PROFESSIONAL SERVICES

User Guide

OnCommand Workflow Automation (WFA)

ACE Data Source

Prepared for: **ACE Data Source - Version 2.0.0**
Date: **October 2015**

Document Version: **2.0.0**

Abstract

The ACE Data Source (ACE-DS) is a general purpose WFA Data Source that can read from Microsoft Excel spreadsheets directly into a WFA scheme and and custom tables. Those tables are then available for search and selection in custom WFA user interface Queries, Filters, Finders and Commands. This allows workflows to align with and reflect customer business-rules, organization and environment information.

TABLE OF CONTENTS

1 Introduction

The ACE Data Source (ACE-DS) simplifies integrating customer business and environment information into the WFA environment and hence usable within their custom WFA workflows. Product Codes, Cost Centers, or Data Center Locations are examples of the sort of business information that can improve the effectiveness and useablility of WFA workflows.

WFA natively supports acquiring this kind of external data via custom WFA *Data Sources Types* (Designer tab -> Data Source Types). Two methods can be used by a *Data Source Type*: SQL or Script based. SQL Data Source Types can acquire information in external relational database. A Script *Data Source Type* is a custom script written in PowerShell than obtains data and formats it such that WFA can import it into custom scheme tables (defined by WFA Dictionaries).

The ACE-DS is a Script based *Data Source Type* that:

- Imports directly from specially formatted Excel spreadsheets into WFA Cache DB tables
- Is general purpose and re-useable. All that is required to re-use the ACE *Data Source Type* for a customer specific use-case is to clone it into a new WFA Scheme.

The ACE-DS bundle comes with a sample WFA Scheme named ace, sample WFA Dictionaries (ace.site and ace.storage) as well as a cooresponding spreadsheet named ace.xlsx with sample data. After importing the dar file into WFA a quick test can be performed by creating an execution Data Source (**Execution** tab -> **Data Sources** -> **New**) and import data from the included ace.xlsx spreadsheet into the WFA ace scheme and tables (site, storage).

NOTE: the 'ace' scheme is just an example and a holder for the ACE Data Source Type. The intention is a customer would create a new scheme and dictionaries and then clone the ACE Data Source Type and set it to be associated with the new scheme.

The contents of the ACE-DS bundle zip file are listed in the table below.

Table 1) ACE-DS Bundle Contents (ACE_DS_Bundle_Bv_#_#_#.zip)

File	Purpose
ACE_DS_Package_Pv_#_#_#.dar	ACE Data Source Type with example scheme and dictionaries
ACE_DS_UserGuide_Dv_#_#_#.docx	This User Guide document in Microsoft Word format
ace.xlsx	Example Excel spreadsheet for the ACE-DS

Where #_#_# indicates actual version number.

2 Prerequisites and Installation

The ACE Data Source Version 2.x.x has the following prerequisites:

- Windows Server 2008R2 or 2012R2 as WFA Server
- OnCommand Workflow Automation - WFA Version 3.0P1 or later
- It is assumed WFA is already installed and setup
- With WFA 3.1 and later the Linux flavor of WFA is not supported
- Microsoft Access Database Engine 2010 Redistributable

The ACE Data Source utilizes the free Microsoft Access Database Engine (aka: ACE driver / provider).

2.1 Installation Steps

In summary, the installation process is installing the ACE driver / provider and importing the ACE-DS dar file into the WFA system.

- Install the **Microsoft Access Database Engine 2010**
    - Search the Microsoft web site for the ***Microsoft Access Database Engine 2010 Redistributable*** installer file. This search string can be cut/pasted and used with google (for example) to quickly find the package:
      `site:microsoft.com “microsoft access database engine 2010 redistributable”`
    - On the Microsoft.com web-page, click the **[Download]** button. On the next page check the box to select the **AccessDatabaseEngine_x64.exe** installer file and download (~27MB) and execute this installer file.
    - The download and install takes no more than 1 to 2 minutes. The only questions are to accept the license agreement and the installation location. Taking the default install location is fine (C:\Program Files\Microsoft Office\)

- The **ACE Data Source** is distributed within a bundled .zip file containing the WFA dar file, an example excel spreadsheet and this documentation.
    - Obtain the bundled zip file and unpack it on your local system (not nessecarily the WFA server but could be if you are running a browser on the WFA Server itself)
    - Access the WFA system via a browser (http://<wfaSvr>/wfa) and login in as a WFA admin user.
    - Use the ***Administration -> Import*** menu selection to browse to the location where you have unpacked the bundle components and select the WFA .dar file:
      ACE_DS_Package_Pv_x_x_x.dar.
    - You will be shown the contents of the .dar container (two dictionaries, a scheme and a ScriptDataProvider). Click **[Import]** to continue.
    - At this point the ACE Data Source is installed. Continue to the next section of this document to create your own scheme and custom dictionaries or test with the included example **ace.xlsx** spreadsheet data.

3 Setup Custom Dictionary(s), Scheme and Spreadsheet

The ACE Data Sources supports a combination of:

- single or multiple independent table(s)
- multiple tables with relationships between them

The following sections step through:

- Mapping Rules: Excel Table-Column to WFA Table-Column,
- Single Spreadsheet Table into a Single WFA Table,
- Multiple Tables with Relations Between (ACE-DS Example Tables)

3.1 Mapping Rules: Excel Table-Column to WFA Table-Column

The overall concepts and mapping rules of the ACE Data Sources are:

- This data source reads an existing Excel .xlsx file specified in the execution Data Source's 'Host name:' field, for example: C:\WFA-Data\ace.xlsx
- The name of the excel file must match the WFA scheme name.
- The names of each worksheet/tab in the .xlsx file should match the names of a WFA Dictionary entity that you create with a matching name
- The names of columns should be placed in row 1 of each worksheet/tab and those names should match up with WFA Dictionary field names
- Specific dictionart entries can be created under any new scheme you'd like. Just clone this 'ACE Data Source' and place it into your new scheme.
- Don't forget to generate your tables using reset scheme, before first use.
- To use relationships between tables, follow these additional rules:
    - Your primary key should be the first column and called "id"
    - Your primary key should be unique, and can be text
    - Your foreign key columns should end with "_id" and match its parents id. It can be text.
    - If you don't have a primary key column, the id column will be added automatically with NULL values.
    - *NOTE: All WFA tables have an id column internally which is not visible in the WFA Dictionary for the table. WFA automatically creates the id field with an incrementing value when the data is acquired. For example, you see it as a column of the cluster table of the cm_storage scheme/database as cm_storage.cluster.id*

Next Steps:

- The next section illustrates a simple table example with instructions for a mythical company named **Acme**. The section is: **Single Spreadsheet Table into a Single WFA Table.**
- The section after illustrates ACE-DS relationships using the example spreadsheet (ace.xlsx) that is supplied with the ACE Data Source bundle. Jump forward to that section: ***Multiple Tables with Relations (ACE Example)*** which allows you to quickly import and test the **ace** example tables.

3.2 Single Spreadsheet Table into a Single WFA Table

The first example is for a mythical company named Acme. We want to create a table of the various departments in Acme along with the names of the storage systems used by each department.

- A WFA Scheme named **acme** and a Dictionary named **department** will be created
- The WFA Dictionary columns will be: **code**, **name**, **cluster**
- The Excel spreadsheet column headings will be: **code**, **name**, **cluster**

The steps to set this all up would be:

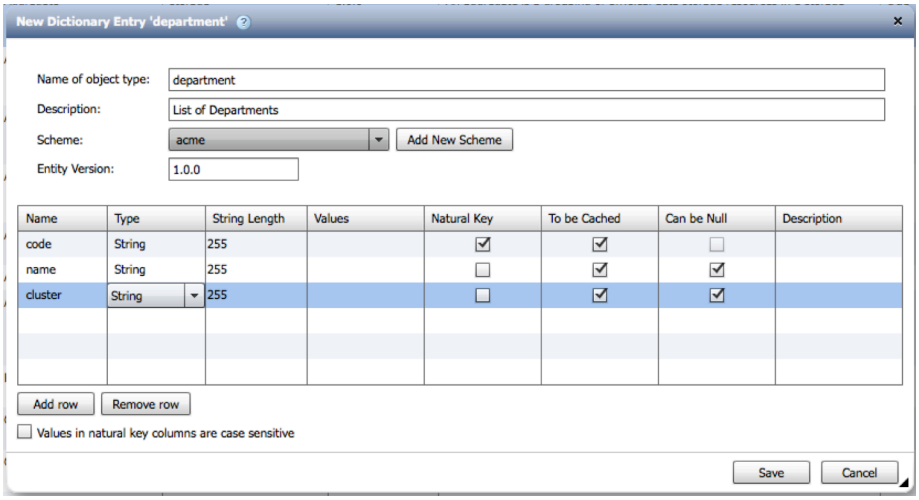
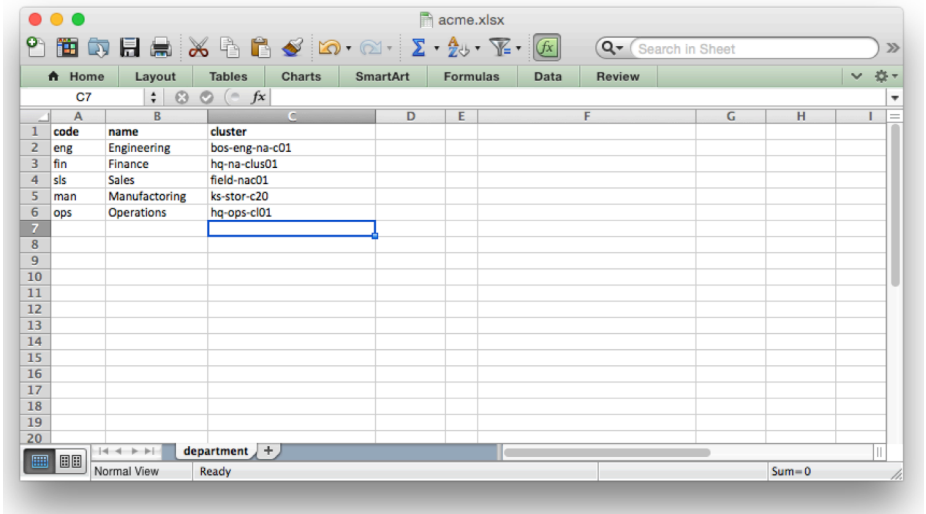
- Create the Excel spreadsheet named **acme.xlsx**. Rename the first worksheet tab to **department**. Fill out the row-1 headings (**code**, **name**, **cluster**) and populate some or all of the data rows. See example spreadsheet which follows.
- *NOTE: Determine where you will place the acme.xlsx file and later you will tell the ACE Data Source where it is located. Normal practice has been to place the file somewhere on the WFA server itself. Having a directory on the WFA server which is shared lets users access and update the spreadsheet from their desktop or laptop instance of Microsoft Excel.*
- Create the WFA Dictionary named **department**. This is done by:
  - **Designer** tab -> **Dictionary ->** click **New** icon button (page+)
  - As you create this Dictionary, use the **[Add New Scheme]** button to create the new scheme named **acme** which this dictionary will be within.
  - Add the Dictionary columns (**code**, **name**, **cluster**). Set the **type** of each column to String. Check the box [x] to make the **code** column a natural key, leave other fields as default.
  - Click **[Save]** to create the WFA Dictionary **department** within the new scheme named **acme**
- Next you will clone the ACE Data Source Type and set it to be associated with the new **acme** scheme. This is done by:
  - **Designer** tab -> **Data Source Types**
  - Select and highlight the data source named: **ACE Data Source**
  - Click the **Clone** icon button or right-click and select **Clone**
  - Fill out the **New Data Source Type** window as follows:
    - Leave the **Data source type:** name field as is (or change it if you like)
    - Leave Entity Version: and Data source version: as they are
    - For **Scheme:** select the scheme name created above: **acme**
    - Leave Default Port, Method and the Script itself alone and click **[Save]**
- Next create an exection Data Source that will read the spreadsheet on a recurring schedule. This is done by:
  - **Execution** tab -> **Data Sources** -> click **New** icon button
  - On the New Data Source window:
    - Name: **Acme Data Source** (for example)
    - Data source type: **ACE Data Source – 2.x.x** (unless you changed the cloned *Data Source Type* name above)
    - Host name: **C:\WFA-Data\acme.xlsx** (or where ever you placed it)

- Leave Port, User Name, Password and other fields as is
- Under the lower Scheduler Configuration section, click on the Scheme name **acme** and enter an interval such as **30** or **60** minutes or maybe **1440** minutes (daily) and click **[Save]**

- When saving this Execution Data Source it {may|should|will?} automatically do it's first acquire. It may be nessecary to right-click **Reset Scheme** before the first acquire. To quickly acquire again after changes have been to the spreadsheet right-click **Acquire Now**.

For this Acme example, the following table shows what the corresponding WFA Dictionary, Scheme, and Excel Spreadsheet would look like.

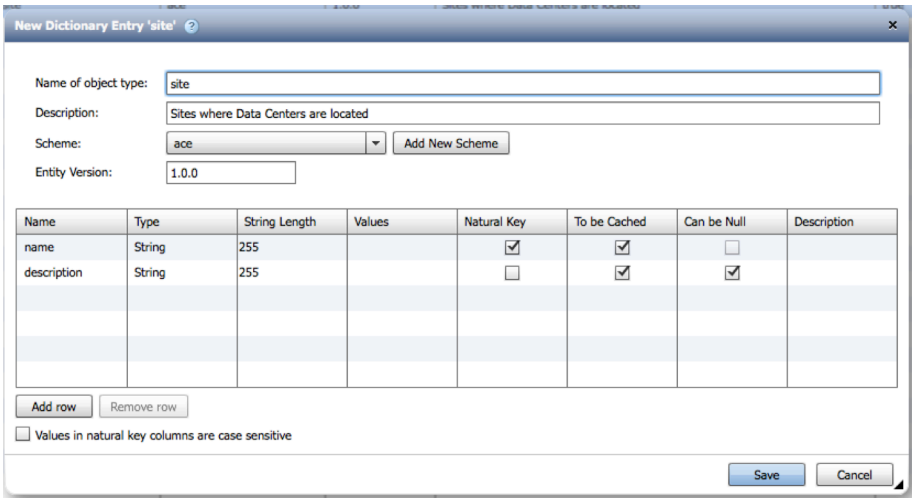
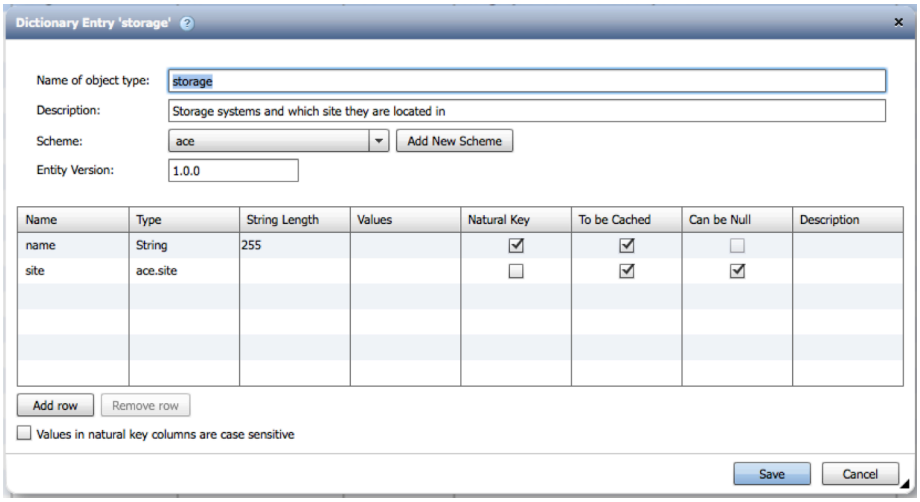
Table 2) Simple Acme Single Table Example

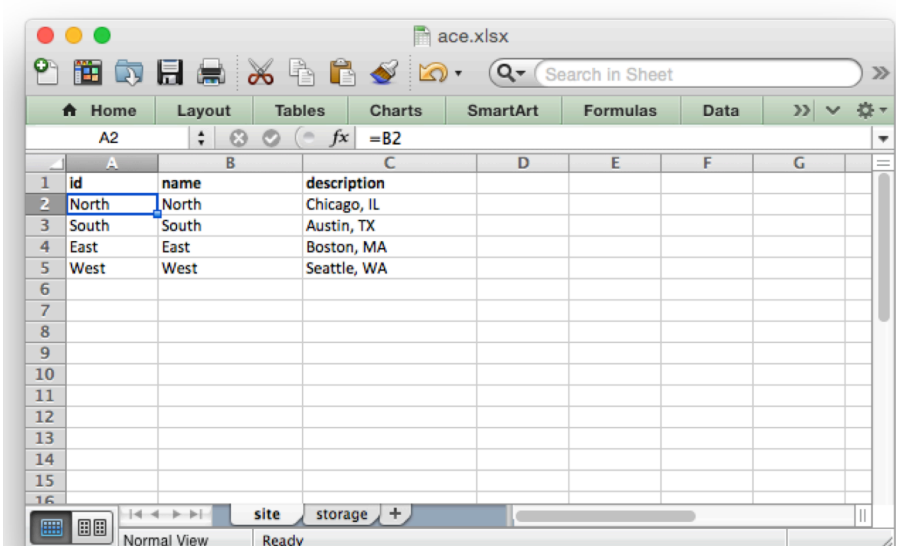
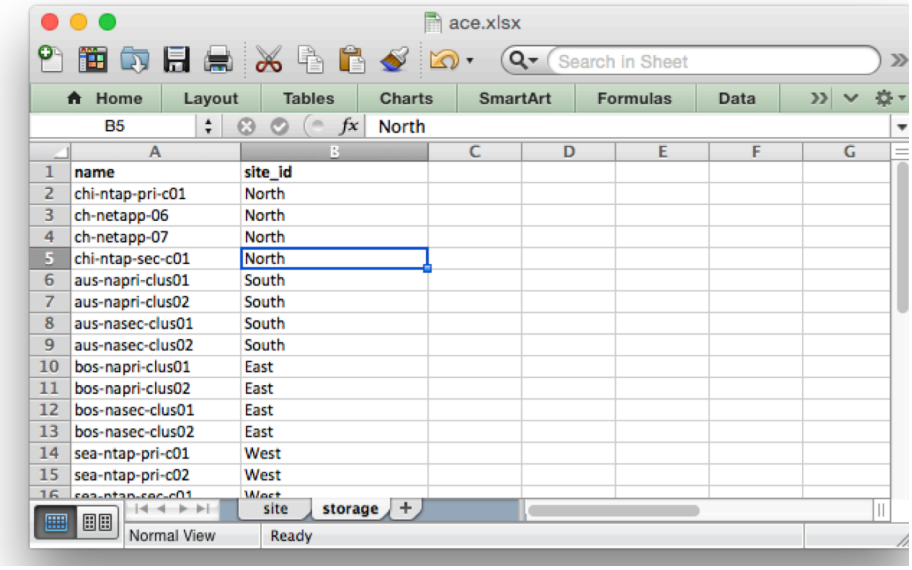
WFA Dictionary: department - Scheme: acme - Excel Spreadsheet: acme.xlsx	Notes
	1) Use the **Add New Scheme** button to create and select the acme scheme 2) Set each column **type** to string 3) Make code a **Natural Key**
	1) Create the file as **acme.xlsx** 2) Rename the first tab to **department** and remove any other tabs 3) Ensure the row-1 values match the above dictionary columns names 4) As a natural key, the **code** column values in each row must be unique

3.3 Multiple Tables with Relations Between (ACE-DS Example Tables)

The ACE-DS imports associated within the WFA scheme named **ace** which also includes two WFA Dictionaries: **site** and **storage** as shown below. These dictionaries map to the columns of the Excel spreadsheet included in the ACE-DS bundle, **ace.xlxs**.

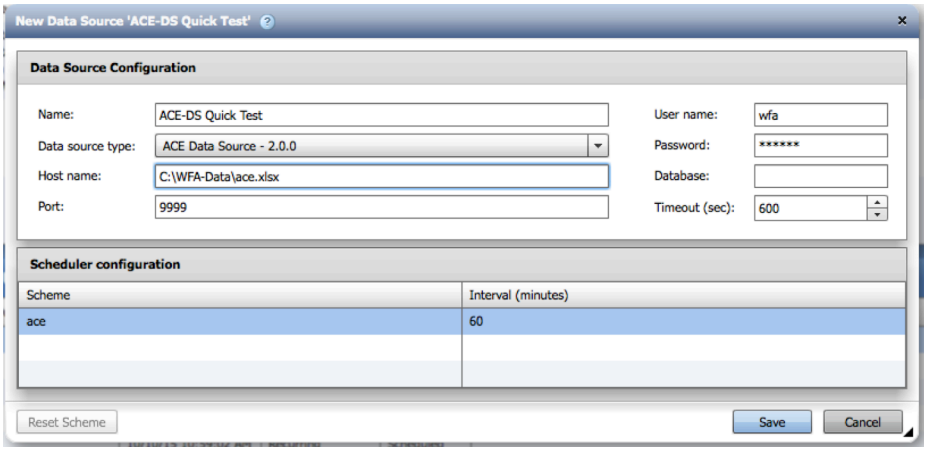
The **site** table defines data data center locations. The **storage** table is a list of storage system names (*such as a cluster_mgmt LIF DNS names*) along with a reference to which site that system is located in. See previous section **3.1 Mapping Rules**. The notes below describe how this relationship is setup.

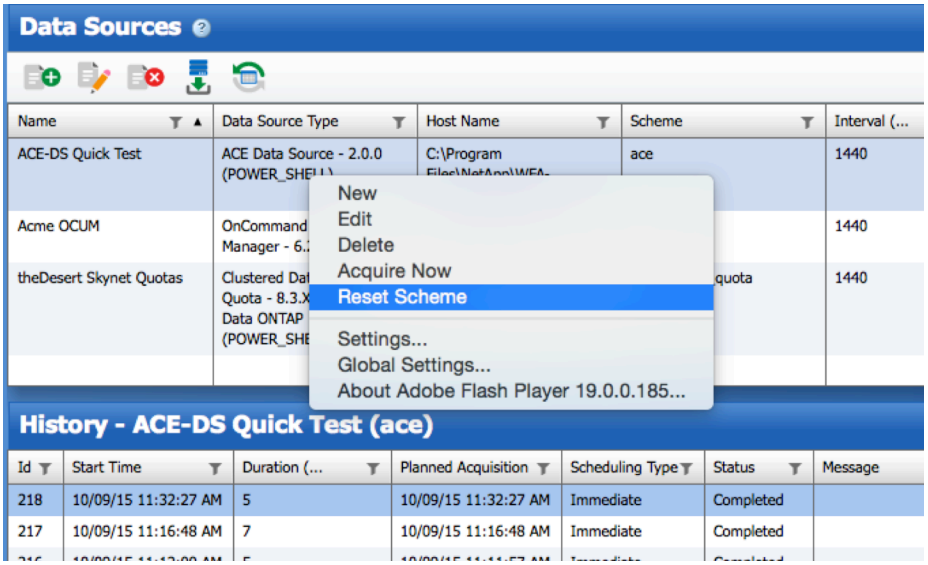
WFA Dictionary: site,storage - Scheme: ace - Excel Spreadsheet: ace.xlsx	Notes
	1) The site table as defined by the dictionary has **name** and **description** columns 2) The **name** column is marked as **natural key** and it's value must be unique. *NOTE: Internally the MySQL **site** table also has an **id** column (site.id) which is the primary key and is normally populated with incrementing numeric values.*
	1) The storage table as defined by the dictionary has **name** and **site** columns 2) Following ADC-DS mapping rules, **site** is set as a foreign key into the **ace.site** table. Select ace.site as the Type. 3) The **storage** table **name** field is a natural key and hence name values must be unique

WFA Dictionary: site,storage - Scheme: ace - Excel Spreadsheet: ace.xlsx	Notes
	1) The **ace.xlsx** file contains two tabs: site and storage which must match the dictionary names. Shown at left is the **site** tab 2) The first column is **id** which based on ACE-DS mapping rules means values must be unique and can be used in other Excel tabs to reference into rows of this table. 3) The **name** column holds the customer's desired *site-name* and it is unique so in Excel we've used the **name** column to create unique values for the **id** column, thus creating unique **id** values (Note: cell A1=B2 which is carried down through each row)
	1) The ace.xlsx file contains two tabs: site and storage which must match the dictionary names. Shown at left is the **storage** tab 2) The columns include name and site_id. The **name** column holds the storage system names. The **site_id** column, based on ACE-DS mapping rules, is treated as foreign key into the 'site' table. So it's values must match an **id** value in the site table. Such as '**North**' which matches one of the **id** rows in the site table above. *NOTE: once loaded into WFA tables, id and site_id are replaced with unqiue numeric incrementing values which reflect the relationship(s).*

The following tables show how to create an execution Data Source on your WFA server to acquire from the example 'acs' tables.

Creating Execution Data Source, Reset Scheme, Acquire Now





Notes

1) The Name: is what ever you like. **ACE-DS Quick Test** in this example.

2) Select the Data source type: as **ACE Data Source – 2.#.#**

3) Enter the path to ace.xlsx in the Host name: field such as: **C:\WFA-Data\ace.xlsx**

4) You may need to **Reset Scheme** immediately after the above step which creates the scheme and tables

5) Then do an **Acquire Now** to invoke the ACE Data Source to read the Excel file

Document History

Version	Date	Document Version History
Version 2.0.0	October 12th, 2015	First shared version

Go further, faster®