Creating Tests with Spreadsheet Import in Data Validation Option
Abstract
Data Validation Option 9.5.2 introduced a new method to create table pairs and to create test for table pairs and single tables. This document describes the new functionality for creating table pairs in a spreadsheet. It also describes how to create tests for table pairs and single tables.

Supported Versions
- Data Validation Option 9.5.2-9.6.0

Table of Contents
Overview .......................................................................................................................................... 2
Sample Testing Scenario............................................................................................................. 2
Summary ...................................................................................................................................... 3
Spreadsheet Import/Export of Table Pair Tests .............................................................................. 3
  Scenario 1 – Creating a Blank Spreadsheet to Use for Import.................................................... 3
  Scenario 2 – Create Tests in the Spreadsheet............................................................................ 5
  Scenario 3 – Import the Spreadsheet into DVO .......................................................................... 6
Spreadsheet Import for Compare Tables ........................................................................................ 7
  Scenario 1 – Adding Tables to the Compare Tables Import Spreadsheet .................................. 8
  Scenario 2 – Importing the Compare Tables Spreadsheet into DVO.......................................... 9

Overview
DVO 9.5.2 introduced a new way to create table pairs, and tests for table pairs and single tables. The first section of this document describes a new and flexible way to create tests for table pairs and single tables. Table pair generation is described later in this document. There are several reasons to use an external spreadsheet to define tests:
- DVO users can work offline when creating tests.
- Non-DVO users who are familiar with the data and testing requirements can define tests and provide them to a DVO user to import and use in DVO.
- ETL (Extract, Transform, Load) or testing specification documents are often defined as a source-to-target spreadsheet. Some of the definitions in those spreadsheets can often be repurposed for test creation.
- Large numbers of tests (particularly those that follow certain patterns) can be defined very quickly in a spreadsheet (using copy/paste and other basic spreadsheet features) and imported into DVO thus saving significant time and effort over test creation via the client tool.

Sample Testing Scenario
Imagine you want to test the following ETL scenario:
A flat file with 200 columns is read. Each column is defined as a string in the source definition. During the ETL, about 90 of those fields, which are actually numeric fields, are converted to decimal and written to the final target table.
Testing this scenario would require creating 200 value tests, one for each column that was read and written to the target table, but for those columns converted to a decimal, an expression (For example, TO_DECIMAL(<fieldname>)) would have to be created as part of those value tests.
While this could be done in the client tool, it would be tedious to manually create those 90 tests. But, by using basic cut/paste or search/replace capabilities within a spreadsheet, these 90 tests could be defined in about 1 minute or less. And, instead of 90 tests, if there were 200 or even 900 similar tests, the time to create them would be minutes at most: a huge productivity increase over using the client tool in this scenario.

Summary
Overall, the spreadsheet import provides an alternative to the client tool that is very effective when large number of similar tests must be generated.

The following sections contain instructions and explanations on using the spreadsheet functionality to create tests for individual table pairs as well as in mass creating table pairs using the compare tables dialog.

Spreadsheet Import/Export of Table Pair Tests

Scenario 1 – Creating a Blank Spreadsheet to Use for Import
Although any spreadsheet with the proper column formats can be used with DVO, samples are provided at the following locations:

- `<DVO client install dir>\samples\TablePair-template.xlsx`
- `<DVO client install dir>\samples\SingleTable-template.xlsx`

But, DVO also provides a simple way to export a pre-formatted spreadsheet with proper column headings and column metadata specific to a particular table pair or single table. This makes it much easier for a spreadsheet user to quickly define the tests they are interested in.

To create a spreadsheet from an existing table pair, perform the following tests:

1. Create a table pair that you want to use.
2. Ensure there is a join condition on the table pair.
3. Right click on the table pair and select Export Tests to Spreadsheet:
4. Select a location and name for the exported file and click **Save**.

The spreadsheet has been exported to your computer.

5. Open the spreadsheet you just generated in Excel. You should see the following:

There are two worksheets. The first takes its name from the table pair that was used to generate the spreadsheet. This worksheet is empty because there were no tests defined in the table pair. If tests had been defined, then those would be visible — one test per row — in the worksheet. This worksheet should have 18 columns defined.

These correspond to fields that are present in the add test dialog in DVO:

<table>
<thead>
<tr>
<th>Column</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Function</td>
</tr>
<tr>
<td>B</td>
<td>Field_A</td>
</tr>
<tr>
<td>C</td>
<td>Expression_A</td>
</tr>
<tr>
<td>D</td>
<td>Type_A</td>
</tr>
<tr>
<td>E</td>
<td>Precision_A</td>
</tr>
<tr>
<td>F</td>
<td>Scale_A</td>
</tr>
<tr>
<td>G</td>
<td>Operator</td>
</tr>
<tr>
<td>H</td>
<td>Field_B</td>
</tr>
<tr>
<td>J</td>
<td>Type_B</td>
</tr>
<tr>
<td>K</td>
<td>Precision_B</td>
</tr>
<tr>
<td>L</td>
<td>Scale_B</td>
</tr>
<tr>
<td>M</td>
<td>Case_Insensitive</td>
</tr>
<tr>
<td>N</td>
<td>Trim_Trailing_Spaces</td>
</tr>
<tr>
<td>O</td>
<td>NullEQNull</td>
</tr>
<tr>
<td>P</td>
<td>Threshold</td>
</tr>
<tr>
<td>Q</td>
<td>Max_Bad_Records</td>
</tr>
</tbody>
</table>
The second worksheet, Fields, (available with exported sheets created by DVO 9.6 and higher) contains the Field/Column metadata based on the Table A/B definitions in the table pair.

The following image is provided for reference purposes, and can be used when creating expressions or in other situations where the column metadata is required:

![Fields worksheet example]

**Scenario 2 – Create Tests in the Spreadsheet**

The first worksheet contains all the column and test metadata (column names, test types, operators etc.) that are normally present in the client tool. This metadata is visible in dropdown menus when you click on a column.

For example, the Function column shows a drop down of all the test types (Count, Value, Outer Value, etc.) which are available. The Field_A and Field_B columns show drop downs of the corresponding columns for Table A and Table B in the table pair:

![Function and Field columns example]

This information speeds test creation in the spreadsheet, helps reduce input errors and eliminates the need to memorize items or specific column names and types. You can use the worksheet to create tests for the table pair.

1. Each row defines one test for the table pair.
2. Click the cells in each column and enter the necessary information to create tests.
3. You can create different types of tests. For example, you can create value tests, outer value, count, max and others in a single worksheet, but all tests must be for the table pair that created the spreadsheet.
4. You can make mistakes in entering the tests (e.g. syntax issues in expressions) or leave some necessary fields empty.
5. When the spreadsheet is imported into DVO, the information will be parsed, and any errors found will be indicated in the client tool. You can correct errors at that time. The following is
an example of tests defined for a table pair:

<table>
<thead>
<tr>
<th>Function</th>
<th>Field A</th>
<th>Expression A</th>
<th>Type A</th>
<th>Precision A</th>
<th>Scale A</th>
<th>Operator</th>
<th>Field B</th>
<th>Expression B</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUE</td>
<td>LASTNAME</td>
<td>= LASTNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COUNT</td>
<td>EMPLOYEEID</td>
<td>= EMPLOYEEID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUE</td>
<td>SALARY</td>
<td>= SALARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUE</td>
<td>substr(firstname, 0, 1) string</td>
<td>= EMAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. When you have finished creating the tests, save the spreadsheet.

**Scenario 3 – Import the Spreadsheet into DVO**

1. Open the DVO Navigator.
2. Right click on the same table pair used to create the empty spreadsheet and select Generate Tests from Spreadsheet:
   - A1-EMPLOYEE_DW (ETL Testing FF to Oracle)
   - DIMEMPLOYEES-DIMEMPLOYEES2
   - Joined SRCEMPLOYEES-DIMEMPLOYEES

3. Using the dialog, click **Browse**, and open the spreadsheet that you just saved:

4. The tests are imported and displayed in the dialog.

5. If you import a lot of tests (many dozens or hundreds), the import process can take some time (e.g. 30-60 seconds or even longer) as the import process validates each row of the spreadsheet and identifies any errors.
6. These errors are highlighted in the import dialog with red/yellow icons:

7. Red icons on the left indicate the test in that row has errors.

8. Yellow icons on individual cells point out specific cells that require attention.

9. Correct the errors found during the import.

10. In the image above, the last two tests have errors in them that need to be addressed.

11. You can add and remove tests using this dialog by clicking on Add and Remove:

12. This provides you with additional controls when correcting imported test definitions.

13. Click Save to generate the tests. Inspect the generated tests to verify that tests imported correctly.

   Note: All tests imported will be added to the table pair, even if the table pair already contains tests with the same definitions. Duplicate checking is not performed when creating the new tests from the spreadsheet metadata.

   Although, this is a very simple example to illustrate how to use the spreadsheet import functionality, the real power of this functionality is when many dozens or hundreds of tests are required and basic spreadsheet capabilities such as copy/paste or simple string expressions can be used in the spreadsheet to quickly create large numbers of tests.

### Spreadsheet Import for Compare Tables

The compare tables dialog has always provided an efficient way to mass create table pairs and their tests based on table metadata. This is very useful when testing scenarios such as product upgrades or migrations where the two data sets being tested are virtually the same. I.e. same table/data structures, and in theory, same data.

One limitation of compare tables was that the table pair generation process assumed that Table names must be identical across the Table A and table B metadata, and there was no way for a user to override that.

While that works for certain scenarios, there are many cases where table structures are the same, but the names are not. For example, in a migration, table names may change or in a replication, tables may be prefixed (or suffixed) across source/target to distinguish them.
The spreadsheet import for compare tables allows you to use the compare tables’ functionality but provide simple metadata for tables when the names don’t exactly match.

There are two test scenarios for the Spreadsheet import for Compare Tables:

1. Adding Tables to the Compare Tables Import spreadsheet.
2. Importing the Compare Tables spreadsheet into DVO.

Each scenario is explained in detail in the following sections.

**Scenario 1 – Adding Tables to the Compare Tables Import Spreadsheet**

The Compare Tables import spreadsheet is much simpler than the table pair import spreadsheet. It has the following seven columns:

<table>
<thead>
<tr>
<th>Column Title</th>
<th>Description</th>
<th>External_ID</th>
<th>Table_A</th>
<th>Table_B</th>
<th>Join_A</th>
<th>Join_B</th>
<th>Comments</th>
</tr>
</thead>
</table>

**Note:** Unlike the table pair example earlier in this document, you cannot create an export spreadsheet for compare tables from the client tool. Given that it only has seven columns, it can be easily recreated manually. Informatica also provides a sample spreadsheet in the following directory on the client machine:

`<DVO client install dir>/samples/CompareTables-template.xlsx`

Of these seven columns, only Columns C and D (Table_A, Table_B) are mandatory. The others are optional.

1. Open the compare tables spreadsheet:

2. Enter the necessary spreadsheet columns to define the table pairs you need to create:

   **Note:** in this spreadsheet only the table names have been entered, but you can enter as much metadata as needed including Description, External_ID, Join_A/B fields and Comments.

3. Once the spreadsheet is complete, save it.
4. Remember that tables with these names need to be defined in the folders that will be selected in the compare tables dialog. For example, in the folders specified in the following image:

![Table A and Table B comparison](image)

5. This image shows the tables (Targets) as they are defined in the PowerCenter repository. The folders DVO_MIGRATION and DVO_UPGRADE are the folders specified in the compare tables dialog:

![Folder structure](image)

Scenario 2 – Importing the Compare Tables Spreadsheet into DVO

1. In DVO, open the compare tables dialog and fill in the appropriate values at the top for Table A and Table B:
2. Select the Table names differ between source and target check box.
3. Select the spreadsheet that has the table names you want to use in the table pairs.
4. Enter the rest of the compare tables dialog as needed and click Next. DVO imports the metadata from the spreadsheet, defines the table pairs and displays them in the dialog box:

The table pairs have been defined (but not yet created), and error checking has been performed. For those table pairs with a warning, click on them and fix the problems identified at the bottom of the dialog.
5. When all issues are addressed, click **Finish** to generate the table pairs:

- A3s-Compare Tables
  - AutoDVOMIGRATION/X_DIMEMPLOYEES -- DVO_UPGRADE/DIMEMPLOYEES
  - AutoDVOMIGRATION/X_DIMPRODUCTS -- DVO_UPGRADE/DIMPRODUCTS
  - AutoDVOMIGRATION/X_FACTORDERS -- DVO_UPGRADE/FACTORDERS
  - AutoDVOMIGRATION/X_FACTSJMORDERSEYPRODUCT -- DVO_UPGRADE/FACTSJMORDERSEYPRODUCT
  - AutoDVOMIGRATION/X_MM_TIME_DAY -- DVO_UPGRADE/MM_TIME_DAY
  - AutoDVOMIGRATION/X_ORDER_BY_ITEMS -- DVO_UPGRADE/ORDER_BY_ITEMS
  - AutoDVOMIGRATION/X_ORDERDETAILS -- DVO_UPGRADE/ORDERDETAILS
  - AutoDVOMIGRATION/X_SALES_PERSON -- DVO_UPGRADE/SALES_PERSON
  - AutoDVOMIGRATION/X_STUDENT_FACT_ASG -- DVO_UPGRADE/STUDENT_FACT_ASG

**Author**

Saeed Khan  
*Principal Product Manager, PowerCenter*