

MetaMatrix Solr Connector

User Documentation

Bank of America

Prepared by: Michael Walker

February 23, 2010

Contents

[1. Confidentiality Clause 4](#_Toc254685613)

[2. Copyright Protection 4](#_Toc254685614)

[3. Document Purpose 5](#_Toc254685615)

[4. Overview 5](#_Toc254685616)

[5. Connector Architecture 5](#_Toc254685617)

[6. Installation Instructions 6](#_Toc254685618)

[6.1 Designer 6](#_Toc254685619)

[6.2 MetaMatrix Server 6](#_Toc254685620)

[7. Data Modeling Instructions 7](#_Toc254685621)

[7.1 Type support 8](#_Toc254685622)

[7.2 Cardinality 9](#_Toc254685623)

[7.3 Example VDB 9](#_Toc254685624)

[8. Querying Solr as a Relational Source 9](#_Toc254685625)

[8.1 SQL Queries 9](#_Toc254685626)

[8.2 Capabilities 9](#_Toc254685627)

[8.2.1 Supported Capabilities 9](#_Toc254685628)

[8.2.2 Non-supported Capabilities 9](#_Toc254685629)

[8.3 Wildcards 10](#_Toc254685630)

[8.4 Reserved Characters 10](#_Toc254685631)

[9. Connector Binding Configuration Options 10](#_Toc254685632)

[9.1 SolrServerURL 10](#_Toc254685633)

[9.2 SoTimeout 10](#_Toc254685634)

[9.3 ConnectionTimeout 10](#_Toc254685635)

[9.4 DefaultMaxConnectionsPerHost 10](#_Toc254685636)

[9.5 AllowCompression 11](#_Toc254685637)

[9.6 MaxRetries 11](#_Toc254685638)

[9.7 LowerCaseSearch 11](#_Toc254685639)

[9.8 Max In Criteria Size 11](#_Toc254685640)

[9.9 Connector Pool Cleaning Interval 11](#_Toc254685641)

[9.10 Connector Pool Live and Unused Time 11](#_Toc254685642)

[9.11 Connector Pool Max Connections 11](#_Toc254685643)

[9.12 Connector Pool Wait For Source Time 11](#_Toc254685644)

[9.13 Result Set Cache Enabled 11](#_Toc254685645)

[9.14 Result Set Cache Maximum Age 11](#_Toc254685646)

[9.15 Result Set Cache Maximum Size 11](#_Toc254685647)

[9.16 Data Source Monitoring Enabled 11](#_Toc254685648)

[10. Performance Tuning Notes 12](#_Toc254685649)

[10.1 Connector Batch Size 12](#_Toc254685650)

[10.2 Compression 12](#_Toc254685651)

[11. Use with Solr as Client 12](#_Toc254685652)

# Confidentiality Clause

The information presented in this proposal is exclusively confidential to Amentra, Inc. In no event shall all or any part of this document be disclosed or disseminated without the expressed written permission of Amentra, Inc.

# Copyright Protection

Copyright © 2010 by Amentra, Inc. of Richmond, Virginia. All rights reserved. No part of the work covered by the copyright herein may be reproduced or used in any form or by any means- graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems- without permission in writing from Amentra.

# Document Purpose

This document provides information on how to install, configure, and use the Solr Connector for MetaMatrix.

# Overview

The Solr Connector provides the ability to incorporate Apache Solr 1.4 as a data source in MetaMatrix. Solr indexes can be represented as if they were relational tables in a database. The connector translates SQL queries into the appropriate Solr search, fetches the results, and returns them in a ResultSet format.

The benefits of the connector include the following:

* Data from Solr indexes can be federated with other data sources
* Logical views and Data Services based Solr indexes as a source can be created
* Third-party tools that use standard SQL (e.g. reporting tools, profiling tools, JDBC/ODBC query tools, Solr, etc) can now query Solr Indexes via MetaMatrix without any customization
* Multiple, distributed Solr indexes can be queried simultaneously and their results can be combined in Metamatrix

The connector was developed and tested against Solr 1.4 and its associated libraries, and supports Solr 1.4.

Detailed design documentation for the connector can be found in the document “MMX Solr Connector Design Document.docx”.

# Connector Architecture



Figure Connector Logical Architecture

Clients will send queries to a VDB that has been bound to a Solr Connector. The Solr Connector will receive SQL queries and translate them into Solr Search statements. The search statements will be executed against a Solr server using the SolrJ API. The data will be returned in binary format (compressed, if the appropriate binding option is used), and transformed into a result set. This result set will be returned to the MetaMatrix federated query engine, where it will be potentially further transformed and/or combined with other data, and finally returned back to the client.

# Installation Instructions

## Designer

To install the connector in MetaMatrix Designer 5.5.4, open Designer and click the “Import Connectors (.cdk, .caf)” button, highlighted in red in the illustration below:



Figure Import Connector to Designer

Select the SolrConnector-***Designer***.caf file, and click Finish to complete the installation.

**IMPORTANT** You must select the SolrConnector-***Designer***.caf file, and not the SolrConnector.caf file. This is because Designer runs in a 1.5 JVM and requires a different connector than the one designed for the Server’s 1.6 JVM.

The connector type is now imported into MMX Designer. You can now create connector bindings of type SolrConnector.

## MetaMatrix Server

To install the Solr Connector in the MetaMatrix 5.5.4 Server, open Console and go to Configuration > Integration Server > Connector Types. Select “Import”, and select SolrConnector.caf:



Figure Import Connector to Server

The connector type is now imported into MMX Server. You can now create bindings of type SolrConnector.

# Data Modeling Instructions

The Solr Index is represented in MetaMatrix using the Physical Relational metamodel type. Since the Solr Connector is custom, no importer exists to automatically import Solr Index metadata to build the model in MetaMatrix. Therefore, the model can be created by hand.

To model a Solr index to be used in a VDB:

1. Create a new Physical Relational model.
2. Create a table to represent the index.
3. Create columns to represent each field.

Each column in the table will correspond to a field in the index. On each column, the following properties must be set:

|  |  |  |
| --- | --- | --- |
| Property Name | Purpose | Example |
| NameInSource | Field Name | Id |
| Type | Type corresponding to MMX built-in types that match the type used in Solr | String |

For information on which types are supported by the connector, see section 6.1.

Note that the name is determined by NameInSource, not the name itself. This allows special characters to be used in the field name, even if they are not valid for column names. For example, “$uid$” might be a field name in Solr, but this is not a valid ANSI SQL column name since it starts with “$”. By setting NameInSource to “$uid$”, the field will be correctly accessed. The column name can be any valid ANSI SQL column name, e.g. “uid”.

The illustration below shows a table “RepoRefreshIndex” that models an index with two exposed fields.



Figure Physical Model Example

From this point onward, the table can be treated as another relational source, so it can be joined with other data, queried using SQL, etc.

## Type support

The following types are supported:

* String
* Date
* Integer

## Cardinality

By setting cardinality on each table that represents an Index, MetaMatrix will use cost-based query planning. This can result in dependent join plans that produce better query performance.

Consider adding cardinality to each model representing the approximate number of documents in the index. Examine the query plan and performance profiles to determine the best fit.

## Example VDB

See the MDEX project example models and VDB for further examples of modeling Solr data sources.

# Querying Solr as a Relational Source

## SQL Queries

In general, any valid ANSL SQL query can be sent to MetaMatrix, and the query will be evaluated against Solr as if it were a relational source.

## Capabilities

Since Solr is not a relational database, the Solr Connector translates SQL queries to appropriate Solr searches. Some SQL does not have a matching equivalent in Solr, or was considered out-of-scope for the development effort. However, any valid SQL statement can be issued against a Solr Connector-bound data source.

MetaMatrix will use the connector’s *capabilities* to determine what SQL can be processed by the Solr Server. This is known as “query pushdown”, because the query is being pushed down to the source. SQL that is not supported by these capabilities will be performed by MetaMatrix in a post-processing step.

It is important to note that pushdown results in the best performance, since the work is performed at the data source, and only the minimum amount of data is retrieved. If a query statement is not supported by the connector capabilities, then the performance may be impacted, since more data may need to be fetched and processed in MetaMatrix. Therefore, it is useful to know what capabilities are supported by the connector, and what will be processed by MetaMatrix.

### Supported Capabilities

* WHERE Clauses
* Equals Criteria
* Compound Criteria:
  + AND
  + OR
  + NOT
  + LIKE (%, \_ wildcards)
* LIMIT, OFFSET

### Non-supported Capabilities

* JOIN
* BETWEEN
* Aggregations (DISTINCT, SUM, COUNT, MAX, MIN, etc)
* Functions
* ORDER BY
* GROUP BY
* Greater Than, Less than, Not Equal To

If the non-supported capabilities are used in queries, it is advisable to view the query plan and examine the performance, etc., to ensure the MMX system is not overtaxed.

## Wildcards

The Solr wildcards (e.g. “\*”, “?”) can be included in literals, and they will be passed into Solr. For example:

SELECT fieldA from index WHERE fieldB=’my\*string’

This query will pass the wildcard “\*” to Solr and create the following search:

fieldB:my\*string

## Reserved Characters

To search for reserved characters, the characters should be double-quoted, e.g.:

SELECT fieldA from index WHERE fieldB=’my\”\*\”string’

Will return documents where fieldB is set to the string literal “my\*string”. However:

SELECT fieldA from index WHERE fieldB=’my\*string’

Will return documents where fieldB is set to “mystring”, “my123string”, etc.

Documentation on reserved characters can be found at:

<http://lucene.apache.org/java/2_3_2/queryparsersyntax.html#Escaping%20Special%20Characters>

The current set of reserved characters are:

+ - && || ! ( ) { } [ ] ^ " ~ \* ? : \

# Connector Binding Configuration Options

## SolrServerURL (REQUIRED)

URL of the SolrServer, e.g.:

http://localhost:8080/solr/search

## SoTimeout

Sets SolrServer.setSoTimeout(Integer).

Defaults to NULL (uses Solr default).

## ConnectionTimeout

Sets SolrServer.setConnectionTimeout(Integer).

Defaults to NULL (uses Solr default).

## DefaultMaxConnectionsPerHost

Sets SolrServer.setMaxTotalConnections(Integer).

Defaults to NULL (uses Solr default).

## AllowCompression

Sets SolrServer.setAllowCompression(boolean).

Defaults to TRUE.

## MaxRetries

Sets SolrServer.setMaxRetries(Integer).

Defaults to NULL (uses Solr default).

## LowerCaseSearch

Automatically translates all string search criteria to lower-case, to accommodate for Solr filter built on lower-case values.

Defaults to TRUE.

## Max In Criteria Size

Determines the maximum number of clauses to use in a single IN criteria statement. MetaMatrix will break up queries that exceed this limit into multiple queries.

## Connector Pool Cleaning Interval

Interval of time before the pool is checked for connections that have been alive and unused for longer than the Connector Pool Live and Unused Time.

## Connector Pool Live and Unused Time

Amount of time before a connection in the pool is considered closable.

## Connector Pool Max Connections

Maximum number of connections in the pool. If a connection is requested and the pool is at a maximum, the Connector Pool Wait For Source Time will determine how long the thread will wait until an error is thrown.

## Connector Pool Wait For Source Time

Determines how long a thread will wait for a connection (e.g. if the connection pool is temporarily full).

## Result Set Cache Enabled

Activates result set caching.

Default=FALSE

## Result Set Cache Maximum Age

Maximum age, in milliseconds, of data in the result set cache before it can be considered stale.

## Result Set Cache Maximum Size

Maximum size of the result cache before data is evicted.

## Data Source Monitoring Enabled

Provides status information about the source connection when enabled.

# Performance Tuning Notes

## Connector Batch Size

The connector batch size determines how many documents will be fetched per each call to the SolrServer. It corresponds to the SolrQuery.setRows(int) method in SolrJ. The batch size can have a large influence on performance.

An average batch size of 2000 rows is recommended for smaller queries (e.g. queries returning between 1-100,000 rows of data). This is the default setting.

A larger batch size can be used when pulling very large data sets from Solr (e.g. 1M rows or more). Increasing the batch size will also increase the RAM requirements for the MetaMatrix server, since it will effectively increase the number of records that will be stored in memory by the SolrJ Client libraries. So, be sure to monitor RAM usage and experiment with the batch size to find an optimal setting. In lab testing, we found that a batch size of 200,000 was possible, and improved performance for a data fetch of 2M rows by a four-fold factor over the default size of 2000 rows.

To set the batch size, go to Console > Configuration > System Properties > buffer and choose “All Properties” to display Connector Batch Size (rows). The server must be restarted for the change to take effect.

## Compression

The connector, and corresponding Solr libraries, supports the ability to compress data passed from the Solr Server to the connector. Compression may yield benefits when the amount of data is large, or when the network bandwidth is poor. Compression may detract from performance when the data size is small and/or network bandwith is high, since it adds to the time required to prepare the data. It is advistable to adjust the settings according to each environment as part of a performance tuning step.

# Use with Solr as Client

During testing, it was observed that the Solr Data Source Import Handler attempts to establish a DataSource connection over JDBC that requires support for XATransactions. The SolrConnector does not support XATransactions (as it is read-only) and this can result in an error.

To workaround this issue, be sure to use “autoCommit=true” option when configuring the import handler, e.g.:

<dataSource driver="com.metamatrix.jdbc.MMDriver" autoCommit="true"

url="jdbc:metamatrix:test@mm://localhost:31000;user=abc;password=123;txnAutoWrap=OFF" />