RDB via Table How to Configure

DRAFT FOR REVIEW

This tutorial walks through the steps necessary to crawl an RDBMS repository using the Aspire RDB via Table connector.

On this page

Before Beginning: Create User Account

A prerequisite for crawling any RDBMS is to have an RDBMS account. The recommended name for this account is "aspire_crawl_account" or something similar.

The username and password for this account will be required below.

Step 1: Set RDBMS Access Rights

The "aspire_crawl_account" will need to have sufficient access rights to read all of the documents in the RDBMS that you wish to crawl.

To set the rights for your "aspire_crawl_account", do the following:

1. Log into the RDBMS as an Administrator.
2. Make the role of the "aspire_crawl_account" either administrator or superuser (so that it has access to all RDBMS content).

You will need this login information later in these procedures, when entering properties for your RDB Connector via Table.

Step 2: Launch Aspire and open the Content Source Management Page

Launch Aspire (if it's not already running).

See Launching Aspire

Browse to: http://localhost:50505. For details on using the Aspire Content Source Management page, please refer to UI Introduction.

Step 3: Install and Configure the RDB Connector via Table Content Source into Aspire

To specify exactly what RDBMS to crawl, we will need to create a new "Content Source".

To create a new content source:

1. From the Aspire 2 Home page, click on "Add Source" button.
2. Click on "'RDB Connector via Table'".

Step 3a: Specify Basic Information

In the "General" tab in the Add New Content Source window, specify basic information for the content source:

1. Enter a content source name in the "Name" field.
2. This is any useful name which you decide is a good name for the source. It will be displayed in the content source page, in error messages, etc.
3. Click on the "Active?" checkbox to add a checkmark.
4. Unchecking the "Active?" option allows you to configure content sources but not have them enabled. This is useful if the folder will be under maintenance and no crawls are wanted during that period of time.
5. Click on the "Schedule" drop-down list and select one of the following: "Manually," "Periodically", "Daily," or "Weekly."
6. Aspire can automatically schedule content sources to be crawled on a set schedule, such as once a day, several times a week, or periodically (every N minutes or hours).

7. For the purposes of this tutorial, you may want to select "Manually" and then set up a regular crawling schedule later.
8. After selecting a "Schedule" type, specify the details, if applicable:
    - "Manually": No additional options.
    - "Periodically": Specify the "Run every:" options by entering the number of "hours" and "minutes."
    - "Daily": Specify the "Start time:" by clicking on the hours and minutes drop-down lists and selecting options.
    - "Weekly": Specify the "Start time:" by clicking on the hours and minutes drop-down lists and selecting options, then clicking on the day checkboxes to specify days of the week to run the crawl.
    - "Advance": Enter a custom CRON Expression (e.g. 0 0 0 ? * *)

Step 3b: Specify RDBMS Properties & Connection Details

In the "Connector" tab, specify the connection information to crawl the RDBMS

Starting Point

1. In the "JDBC Url" field in the Properties section, enter the JDBC URL for the database to be crawled.
2. Specify the username and password of the crawl account you created earlier.
    - It needs sufficient access to crawl the RDBMS documents and folders in the path that you specified.
      Note: *The password will be automatically encrypted by Aspire.*
3. In the "JDBC Driver Jar:" field, enter the name of the JAR file containing the driver for your database.
4. In the "JDBC Driver Class" field, enter the Java class name for the driver (optional)
5. If your JDBC Driver jar does not contain all the necessary classes, you may add additional jar files in to the class path. To do this, select "Specify classpath" and enter the classpath to the jars containing the classes. The classpath may include jar files, directories or individual classes.

SQL Configuration

The connector uses a number of SQL statements to define what is exacted from the database when a crawl is run. In full mode, a single statement is used to extract data. In incremental mode, a number of statements are used. When data is extracted from the database, that data is put in to Aspire by column name. If you want is to appear by another name, use the SQL "as" operator in your select statements.

For the purposes of this tutorial, you'll need to understand the schema of your database and have access to a third party SQL client that will allow you to run SQL statements.

See here for further details on configuring SQL for crawls

Full Crawl SQL

The full crawl SQL statement is executed once when the "Full" button is pressed. It should extract all the data you wish to be submitted to Aspire and can extract from one or more tables. In it's simplest form, it may look something like:

```
SELECT
   id,
   col1,
   col2,
   col3,
   col4
 FROM
   main_data
```

This will result in an Aspire Job for each row returned, each comprising a document which hold fields named "id", "col1", "col2", "col3" and "col4"

Use Slices for Full Sql

If you wish, when you perform a full crawl, the connector will split the data in to "slices", allowing these "slices" to be processed in parallel and thus decreasing the length of time taken to perform the crawl. If you check this option, you'll be able to specify the number of slices to use. The sql you specified for the full crawl will then be modified by the connector to include a where clause performing the "slice".

For example, if your sql was *select * from table* and you chose 10 slices, then 10 sql statements would be executed at the server. The sql executed would be *select * from table where id mod 10 = n* (where n is a value between 0 and 9).

ID

Enter the column that holds the id of the row and specify if this column is a string

Enable Post Crawl SQL

Check this box if you want the ability to run a piece of sql once the crawl has completed, and enter that SQL in the *Post Crawl Sql* entry

Configure Incremental Crawl

Check this box if you want to configure incremental crawls and then select from the various options.

The incrementals would only work if the "Distributed Children Processing" is unchecked in the "Advanced Connector Properties" section.

Incremental crawl bounding

Checking this option allows incremental crawls to use SQL that is bounded by a condition. When entering SQL you may use the variables in a WHERE clause to limit the data collected. The upper bound will be calculated at the start of the crawl and the lower will be the upper from the previous crawl. Two types of bounding are available - *Timestamp* uses the current system time whilst *SQL* allows you to define SQL to return the bounds when the crawl starts.

Pre incremental Crawl SQL

SQL statements to run before an incremental crawl. This SQL can be used to mark documents for update, save timestamps, clear update tables, and other actions as needed to prepare for an incremental crawl. This field can be left blank if you never do an incremental crawl

Incremental Crawl SQL

SQL statements to run for an incremental crawl. This SQL should provide a list of all adds and deletes to the documents in the index. Some field names have special meaning, such as title, content, url, and id,. Note the special column "action" should report I (for insert), U (for update), or D (for delete).

Post incremental Crawl SQL

SQL statement sot run after each record is processed. This SQL can be used to unmark/delete each document from the table after it is complete.

Post incremental Crawl SQL (failures)

SQL statements to run after each record if processing fails. If this SQL field is left blank, the SQL entered in the "Post incremental crawl SQL" field will run instead.

Action column

Enter the name of the column in the returned data which holds action of the update (ie Insert, Update or Delete). This must match the name returned by the SQL. If the column is aliased using the SQL "AS" construct,
you should provide the alias name here

Sequence column

Enter the name of the column in the returned data which holds the sequence number of the update. This match the name returned by the SQL. If the column is aliased using the SQL "AS" construct, you should provide the alias name here

ACL fetching

If required, acls can be added to the documents collected by the connector. Choose *Column* to specify the name of a column in the main data holding the acls, or *SQL* to enter a separate piece of sql to collect the ACLs

RDBMS URLs

An RDBMS "URL" is needed to tell the connector application what database to crawl. The exact form of this URL is dictated by the JDBC driver and therefore the database vendor, but will be of the form

```
jdbc:<vendor>://<server>:<port>/<database>
```

> ***For example:*** jdbc:mysql://192.168.40.27/wikidb

See your database vendor's documentation for more information on JDBC URLs.

Step 3c: Specify Workflow Information

In the "Workflow" tab, specify the workflow steps for the jobs that come out of the crawl. Drag and drop rules to determine which steps should an item follow after being crawled. This rules could be where to publish the document or transformations needed on the data before sending it to a search engine.

1. For the purpose of this tutorial, drag and drop the ''Publish To File'' rule found under the ''Publishers'' tab to the '''onPublish''' Workflow tree.
    a. Specify a "Name" and ''Description'' for the Publisher.
    b. Click ***Add***.

After completing this steps click ***Save*** and you'll be sent back to the **Home** page.

Step 4: Initiate the Full Crawl

Now that everything is set up, actually initiating the crawl is easy.

Now that the content source is set up, the crawl can be initiated.

1. Click on the crawl type option to set it as "Full" (is set as "Incremental" by default and the first time it'll work like a full crawl. After the first crawl, set it to "Incremental" to crawl for any changes done in the repository).
2. Click ***Start***.

Note that content sources will be automatically initiated by the scheduler based on the schedule you specified for the content source, be it once a day, once a week, every hour, etc. But you can always start a crawl at any time by clicking on the "Full" button.

Be aware that Aspire will never initiate multiple simultaneous crawls on the same content source. Of course, multiple jobs may be crawling different content sources at the same time.

This means that you can click on "Full" or "Update" and not have to worry about the scheduler perhaps scheduling a crawl on the same content source. The scheduler will always check to see if the content source is actively crawling before starting its own crawl of that same content source.

During the Crawl

During the crawl, you can do the following:

- Click on the "Refresh" button on the Content Sources page to view the latest status of the crawl.
  The status will show "RUNNING" while the crawl is going, and "CRAWLED" when it is finished.
- Click on "Complete" to view the number of documents crawled so far, the number of documents submitted, and the number of documents with errors.

If there are errors, you will get a clickable "Error" flag that will take you to a detailed error message page.

Step 5: Initiate an Incremental Crawl

If you only want to process content updates from the RDBMS (documents which are added, modified, or removed), then click on the "Update" button instead of the "Full" button. The RDB connector via Table will automatically identify only changes which have occurred since the last crawl.
If this is the first time that the connector has crawled, the action of the "Update" button depends on the exact method of "change" discovery. It may perform the same action as a "Full" crawl crawling everything, or it may not crawl anything. Thereafter, the Update button will only crawl updates. Scheduled crawls are always "Update" crawls. This means that the you may need to manually perform a "Full" crawl initially before using scheduled jobs after that to perform "update" crawls.

Statistics are reset for every crawl.