CSV Data Imports

- Overview
- Import Methods
  - Report Builder Import
  - View Builder Import
- Setup
  - Writable Data Source
    - Writable HSQL Data Source
  - Role Permissions
    - Report Builder
    - View Builder
- Prepare CSV File
  - Do
  - Don't

Overview

top
Yellowfin's CSV functionality allows business users to transform their existing spreadsheets into reports. This means that any of Yellowfin's reporting capabilities can be quickly applied to spreadsheet contents, including the ability to drill down and categorise data.

Yellowfin allows spreadsheet data (stored as CSVs) to be imported into a specified data source and used for reporting. There are two methods for importing CSV data to use for report building;

1. Import a new CSV file through the Report Builder process
2. Import a new CSV file through the View Builder process

Import Methods

top
When importing a CSV file for reporting use, it's important to consider what the data will be used for, and which import method would best suit that use.

Report Builder Import

If the CSV has been developed for stand-alone use, or the CSV data needs to be stored in a separate data source from other reporting data, and the desired reporting results can easily be achieved with advanced sub query functionality, then the Report Builder import method should be used.

When the Report Builder import process is used, the following happens:

1. The user defines any calculations or converters to be applied to the data, creating additional fields in the final table.
2. The user defines all default formatting and relationships between fields, such as drill down, through the CSV step. These will be reused whenever someone selects this CSV for report building.
3. Yellowfin creates a View with all of the above fields, formatting, and relationships for reuse by other report writers. This View will be available during the report writing process, and the View itself can be edited by users with permission in order to adjust the default settings defined.

View Builder Import

If the CSV data is to be used in conjunction with tables from a reporting data source, the view import method should be used. This option also allows the import to be used across multiple views, rather than just multiple reports.

When the View Builder import process is used, the following happens:

1. The user defines any converters to be applied to the data, creating additional fields in the final table.
2. Yellowfin creates a table that is listed in the View Builder, allowing the user to select it, as they would any other table from the selected source connection. This table will be available to any View built on the writable source connection.
3. The user selects the CSV table and adds it to the View like they would any other table, joining it to other tables, etc.
4. The user defines any default formatting and relationships between fields, such as drill down, through the standard View Builder steps, rather than the CSV step.

Setup

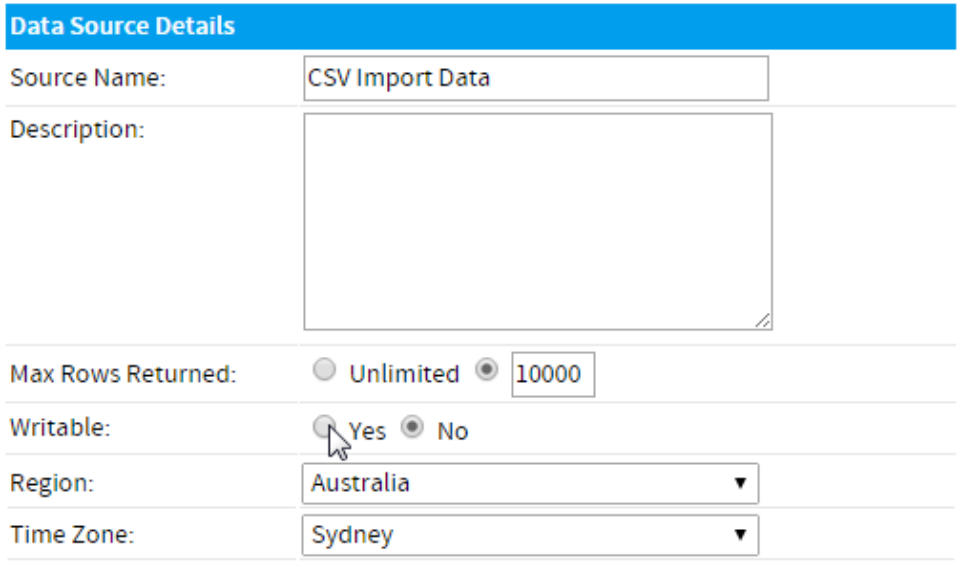
top

Writable Data Source

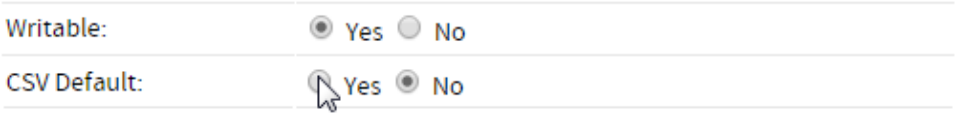
Yellowfin requires a Data Source connection to be defined as writable in order to write the contents of the CSV file to a table in the database. Often the writable connection will be pointing to an empty database, created solely for the purpose of importing CSVs or caching views. This is to ensure that any imports are kept separate from the main data sources.

In order to define a writable connection, complete the following:

1. Navigate to **Administration** > **Admin Console**.
2. Expand the **Data Sources** section of the Admin Console main panel.
3. If you already have a connection defined that you wish to set to writable, simply click on the connection name, otherwise create a new connection by clicking the **Add** button.
4. In the **Data Source Details** section of the connection setup, set the **Writable** option to **Yes**.



5. In the case that a user has permission to import a CSV, but not permission to access the advanced options, they will need there to be a writable source connection that has been set to the **CSV Default** so Yellowfin knows which connection to select for them. If this is the case for some of the users in the system, set the **CSV default** to **Yes**.



6. Complete the process by clicking the **Save** button at the bottom of the page.

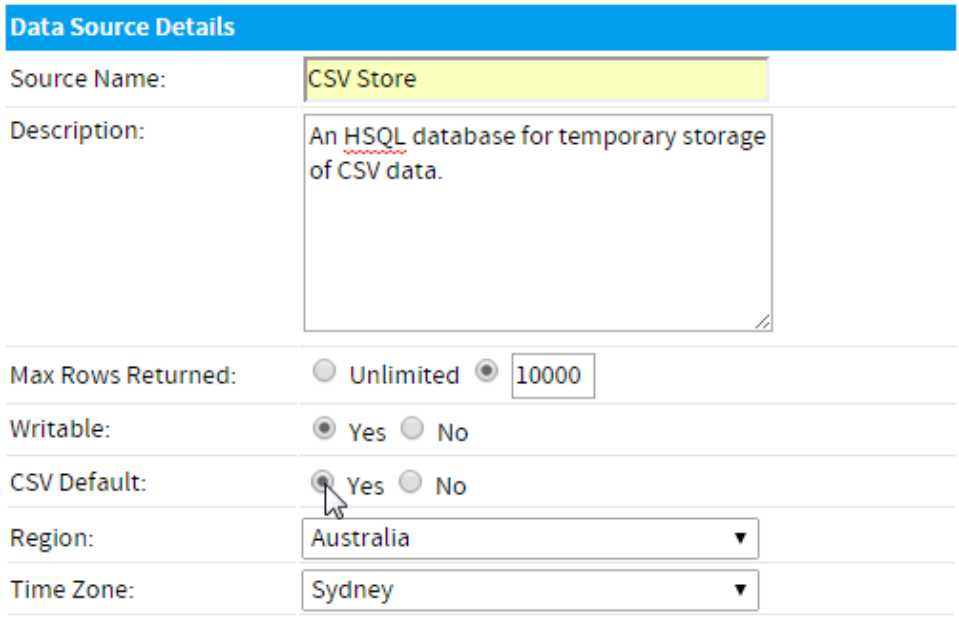
Writable HSQL Data Source

Often when working outside of a Development, QA, or Production instance of Yellowfin, you may have a need to import CSV files on a temporary or trial basis. An example of this would be trialling CSV file formats on a temporary setup, or perhaps creating sample content in a Proof Of Concept (POC) type scenario. In this situation you may not want, or be able, to set up a stable database for the purpose of storing CSV data. A temporary solution to this would be to create a writable HSQL database, similar to the one you can install Yellowfin into. In order to do this, follow these steps:

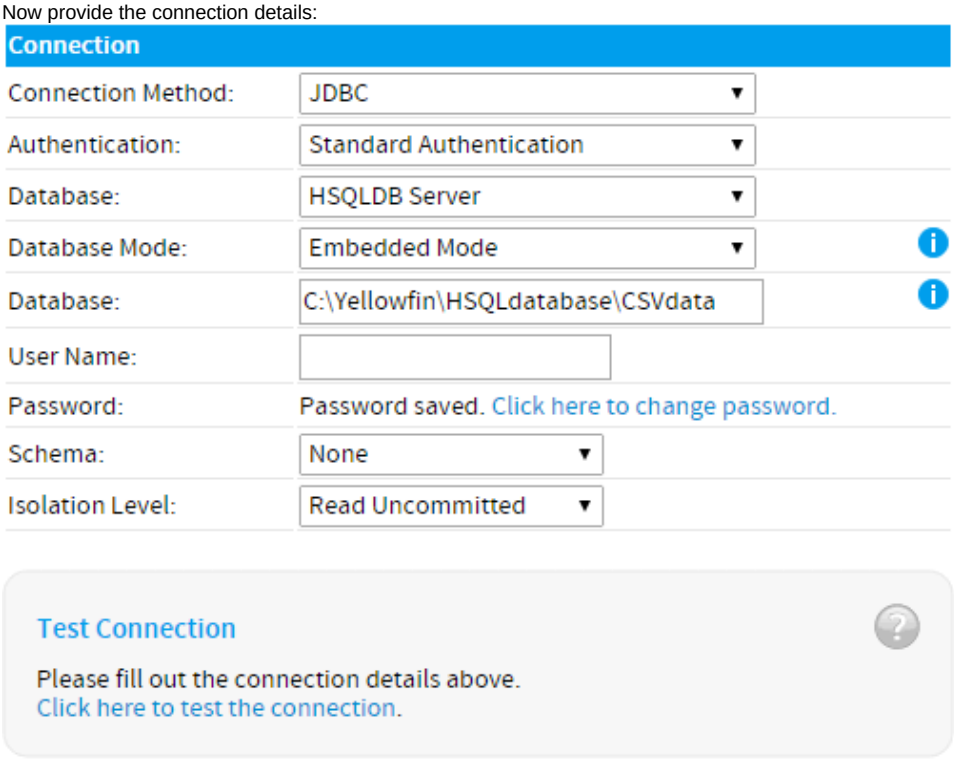
Note: HSQL databases are not stable and should never be used as part of a long term solution.

1. Create a directory, within your Yellowfin installation directory, to store the HSQL database. For example: `C:\Yellowfin\HSQLdatabase`
2. In Yellowfin, navigate to **Administration** > **Admin Console**.
3. Expand the **Data Sources** section of the Admin Console main panel.
4. Click the **Add** button.

5. Give your data source a **Name**, **Description**, and set the **Writable** option to **Yes**. You can also set it to **CSV Default**, this is optional.

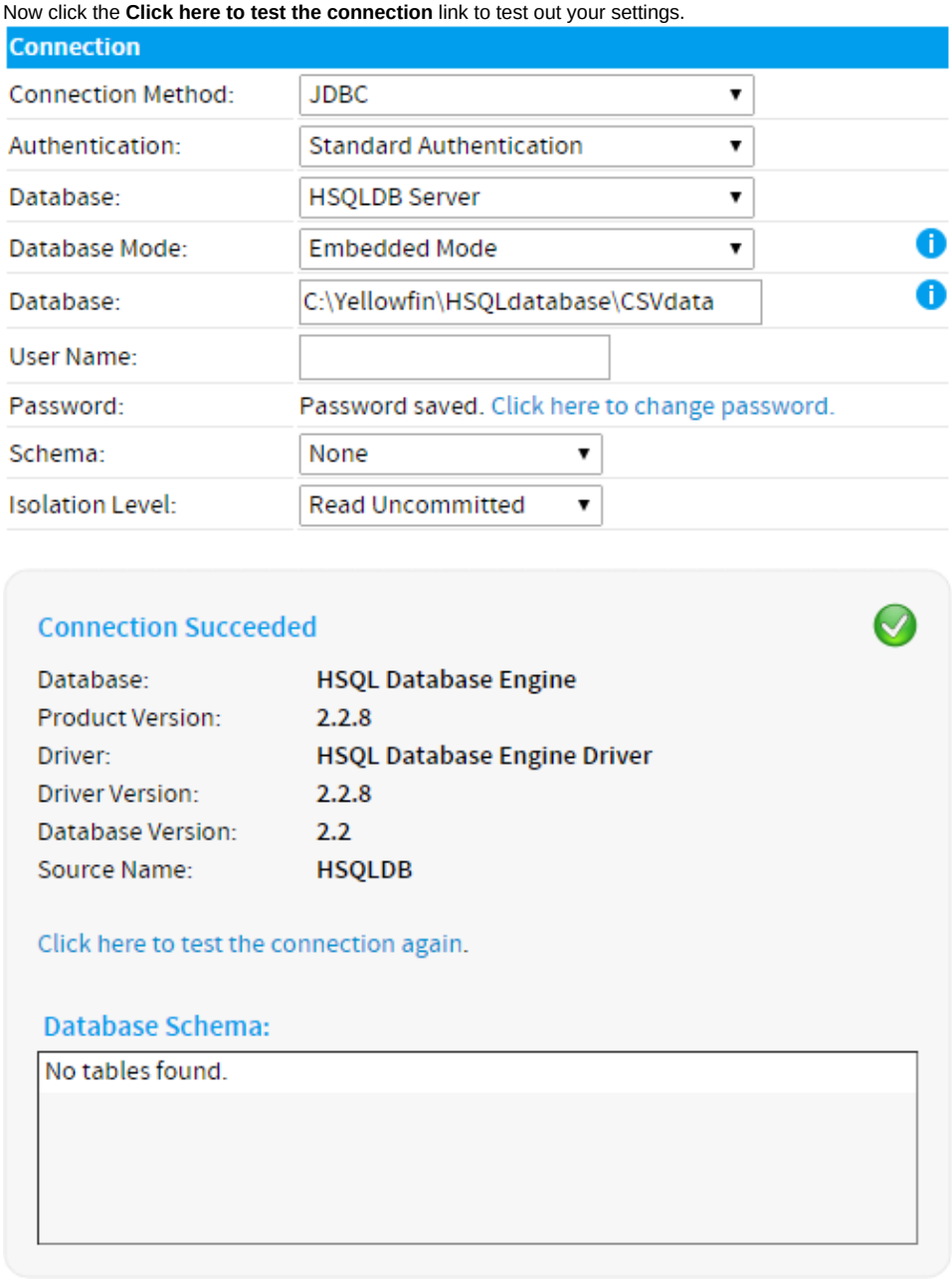


6. Now provide the connection details:



   a. **Connection Method:** JDBC
   b. **Authentication:** Standard Authentication
   c. **Database:** HSQLDB Server
   d. **Database Mode:** Embedded Mode
   e. **Database:** `C:\Yellowfin\HSQLdatabase\CSVdata`
      The value of this should be the directory you wish to store the HSQL database in, created in step 1, followed by the name of your database (Yellowfin will create this) - in this example we used **CSVdata** as the name.
   f. **Username:** *leave this blank*
   g. **Password:** *leave this blank*
   h. **Schema:** None
   i. **Isolation Level:** Read Uncommitted

7. Now click the **Click here to test the connection** link to test out your settings.



8. Complete the process by clicking the **Save** button at the bottom of the page.

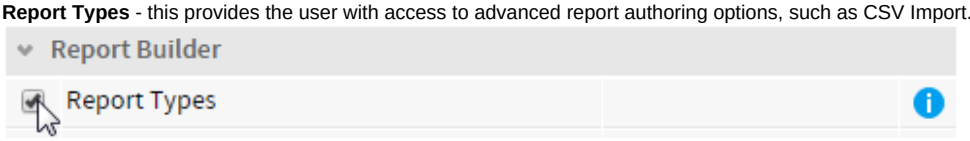
Role Permissions

In order for a user to import CSVs in Yellowfin they must have the appropriate role permissions defined. As there are two methods of importing CSVs, there are also two sets of role permissions available.

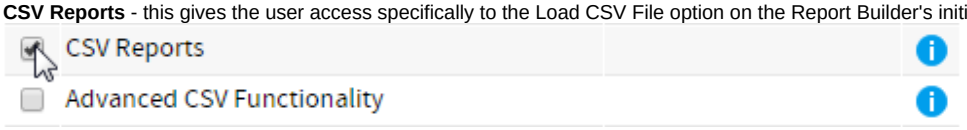
Report Builder

A user may be granted permission to import CSV files through the Report Builder. This is the less advanced import method of the two available. If the user needs to import CSV data through the Report Builder, they will require the following Role Permissions.

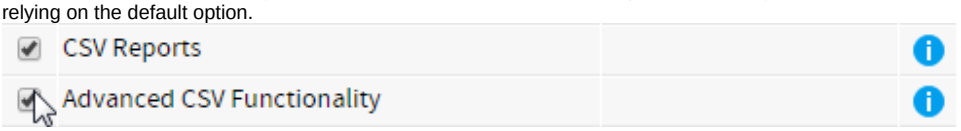
1. Navigate to **Administration** > **Admin Console**.
2. Open the **Roles** section and click on the role you wish to enable CSV access for.
3. Enable the following:
    a. **Report Types** - this provides the user with access to advanced report authoring options, such as CSV Import.



b. **CSV Reports** - this gives the user access specifically to the Load CSV File option on the Report Builder's initialisation step.



c. **Advanced CSV Functionality** - this allows users to select advanced options, including which writable data source to use, rather than relying on the default option.



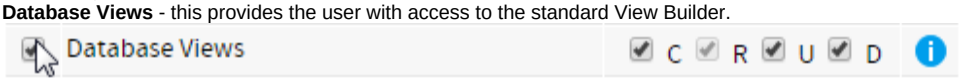
Note: this is an *optional* role permission.

4. Save the role.

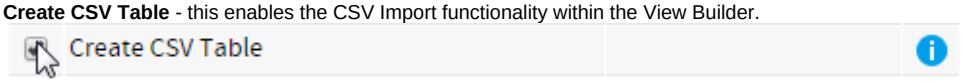
View Builder

A user may be granted permission to import CSV files through the View Builder. These permissions are generally only granted to users that maintain standard Yellowfin Views as part of their role already. They may also be granted the Report Builder import permissions, allowing them to user either method. If the user needs to import CSV data through the View Builder, they will require the following Role Permissions.

1. Navigate to **Administration** > **Admin Console**.
2. Open the **Roles** section and click on the role you wish to enable CSV access for.
3. Enable the following:
   a. **Database Views** - this provides the user with access to the standard View Builder.



   b. **Create CSV Table** - this enables the CSV Import functionality within the View Builder.



4. Save the role.

Prepare CSV File

top

Before the import process can begin, you will need to obtain or create your CSV file. There are some guidelines in place to ensure a compatible file.

Do

1. **Column Based** - ensure the data in your CSV is stored as columns, with no empty columns or rows at the start of the file (if examining through a spreadsheet tool).
2. **Date Format** - the date format recognised by Yellowfin is **`yyyy/MM/dd`**, so for example: **`2014/08/31`**. If possible, it's a good idea to ensure your dates are stored in this format. Yellowfin does have converters available as part of the import process, but ensuring your dates are in the correct format to start will avoid extra work during import.
3. **No Totals** - the CSV columns should not have any column or row totals applied, or other calculations and formats that may be lost during import, or mistaken for data values (rather than totals).
4. **Consistent Data per Column** - ensure that the data contained in each column is the same type throughout the file. This will ensure that Yellowfin can correctly identify and assign the field type for each column. For example, if you have a date column, and all the data is dates, except a few records that have the text "N/A", remove the text and leave that empty. This will mean Yellowfin recognises the field as a date, rather than text.
5. **File Format** - Yellowfin will only accept CSV files, no XLS, XLSX, or other formats will be accepted.

Don't

1. **Crosstab/Pivot** - do not attempt to import data that is in crosstab/pivot format as Yellowfin will examine the file as though it was in column format and as such will store the data in an unintended and unusable structure.
2. **Totals** - do not apply totals to your fields, as Yellowfin will not differentiate them from the rest of your data.
3. **XLS** - do not store your data in XLS or XLSX formats as Yellowfin will only read CSVs.

top