </parameter>
   <parameter name="dm_legal_share_libindexname" comment="FAST index name
for the library name">
      <![CDATA[databaseid]]>
    </parameter>
    <parameter name="dm_legal_share_urlbase" comment="Absolute url for DM
Web Server to retrieve a document">
      <![CDATA[http://kareta/CyberDOCS/Libraries/Default_Library/Common/
getdocact.asp]]>
    </parameter>
   <parameter name="dm_legal_share_libname" comment="FAST index value for
the library">
      <![CDATA[dm_legal_share]]>
    </parameter>
    <parameter name="dm_legal_share_exclude">
    <!--
     List of query sections for deletion from (or non inclusion into) the
index.
      Each bar-separated item is a name of a section.
      Each section should contain three items - dofields, sql and values.
      - dofields is a bar-separated list of DM metadata fields to check.
      - sql is the SQL statement for the extraction of these fields.
      - values is a bar-separated list of the corresponding values.
      Document is excluded if value of at least one of the "dofields"
matches
      the corresponding value in "values".
    -->
      <![CDATA[e3|e4]]>
    </parameter>
    <parameter name="dm_legal_share_queries">
    <!--
     List of query sections for DM metadata. Each bar-separated item is a
name of a section.
      Each section should contain three items - dofields, sql and
fastfields.
      - dofields is a bar-separated list of DM metadata fields.
      - sql is the SQL statement for the extraction of these fields.
     - fastfields is a bar-separated list of the corresponding FAST fields
    -->
    <!--
      <![CDATA[q1|q2]]>
    -->
    </parameter>
```

<-- ORACLE DB -->

```
    <parameter name="dm_legal_ora_domainid" comment="Domain ID for
security submission">
      <![CDATA[hum]]>
    </parameter>
   <parameter name="dm_legal_ora_hummingbirduser" comment="Name of a user
that has read access to all Hummingbird documents to be indexed">
      <![CDATA[FASTUSER]]>
    </parameter>
    <parameter name="dm_legal_ora_hummingbirdpassword" comment="Password
for the user specified in the hummingbirduser parameter">
      <![CDATA[FastPassword]]>
    </parameter>
    <parameter name="dm_legal_ora_includeuser" comment="If using user
privileges to index data enter user name here">
      <![CDATA[FASTUSER]]>
    </parameter>
    <parameter name="dm_legal_ora_docprocesslist" comment="Name of file
with the list of documents that were not processed in the previous run and
have to be processed now. Format of each record is LIBRARYNAME|DOCNUMBER
This file is created as ErrDocsFile (see prev item)">
      <![CDATA[dm_legal_ora_ProcDocsFile.proc]]>
    </parameter>
    <parameter name="dm_legal_ora_connectstring" comment="Start of the
database connect string (included in the log)">
      <![CDATA[Provider=MSDAORA;Data Source=TAN;User
Id=system;Password=]]>
    </parameter>
    <parameter name="dm_legal_ora_connecttail" comment="Remainder of the
database connect string (not included in the log)">
      <![CDATA[manager]]>
    </parameter>
    <parameter name="dm_legal_ora_fasthost" comment="FAST host name or
IP">
      <![CDATA[cosk]]>
    </parameter>
   <parameter name="dm_legal_ora_fastcollection" comment="FAST collection
name">
      <![CDATA[OracleQA]]>
    </parameter>
    <parameter name="dm_legal_ora_fastport" comment="#FAST port number">
      <![CDATA[16099]]>
    </parameter>
    <parameter name="dm_legal_ora_serverindexname" comment="FAST index
name for the unique ID of the DOCSOpen server">
```

```
      <![CDATA[servername]]>
    </parameter>
    <parameter name="dm_legal_ora_libindexname" comment="FAST index name
for the library name">
      <![CDATA[databaseid]]>
    </parameter>
    <parameter name="dm_legal_ora_urlbase" comment="Absolute url for DM
Web Server to retrieve a document">
      <![CDATA[http://kareta/CyberDOCS/Libraries/Default_Library/Common/
getdocact.asp]]>
    </parameter>
    <parameter name="dm_legal_ora_libname" comment="FAST index value for
the library">
      <![CDATA[dm_legal_ora]]>
    </parameter>

    <parameter name="dm_legal_ora_exclude">
    <!--
     List of query sections for deletion from (or non inclusion into) the
index.
      Each bar-separated item is a name of a section.
      Each section should contain three items - dofields, sql and values.
      - dofields is a bar-separated list of DM metadata fields to check.
      - sql is the SQL statement for the extraction of these fields.
      - values is a bar-separated list of the corresponding values.
      Document is excluded if value of at least one of the "dofields"
matches
      the corresponding value in "values".
    -->
    <!--
      <![CDATA[e3|e4]]>
    -->
    </parameter>
    <parameter name="dm_legal_ora_queries">
    <!--
     List of query sections for DM metadata. Each bar-separated item is a
name of a section.
      Each section should contain three items - dofields, sql and
fastfields.
      - dofields is a bar-separated list of DM metadata fields.
      - sql is the SQL statement for the extraction of these fields.
     - fastfields is a bar-separated list of the corresponding FAST fields
    -->
    <!--
      <![CDATA[q1|q2]]>
    -->
    </parameter>
```

<-- Query sections-->

```
    <parameter name="q1_dofields" comment="List of field names in the
DOCSOpen query (see next item)">
      <![CDATA[docname|Descr|Author|app]]>
    </parameter>
    <parameter name="q1_sql" comment="SQL query">
      <![CDATA[
      select a.docname, b.description descr, c.full_name author,
      d.description app
      from docsadm.profile a, docsadm.documenttypes b, docsadm.people c,
docsadm.apps d
      where a.documenttype = b.system_id
      and a.author = c.system_id
      and a.application = d.system_id
      and a.docnumber =
      ]]>
    </parameter>
    <parameter name="q1_fastfields" comment="List of field names in the
FAST index delimited with '|'">
      <![CDATA[title|documenttype|author|application]]>
    </parameter>
    <parameter name="q2_dofields" comment="Legal database type with
CLIENT/MATTER tables">
      <![CDATA[matter|client]]>
    </parameter>
    <parameter name="q2_sql" comment="SQL query">
      <![CDATA[
      select b.Matter_name matter, c.client_name client
      from docsadm.profile a, docsadm.matter b, docsadm.client c
      where a.matter = b.system_id
      and b.client_id = c.system_id
      and a.docnumber =
      ]]>
    </parameter>
    <parameter name="q2_fastfields" comment="List of field names in the
FAST index delimited with '|'">
      <![CDATA[matter|client]]>
    </parameter>
```

<-- Exclude sections-->

```
    <parameter name="e1_dofields" comment=" ">
      <![CDATA[dn]]>
    </parameter>
    <parameter name="e1_sql" comment=" ">
      <![CDATA[select docnumber dn from docsadm.profile where docnumber =
]]>
    </parameter>
    <parameter name="e1_values" comment=" ">
      <![CDATA[4]]>
    </parameter>
    <parameter name="e2_dofields" comment=" ">
      <![CDATA[dn]]>
    </parameter>
    <parameter name="e2_sql" comment=" ">
      <![CDATA[select docnumber dn from docsadm.profile where docnumber =
]]>
    </parameter>
    <parameter name="e2_values" comment=" ">
      <![CDATA[5]]>
    </parameter>
  </Accessor>
</AccessorConfig>
```

Database Configuration Files

All database-specific queries are located in database configuration files. This release of the connector contains two database configuration files - DMMSSQL.xml for MS SQL Server 2000 and DMOracle.xml for Oracle 8i.

DMMSSQL.xml - Database Configuration File for MS SQL 2000

```
<?xml version="1.0"?>
<AccessorConfig>
  <Accessor id="dbextract">
    <!-- Sample configuration for Hummingbird DM running on Oracle -->
    <parameter name="timestampquery" comment="Query to get a timestamp for
current connector run in the proper format">
      <![CDATA[
      SELECT convert(bigint, datepart(yyyy, getdate())) * 100000000 +
      datepart(mm, getdate())* 1000000 + datepart(dd, getdate())* 10000 +
      datepart(hh, getdate())* 100 + datepart(n, getdate())
      as probeTime
      ]]>
    </parameter>
    <parameter name="full" comment="Query string to get all the documents
from the profiles">
      <![CDATA[
      SELECT docnumber, max(
      convert(bigint, datepart(yyyy, last_edit_date)) * 100000000 +
      datepart(mm, last_edit_date)* 1000000 + datepart(dd,
last_edit_date)* 10000 +
      datepart(hh, last_edit_time)* 100 + datepart(n, last_edit_time)
      ) as LTFormatted
      FROM docsadm.profile
      WHERE status < 18 AND
      convert(bigint, datepart(yyyy, last_edit_date)) * 100000000 +
      datepart(mm, last_edit_date)* 1000000 + datepart(dd,
last_edit_date)* 10000 +
      datepart(hh, last_edit_time)* 100 + datepart(n, last_edit_time) >
      ]]>
    </parameter>
    <parameter name="incremental" >
      <![CDATA[
      SELECT docnumber, max(
      convert(bigint, datepart(yyyy, start_date)) * 100000000 +
      datepart(mm, start_date)* 1000000 + datepart(dd, start_date)* 10000
+
      datepart(hh, start_time)* 100 + datepart(n, start_time)
      ) as LTFormatted
      FROM docsadm.activitylog
      WHERE activity_type in (0,1,2,8,9,10,12,14,20,26,27)
      AND
      convert(bigint, datepart(yyyy, start_date)) * 100000000 +
      datepart(mm, start_date)* 1000000 + datepart(dd, start_date)* 10000
+
      datepart(hh, start_time)* 100 + datepart(n, start_time) >
      ]]>
    </parameter>
    <parameter name="incrremove" >
      <![CDATA[
      SELECT docnumber, max(
      convert(bigint, datepart(yyyy, start_date)) * 100000000 +
      datepart(mm, start_date)* 1000000 + datepart(dd, start_date)* 10000
+
      datepart(hh, start_time)* 100 + datepart(n, start_time)
      ) as LTFormatted
      FROM docsadm.activitylog
      WHERE activity_type = 6
      AND
      convert(bigint, datepart(yyyy, start_date)) * 100000000 +
      datepart(mm, start_date)* 1000000 + datepart(dd, start_date)* 10000
+
      datepart(hh, start_time)* 100 + datepart(n, start_time) >
      ]]>
    </parameter>
    <parameter name="lastdocedittime" comment="Check when document was
edited ">
      <![CDATA[
      SELECT
      convert(bigint, datepart(yyyy, last_edit_date)) * 100000000 +
      datepart(mm, last_edit_date)* 1000000 + datepart(dd,
last_edit_date)* 10000 +
      datepart(hh, last_edit_time)* 100 + datepart(n, last_edit_time)
      as getter
      FROM docsadm.profile WHERE docnumber =
      ]]>
    </parameter>
    <parameter name="checkdocexists" comment="Check if document exists and
was not deleted">
        <![CDATA[SELECT docnumber FROM docsadm.profile WHERE docnumber = ]]>
    </parameter>
    <!--
    <parameter name="filepathqry" comment="Selection of the current
version filePath">
        <![CDATA[
        SELECT p.docserver_loc server, (p.path + c.path) path
        FROM docsadm.[profile] p
        JOIN docsadm.versions v ON p.docnumber = v.docnumber
        JOIN docsadm.components c ON v.version_id = c.version_id
        WHERE p.docnumber =
        ]]>
    Connector adds ORDER BY v.version_id DESC
    </parameter>
    -->
    <parameter name="getlastdocversionid" comment="Get the latest version
ID for a given docnumber">
        <![CDATA[
        SELECT MAX(version_id) LastVersion
        FROM docsadm.versions v
        WHERE  docnumber =
        ]]>
    </parameter>
    <parameter name="versionqry" comment="Selection of the current
version+subversion">
        <![CDATA[
        select
        substring(max(convert(char(3), v.version + 100) + v.subversion), 2,
3) getter
        from docsadm.versions v where v.docnumber =
        ]]>
    </parameter>
    <parameter name="useracc" comment="The query to get all user IDs with
read access to the doc.">
        <![CDATA[
        select personorgroup v1
        from docsadm.security s
        join docsadm.people p on p.system_id = s.personorgroup
        join docsadm.profile pr on thing = pr.system_id
        where (accessrights & 20) > 0
        and pr.docnumber =
        ]]>
    </parameter>
    <parameter name="userdeny" comment="The query to get all user IDs with
read access to the doc denied.">
        <![CDATA[
        select personorgroup v1
```

```
      from docsadm.security s
      join docsadm.people p on p.system_id = s.personorgroup
      join docsadm.profile pr on thing = pr.system_id
      where (accessrights & 20) = 0
      and pr.docnumber =
      ]]>
    </parameter>
    <parameter name="groupacc" comment="The query to get all group IDs
with read access to the doc.">
      <![CDATA[
      select personorgroup v1
      from docsadm.security s
      join docsadm.groups p on p.system_id = s.personorgroup
      join docsadm.profile pr on thing = pr.system_id
      where (accessrights & 20) > 0
      and pr.docnumber =
      ]]>
    </parameter>
    <parameter name="groupdeny" comment="The query to get all group IDs
with read access to the doc denied.">
      <![CDATA[
      select personorgroup v1
      from docsadm.security s
      join docsadm.groups p on p.system_id = s.personorgroup
      join docsadm.profile pr on thing = pr.system_id
      where (accessrights & 20) = 0
      and pr.docnumber =
      ]]>
    </parameter>
   <parameter name="securitytab" comment="Select all security records for
documents">
      <![CDATA[
      SELECT docnumber, personorgroup, accessrights
      FROM [docsadm].[security] s JOIN [docsadm].[profile] p on s.thing =
p.system_id
      WHERE thing in
      (SELECT DISTINCT thing from docsadm.security WHERE {userclause} AND
(accessrights & 20) > 0)
      ]]>
    </parameter>
    <parameter name="includedoc" comment="1 if document should be
included">
      <![CDATA[
      SELECT DISTINCT 1 getter
      FROM docsadm.security s
      JOIN docsadm.profile p ON p.system_id = s.thing
      WHERE {userclause} AND accessrights & 20 > 0 AND docnumber =
      ]]>
    </parameter>
    <parameter name="usergroups" comment="Select all groups for specific
user">
      <![CDATA[
      select pg.people_system_id pSID, pg.groups_system_id gSID
      from docsadm.peoplegroups pg
      join docsadm.people p on pg.people_system_id = p.system_id
      where p.user_id = ']]>
    </parameter>
    <parameter name="checklibname" comment="Check library name">
      <![CDATA[
      select LIBRARY_NAME getter
      from docsadm.REMOTE_LIBRARIES
      where DATABASE_NAME = ']]>
    </parameter>
  </Accessor>
</AccessorConfig>
```

DMOracle.xml - Database Configuration File for Oracle 8i

```
<?xml version="1.0"?>
<AccessorConfig>
  <Accessor id="dbextract">
    <!-- Sample configuration for Hummingbird DM running on Oracle -->
    <parameter name="timestampquery" comment="Query to get a timestamp for
current connector run in the proper format">
      <![CDATA[
      SELECT to_number(to_char(sysdate,'YYYYMMDDHH24MI'))
      as probeTime
      from Dual
      ]]>
    </parameter>
    <parameter name="full" comment="Query string to get all the documents
from the profiles">
      <![CDATA[
      SELECT docnumber, max(
      to_number(concat(to_char(last_edit_date,'YYYYMMDD'),
      to_char(last_edit_time,'HH24MI')))
      ) as LTFormatted
      FROM docsadm.profile
      WHERE status < 18 AND
      to_number(concat(to_char(last_edit_date,'YYYYMMDD'),
      to_char(last_edit_time,'HH24MI'))) >
      ]]>
    </parameter>
    <parameter name="incremental" >
      <![CDATA[
      SELECT docnumber, max(
      to_number(concat(to_char(start_date,'YYYYMMDD'),
      to_char(start_time,'HH24MI')))
      ) as LTFormatted
      FROM docsadm.activitylog
      WHERE activity_type in (0,1,2,8,9,10,12,14,20,26,27)
      AND
      to_number(concat(to_char(start_date,'YYYYMMDD'),
      to_char(start_time,'HH24MI'))) >
      ]]>
    </parameter>
    <parameter name="incrremove" >
      <![CDATA[
      SELECT docnumber, max(
      to_number(concat(to_char(start_date,'YYYYMMDD'),
      to_char(start_time,'HH24MI')))
      ) as LTFormatted
      FROM docsadm.activitylog
      WHERE activity_type = 6
      AND
      to_number(concat(to_char(start_date,'YYYYMMDD'),
      to_char(start_time,'HH24MI'))) >
      ]]>
    </parameter>
    <parameter name="lastdocedittime" comment="Check when document was
edited ">
      <![CDATA[
      SELECT
      to_number(concat(to_char(last_edit_date,'YYYYMMDD'),
      to_char(last_edit_time,'HH24MI')))
      as getter
      FROM docsadm.profile WHERE docnumber =
      ]]>
    </parameter>
    <parameter name="checkdocexists" comment="Check if document exists and
was not deleted">
      <![CDATA[SELECT docnumber FROM docsadm.profile WHERE docnumber = ]]>
    </parameter>
    <!--
    <parameter name="filepathqry" comment="Selection of the current
version filePath">
      <![CDATA[
      SELECT p.docserver_loc server, concat(p.path,c.path) path
      FROM docsadm.profile p, docsadm.versions v, docsadm.components c
      WHERE  p.docnumber = v.docnumber
      AND v.version_id = c.version_id
      AND p.docnumber =
      ]]>
    <!-- Connector adds ORDER BY v.version_id DESC
    </parameter>
    -->
    <parameter name="getlastdocversionid" comment="Get the latest version
ID for a given docnumber">
      <![CDATA[
      SELECT MAX(version_id) LastVersion
      FROM docsadm.versions v
      WHERE  docnumber =
      ]]>
    </parameter>
    <parameter name="versionqry" comment="Selection of the current
version+subversion">
      <![CDATA[
      select
      substr(max(concat(to_char(v.version + 100, '009'),
v.subversion)),3,3) getter
      from docsadm.versions v where v.docnumber =
      ]]>
    </parameter>
    <parameter name="useracc" comment="The query to get all user IDs with
read access to the doc.">
      <![CDATA[
      select personorgroup v1
      from docsadm.security s, docsadm.people p, docsadm.profile pr
      where p.system_id = s.personorgroup
      and thing = pr.system_id
      and bitand(accessrights , 20) > 0
      and pr.docnumber =
      ]]>
    </parameter>
    <parameter name="userdeny" comment="The query to get all user IDs with
read access to the doc denied.">
      <![CDATA[
      select personorgroup v1
      from docsadm.security s, docsadm.people p, docsadm.profile pr
      where p.system_id = s.personorgroup
      and thing = pr.system_id
      and bitand(accessrights , 20) = 0
      and pr.docnumber =
      ]]>
    </parameter>
    <parameter name="groupacc" comment="The query to get all group IDs
with read access to the doc.">
      <![CDATA[
      select personorgroup v1
      from docsadm.security s, docsadm.groups p, docsadm.profile pr
      where p.system_id = s.personorgroup
      and thing = pr.system_id
      and bitand(accessrights , 20) > 0
      and pr.docnumber =
      ]]>
    </parameter>
    <parameter name="groupdeny" comment="The query to get all group IDs
with read access to the doc denied.">
      <![CDATA[
      select personorgroup v1
      from docsadm.security s, docsadm.groups p, docsadm.profile pr
      where p.system_id = s.personorgroup
      and thing = pr.system_id
      and bitand(accessrights , 20) = 0
      and pr.docnumber =
      ]]>
    </parameter>
    <parameter name="securitytab" comment="Select all security records for
documents">
      <![CDATA[
      SELECT docnumber, personorgroup, accessrights
      FROM docsadm.security s, docsadm.profile p
      WHERE s.thing = p.system_id AND thing in
      (SELECT DISTINCT thing from docsadm.security WHERE {userclause} AND
bitand(accessrights, 20) > 0)
      ]]>
    </parameter>
    <parameter name="includedoc" comment="1 if document should be
included">
      <![CDATA[
      SELECT DISTINCT 1 getter
      FROM docsadm.security s, docsadm.profile p
      WHERE p.system_id = s.thing AND {userclause} AND
bitand(accessrights, 20) > 0 AND docnumber =
      ]]>
    </parameter>
    <parameter name="usergroups" comment="Select all groups for specific
user">
      <![CDATA[
      select pg.people_system_id pSID, pg.groups_system_id gSID
      from docsadm.peoplegroups pg, docsadm.people p
      where pg.people_system_id = p.system_id
      and p.user_id = ']]>
    </parameter>
    <parameter name="checklibname" comment="Check library name">
      <![CDATA[
      select LIBRARY_NAME getter
      from docsadm.REMOTE_LIBRARIES
      ]]>
      <!-- No DATABASE_NAME in Oracle -->
    </parameter>
  </Accessor>
</AccessorConfig>
```