[MS-FSCDBS]:
Connector Database Schema

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
11/06/2009	0.1	Major	Initial Availability
02/19/2010	1.0	Editorial	Revised and edited the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Connector Database Schema. This specification describes the database schema for the JDBC indexing connector component and the Lotus Notes **indexing connector** component.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

access control list (ACL)
checksum
Coordinated Universal Time (UTC)

The following terms are defined in [MS-OFCGLOS]:

crawl
data source
indexing connector
item
Transact-Structured Query Language (T-SQL)

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MSDN-TSQL-Ref] Microsoft Corporation, "Transact-SQL Reference", http://msdn.microsoft.com/en-us/library/ms189826(SQL.90).aspx

[MS-FSCF] Microsoft Corporation, "Content Feeding Protocol Specification"

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

1.2.2 Informative References

[JDBC] Sun Microsystems, Inc., "The Java Database Connectivity (JDBC) API", JDBC 3.0 API, http://java.sun.com/javase/6/docs/technotes/guides/jdbc/

[LotusNotes] IBM, "Lotus Notes - Business email solution", http://www-01.ibm.com/software/lotus/products/notes/

[MS-FSSAC] Microsoft Corporation, "Search Authorization Connector Protocol Specification"

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

1.3 Structure Overview (Synopsis)

This document describes the database definition used by the indexing connector components.

The JDBC indexing connector component **crawls** the JDBC **data source(1)** and passes the **items** to the Content Feeding Protocol server as described in [MS-FSCF]. The JDBC indexing connector component uses the database to store item status and to detect changes in the JDBC data source(1). A JDBC data source is described in [JDBC].

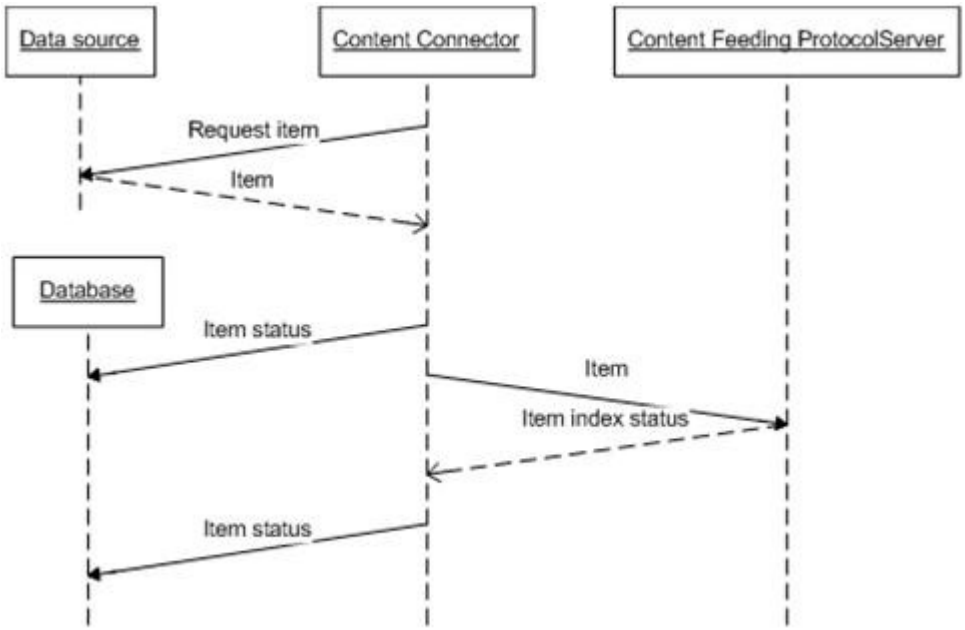

Figure 1: Indexing connector item status flow chart

The Lotus Notes indexing connector component crawls the Lotus Notes data source(1), and passes the items to the Content Feeding Protocol server as described in [MS-FSCF], and the corresponding **access control list (ACL)** information is passed to the Search Authorization Connector Protocol Specification as described in [MS-FSSAC]. The Lotus Notes indexing connector component uses the database to store item status and detect changes in the Lotus Notes data source(1). A Lotus Notes data source is described in [LotusNotes].

1.3.1 Item status

Item status is the status of the retrieved items, and their indexing status. The index status describes whether the item has been sent for indexing, successfully indexed, failed while indexing or has unknown status. The item status is also made available through Microsoft® SQL Server® 2008 Reporting Services.

1.3.2 State tracking

Lotus Notes item status is stored in the state tracking tables, and describes whether items were changed.

1.3.3 Change detection

The JDBC indexing connector component can detect changes by using persisted storage in database tables. It detects change with a **checksum**, which it stores in the database and compares against later checksums.

1.4 Relationship to Protocols and Other Structures

None.

1.5 Applicability Statement

The database schema described in this document can be used as a basis for alternative implementations of indexing connectors.

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

None.

2 Structures

The following figure specifies the database tables for JDBC indexing connector component and Lotus Notes indexing connector component.

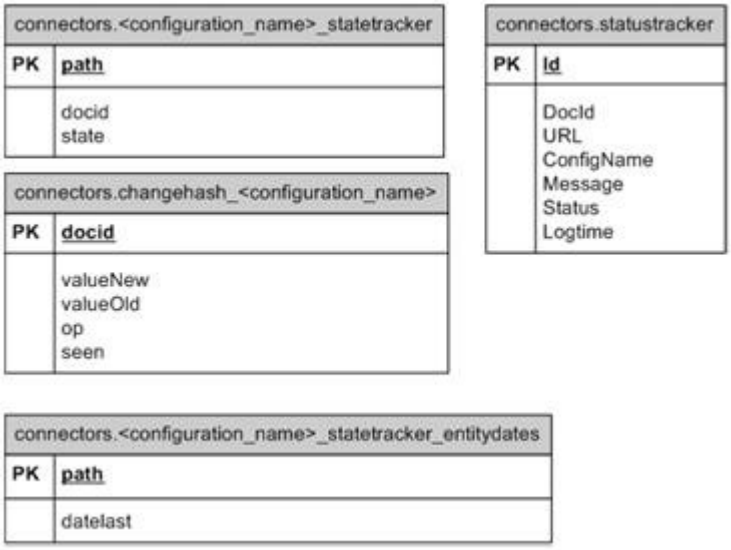

Figure 2: Database tables

The database tables do not contain any foreign keys, stored procedures or triggers. Each item MUST have an associated configuration name that partitions the items into tables. An indexing connector uses the same configuration name for all items coming from the same data source. The configuration name MUST consist of the characters [a-f][0-9], a minimum of 3 characters and a maximum of 32 characters.

2.1 changehash

This table stores the item checksum for change detection for later comparison. If the checksum is changed, the document has been updated. The checksum is an implementation specific hash value computed from the item elements. The table MUST be a composite of the value of *changehash_* and the configuration name. The T-SQL (Transact-Structured Query Language) syntax for the table is as follows. The **T-SQL** definition is specified in [MSDN-TSQL-Ref].

2.1.1 Table definition

The table is contained in a schema called "connectors". The T-SQL syntax for the table is as follows:

```
TABLE connectors.changehash_<configuration name> (docid varchar(255) NOT NULL CONSTRAINT
changehash_<configuration name>pk_docid PRIMARY KEY,valueNew varchar(255) default
NULL,valueOld varchar(255) default NULL,op int default NULL,seen bigint default NULL )
```

docid: The item identifier.

valueNew: Stores a hash value calculated from the item.

valueOld: Stores a temporary hash value calculated from the previous item in case a rollback is required.

op: The operation status. It specifies the item change status in the index node. The column MUST have one of the following values.

Value	Meaning
0	Item added or updated, and successful callback received
1	No update
2	New update
3	Update existing
4	Update, but no change
5	Delete

seen: UTC (Coordinated Universal Time) time in seconds that have elapsed between 1970-01-01T00:00:00 and the time that the item was last associated with the indexing connector.

2.2 statetracker

This table stores state information about Lotus Notes documents and databases. The state is the crawl status and the Content Feeding Protocol, [MS-FSCF], callback information.

2.2.1 Table definition

This is the configuration name appended with the string "_statetracker". The table is contained in a schema called "connectors". The T-SQL syntax for the table is as follows:

```
TABLE connectors.<configuration_name>_statetracker (path varchar(255) NOT NULL CONSTRAINT
<configuration_name>_statetrackerpk_path PRIMARY KEY,docid varchar(255) default NULL,state
int default NULL )
```

path: The item identifier.

docid: The item identifier.

state: The indexing state. The column MUST have one of the following values.

Value	Meaning
0	Stable. Successful callback from Content Feeding Protocol, as specified in [MS-FSCF]
1	Adding
2	Updating
3	Deleting. On successful callback from Content Feeding Protocol feeder, as specified in [MS-FSCF] , the row is deleted

2.3 statetrackerdate

This table stores information about the content that was last crawled. Each row specifies when a database was last crawled.

2.3.1 Table definition

The table MUST be the same as the configuration name and appended with the string constant "_statetracker_entitydates". The table is contained in a schema called "connectors". The T-SQL syntax for the table is as follows:

```
TABLE connectors.<configuration_name>_statetracker_entitydates (path varchar(255) NOT NULL
CONSTRAINT     <configuration_name>_statetracker_entitydatespk_path2 PRIMARY KEY,
datelast varchar(255) default NULL )
```

path: Path to the [LotusNotes] database.

datelast: Last date the database was crawled. The value of the *datelast* variable is stored as **UTC** time in seconds that have elapsed since 1970-01-01T00:00:00.

2.4 statustracker

This table contains information about the item status. The item status MUST be created during crawl, and updated based on the callback by the Content Feeding Protocol, as specified in [MS-FSCF]. Statustracker item status is not required for the product to operate.

2.4.1 Table definition

The table is contained in a schema called "connectors". The T-SQL (Transact-Structured Query Language) syntax for the table is as follows:

```
TABLE connectors.statustracker ( Id bigint NOT NULL IDENTITY,DocId nvarchar(200) default
NULL,URL nvarchar(2000) default '',ConfigName nvarchar(200) default '',Message nvarchar(4000)
default '',Status nvarchar(50) default '',Logtime datetime default NULL )
```

id: A unique identifier assigned to the status tracker.

DocId: The item identifier.

URL: The URL to the item.

ConfigName: The name of the configuration used to extract the item.

Message: Index status message reported by the Indexer node.

Status: Item index status. The column MUST contain one of the following values.

Value	Meaning
Pending - add/update	Item added, and waiting for feedback from index node
Pending – delete	Item deleted, and waiting for feedback from index node
Failed - add/update	Indexer node failed to add or update item
Failed – delete	Indexer node failed to delete item
Success - add/update	Indexer node successfully added or updated item
Success – delete	Indexer node successfully deleted item

Value	Meaning
Failed - unknown operation	Indexer node returned failure for unknown item
Success - unknown operation	Indexer node returned success for unknown item

Logtime: The date and time when the entry was updated.

3 Structure Examples

3.1 Add new pending item to statustracker

Adding or updating an item to an indexer node also adds a new entry to the statustracker database table. The following T-SQL statement creates a new database table row where p0 is the item identifier and p1 is the configuration name.

```
INSERT INTO connectors.statustracker ( DocId, ConfigName, Status, Logtime ) VALUES( @p0, @p1,
'Pending - add/update', CURRENT_TIMESTAMP )
```

3.2 Update pending item status

Updating an item to a status successfully indexed also updates the *statustracker* database table. The following T-SQL statement updates the database row, where p0 is the item identifier of the pending item.

```
UPDATE connectors.statustracker SETStatus = 'Success - add/update',Logtime =
CURRENT_TIMESTAMPWHERE DocId = @p0
```

3.3 Add new item to statetracker database table

Adding a new item also adds a new entry to the *statetracker* database table. The following T-SQL statement creates a new database table row where p0 is the path, and p1 is the item identifier.

```
INSERT INTO connectors.<configuration_name>_statustracker ( path, docid, state ) VALUES( @p0,
@p1, 1 )
```

3.4 Update pending item status

Updating the status of an item to a value of "successfully indexed" will also update the statetracker database table. The following T-SQL statement updates the database table row where p0 is the path, and p1 is the item identifier.

```
UPDATE connectors.<configuration_name>_statustracker SET state = 0 WHERE path=@p0 AND
docid=@p1
```

3.5 Add new entry to statetracker_entitydates database table

Completing a crawl of a Lotus Notes database will also update the statetracker_entitydates database table. The following T-SQL statement creates a new database table row where p0 is the path to the database, and p1 is when the database was last updated.

```
INSERT INTO connectors.<configuration_name>_statustracker_entitydates( path, datelast )
VALUES( @p0, @p1 )
```

4 Security Considerations

Only the user account that hosts the indexing connector component implementation is granted access to the database instance that hosts the database schema specified by this protocol.

5 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® FAST™ Search Server 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

6 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

7 Index