

## SimSage Database Crawler set up

This document sets out the process for setting up a database crawler on the SimSage platform. In addition it covers what is needed for running a local copy of that database crawler to ingest information from the database into SimSage.

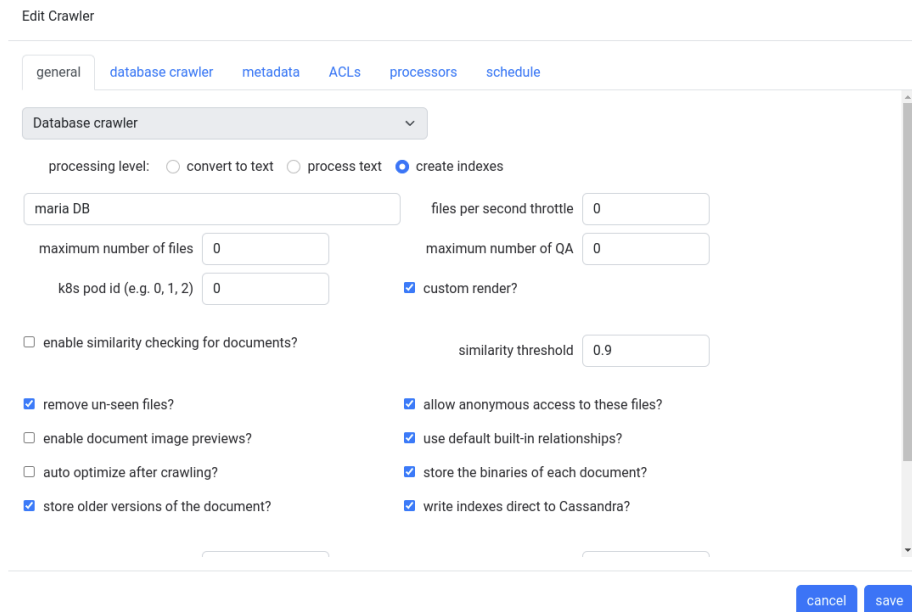
### SimSage Source configuration

1. Navigate to the sources section of the admin UX.
2. Create a new crawler in the admin UX, using the add new crawler button.
3. Follow the below steps to populate the New crawler for.

#### The General tab

1. Select “Database Crawler” from the dropdown list.
2. Check custom render.
3. Unselect the “enable document image previews” checkbox.
4. Depending on your specific security requirements check/uncheck the “allow anonymous access to these files” checkbox.

(At present we cannot map any fields of the database into ACLs)



The screenshot shows the 'Edit Crawler' form in the SimSage admin interface. The 'general' tab is selected. At the top, there are tabs for 'general', 'database crawler', 'metadata', 'ACLs', 'processors', and 'schedule'. Below these, a dropdown menu is set to 'Database crawler'. The form contains several configuration options:

- processing level:** Radio buttons for 'convert to text', 'process text', and 'create indexes' (selected).
- Database:** A text input field containing 'maria DB'.
- files per second throttle:** A text input field containing '0'.
- maximum number of files:** A text input field containing '0'.
- maximum number of QA:** A text input field containing '0'.
- k8s pod id (e.g. 0, 1, 2):** A text input field containing '0'.
- custom render?:** A checked checkbox.
- enable similarity checking for documents?:** An unchecked checkbox.
- similarity threshold:** A text input field containing '0.9'.
- remove un-seen files?:** A checked checkbox.
- allow anonymous access to these files?:** A checked checkbox.
- enable document image previews?:** An unchecked checkbox.
- use default built-in relationships?:** A checked checkbox.
- auto optimize after crawling?:** An unchecked checkbox.
- store the binaries of each document?:** A checked checkbox.
- store older versions of the document?:** A checked checkbox.
- write indexes direct to Cassandra?:** A checked checkbox.

At the bottom right, there are 'cancel' and 'save' buttons.

#### The Database crawler tab

The database crawler is an external crawler. This means that we give you a java based piece of software that connects to SimSage, reads the details provided securely, and connects to the local instance of your database. This external crawler then encrypts each record and sends it over HTTPS to SimSage.

1. Populate the various fields using the descriptions provided in the below table, as seen in the screenshot.

UI name	Description
user name	We need to provide a user-name and a password for the user connecting to the database using JDBC. JDBC does not support Microsoft passthrough security, so if you're using a Microsoft SQL server you'll need to enable username/password authentication if you haven't done so already.
password	See user-name above
jdbc string	The JDBC string determines what server we connect to. The JDBC string is one of the following (assuming the default ports for each database).  jdbc:mysql://<ip-address>:3306/<database-name> jdbc:microsoft:sqlserver://<ip-address>:1433;DatabaseName=<database-name> jdbc:postgresql://<ip-address>:5432/<database-name>  The <i>ip-address</i> can be a hostname if that name is resolvable from the machine you are running the crawler from. The <i>database-name</i> is the name of the database on the SQL server. This is the "use <database-name>;" part you'd ordinarily use in SQL to select a database.
database	Select the database type from the database dropdown. At present we support three different databases. Microsoft SQL, MariaDB / MySQL and Postgres.
pk field	The primary key is the name of a unique key used in the select statement below that uniquely identifies this record. It can be a string or a number but must be unique.
web fields	Web fields is disabled because of checking [x] customer render in th previous steps. This can be used alternatively by unchecking this value to point to a web based view of a record using HTTP or HTTPS if available.
select	This is the select statement which can be a SQL view or a join of several tables. Ultimately this is the SQL query run (with pagination) against your database. The fields selected must be inside this query, <b>do not use * to select all fields</b> .
text index template	This is what SimSage indexes. This should be text and numbers. SimSage creates a "text file" from the fields in this multi-line edit box. You can add additional text in this box as you like, but SQL fields must be in the select statement and must be enclosed in square brackets as shown in the screenshot above. Punctuation can be helpful to separate data into separate sentences for SimSage.
html template	Not used at present, please set to <div />

Edit Crawler

general
database crawler
metadata
ACLs
processors
schedule

user name
simsage
password
password

jdbc string
jdbc:mysql://192.168.7.114:3306/classicmodels

database
MySQL
pk field
customerNumber

web fields
document http/https reference SQL fields in square brackets [FIELD-NAME]

select
select customerNumber, customerName, contactFirstName, contactLastName, addressLine1, addressLine2, city, state, postalCode, Country from customers ;

text index template
[customerNumber]. [customerName]. [contactFirstName] [contactLastName]. [addressLine1], [addressLine2], [city], [state], [postalCode], [country].

html template
<div />

cancel
save

## The metadata tab

1. Remove any values by clicking the garbage can icon until you see the following page.

Edit Crawler

[general](#) [database crawler](#) [metadata](#) [ACLs](#) [processors](#) [schedule](#)

All rows in order of UI. Use 'actions' arrows to re-arrange existing rows.

data-type	source field	UI display-name	metadata name	sortable	actions

[cancel](#) [save](#)

## The ACLs tab

In this tab we can customise the security based on existing SimSage Users and Groups.  
(NB: Users and Groups can be imported from external systems).


In the below screenshot example we have selected the SimSage default User group as having access. Note, if you have opted to use the “allow anonymous access to these files” configuration, these ACLs settings will be overridden. *SimSage always recommends setting up a group initially in case you change your mind at any point.*

Edit Crawler


[general](#) [database crawler](#) [metadata](#) [ACLs](#) [processors](#) [schedule](#)

this list sets a default set of Access Control for this source

ACLs filter

 Users R W D M

available filter

 Administrators

[cancel](#) [save](#)



### The Processors tab

For this particular set up, you can ignore this tab.

### The Schedule tab

Here you can specify when you would like the external crawler to be allowed access to the SimSage platform.

This is done by setting desired timeframes to active. In the example provided we used the “select all” button to allow SimSage to always be available to your external crawler.

Edit Crawler

---

[general](#) [database crawler](#) [metadata](#) [ACLs](#) [processors](#) [schedule](#)

---

all times in GMT (now Tue, 14:58)

	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
Monday																								
Tuesday																								
Wednesday																								
Thursday																								
Friday																								
Saturday																								
Sunday																								

☒ active ☐ inactive

---

## External Crawler Configuration

The external crawler is a small Java program, provided by SimSage for the version of your platform.

```
rock@rock-office:~/crawlers$ ll
total 24
-rwxrwxr-x 1 rock rock 1943 Jan 17 13:07 crawler.sh*
drwxrwxr-x 2 rock rock 12288 Jan 17 13:07 lib/
-rw-rw-r-- 1 rock rock 695 Jan 17 13:18 system.properties
rock@rock-office:~/crawlers$
```

The software provided consists of,

- A *lib* folder with all the java libraries required to run the crawler.
- A *crawler.sh* executable shell file for running the crawler,
- and a *system.properties* file that needs to be edited to match your platform.

If you are using windows you can use the content of the *crawler.sh* file to create a batch file to run your crawler.

The content of the *crawler.sh* file is as follows:

```
#!/bin/bash

if [ "$JAVA_HOME" == "" ]; then
    echo "JAVA_HOME not set"
    exit 1
fi

rp=$(realpath "$0")
HOME=$(dirname "$rp")

# include the setup environment
source $HOME/system.properties

CP=`echo $HOME/lib/*.jar | tr ' ' ':'`
$JAVA_HOME/bin/java \
    -Djdk.attach.allowAttachSelf=true \
    --add-modules java.se \
    --add-exports java.base/jdk.internal.misc=ALL-UNNAMED \
    --add-exports java.base/jdk.internal.ref=ALL-UNNAMED \
    --add-exports java.base/sun.nio.ch=ALL-UNNAMED \
    --add-exports java.management.rmi/com.sun.jmx.remote.internal.rmi=ALL-UNNAMED \
    --add-exports java.rmi/sun.rmi.registry=ALL-UNNAMED \
    --add-exports java.rmi/sun.rmi.server=ALL-UNNAMED \
    --add-exports java.sql/java.sql=ALL-UNNAMED \
    --add-opens java.base/java.io=ALL-UNNAMED \
    --add-opens java.base/java.lang=ALL-UNNAMED \
    --add-opens java.base/java.nio=ALL-UNNAMED \
    --add-opens java.base/java.lang.module=ALL-UNNAMED \
    --add-opens java.base/jdk.internal.loader=ALL-UNNAMED \
    --add-opens java.base/jdk.internal.ref=ALL-UNNAMED \
    --add-opens java.base/jdk.internal.reflect=ALL-UNNAMED \
    --add-opens java.base/jdk.internal.math=ALL-UNNAMED \
    --add-opens java.base/jdk.internal.module=ALL-UNNAMED \
    --add-opens java.base/jdk.internal.util.jar=ALL-UNNAMED \
    --add-opens jdk.management/com.sun.management.internal=ALL-UNNAMED \
    --add-opens java.management/sun.management=ALL-UNNAMED \
    -XX:+ExitOnOutOfMemoryError -XX:+CompactStrings -cp $CP -Dsimsage.external.crawler=true \
    nz.simsage.external.crawler.MainKt "$@"
```

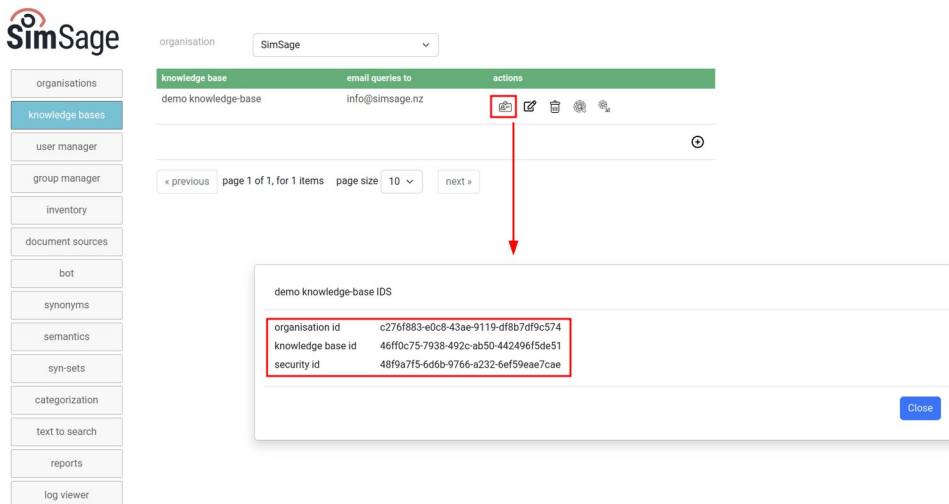
This shell file uses Java 11, but is compatible with Java 8 and Java 17.

(NB: Java 8 doesn't support any of the “*add-modules*” or “*add-exports*” or “*add-opens*” required by Java 11 or 17. You can remove these lines if this is the version of you wish to run.)

*HOME* is the folder this command is run from as an absolute path.

*CP* is the class-path for Java and is set to each entry in the *lib* folder separated by a colon without any spaces.

The contents of the *system.properties* file are “imported” into this shell command as environment variables that will be read by our Java program. Its content defines where your SimSage instance is located and what Organisation / Knowledge-base and security-id to use for communication.



The screenshot above shows the default values for the SimSage organisation. **These might differ for your set up. Please copy these values for your instance.**

Modify the *system.properties* file shown below to use the correct values.

```
#
# Crawler settings
#
export organisation_id=c276f883-e0c8-43ae-9119-df8b7df9c574
export kb_id=46ff0c75-7938-492c-ab50-442496f5de51
export sid=48f9a7f5-6d6b-9766-a232-6ef59eae7cae
export source_id=39

export api_version=1
export document_max_binary_size_in_mb=50
export sleep_after_crawl_in_mins=30
export max_text_length_in_kb=500

# the crawlers use a ticket-master and this is how many downloads they will do at once maximum
export crawler_concurrent_downloads=1
export page_size=10

# shared-secret salt for Edge devices and SimSage encrypted communication
export shared_secret_salt=91441972-dee6-411e-9e15-30d96ddfe3ff

# SimSage location
export simsage_endpoint=http://localhost:8080/api
```

The fields used are described below.

(NB: There are no spaces in any of the *export statements* in this script).

Field	Value
organisation_id	The id of your organisation. A GUID as shown in the screenshot above.
kb_id	The id of your knowledge-base. A GUID as shown in the screenshot above.
sid	The security id of your knowledge-base. A GUID as shown in the screenshot above. You can change this number using the admin UX (a random value used to encrypt data)
source_id	The id of the source you wish to talk. This must be the ID of your source in the admin UX.
api_version	The version of the SimSage API, do not change.
document_max_binary_size_in_mb	Not applicable for the database crawler, the maximum allowed size of a binary file in MB. This constant is also set inside the SimSage platform as a default to the same value. You cannot increase this value without changing your SimSage instance first.
sleep_after_crawl_in_mins	The external crawler will run until you stop it. After this crawler completes a run it will go dormant for as many minutes specified here before it starts again and checks for any changes.
max_text_length_in_kb	The maximum allowed size of a text file in KB. This constant is also set inside the SimSage platform as a default to the same value. You cannot increase this value without changing your SimSage instance first.
crawler_concurrent_downloads	Not used by the Database crawler, do not change.
page_size	The number of records to get per paginated select statement.
shared_secret_salt	Not used by the Database crawler, do not change.
simsage_endpoint	The API endpoint of your SimSage server. This needs to be set to the correct server. If your company is "company" then the likely value of this setting should be: <a href="https://company.simsage.ai/api">https://company.simsage.ai/api</a> Make sure you use HTTPS and don't forget the API at the end of this URL.