How To Test Data Abstraction Best Practices

An Open Source Asset for use with TIBCO® Data Virtualization

|  |  |
| --- | --- |
| **Project Name** | AS Assets Data Abstraction Best Practices |
| **Document Location** | This document is only valid on the day it was printed. The source of the document will be found in the ASAssets\_DataAbstractionBestPractices folder (https://github.com/TIBCOSoftware) |
| **Purpose** | Self-paced instructional |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1.0 | 08/28/2013 | Mike Tinius | Initial revision |
| 8.0 | 12/05/2013 | Mike Tinius | Updated for Best Practices 8.0 |
| 8.1 | 05/05/2014 | Mike Tinius | Updated for Best Practices 8.1 and ASAssets change. |
| 8.1.3 | 08/08/2014 | Mike Tinius | Updated docs for Best Practices 8.1.3 release. |
| 8.1.4 | 08/25/2014 | Mike Tinius | Updated for Best Practices 8.1.4 release. |
| 8.1.5 | 11/26/2014 | Mike Tinius | Updated for Best Practices 8.1.5 release |
| 8.1.6 | 05/20/2014 | Mike Tinius | Updated for Best Practices 8.1.6 release |
| 8.1.7 | 11/26/2014 | Mike Tinius | Updated for Best Practices v8.1.7 – added generateViews=2 to allow generating views with a SELECT \* projection. |
| 8.1.8 | 05/24/2017 | Mike Tinius | Updated for Best Practices v8.1.8 – added Privilege scripts. |
| 8.1.9 | 12/06/2017 | Mike Tinius | Transitioned to Tibco for release 8.1.9 |

Related Documents

|  |  |
| --- | --- |
| **Name** | **Version** |
| How To Use Utilities.pdf | 2017Q4 |
| How To Use Data Abstraction Best Practices.pdf | 8.1.9 |
| How To Learn Data Abstraction Best Practices.pdf | 8.1.9 |
| How To Use Data Abstraction Best Practices Manage Annotations.pdf | 8.1.9 |
| How To Use Data Abstraction Best Practices Privilege Scripts.pdf | 8.1.9 |

Supported Versions

|  |  |
| --- | --- |
| **Name** | **Version** |
| TIBCO® Data Virtualization | 7.0 or later |
| AS Assets Utilities open source | 2017Q4 or later |

**Table of Contents**

1 Overview 4

2 Project Maintenance Functional Tests 5

3 Upgrade Project Functional Tests 7

4 Generate Views Functional Test 12

5 Perform Round Trip Test 21

Test Round Trip for “lab00” 21

Test Round Trip for “BestPracticesTestSmall” 24

1. Overview

This document provides a test plan for the Data Abstraction Best Practices. The test plan should be followed whenever the Utilities and Best Practice code is changed. In the text below, any reference to “X\_X\_X” should be replaced with the current version of the Best Practices. The test project that will be used for testing will be called “lab00”. There are also references to the major and minor version XX. In this case the point release is dropped from the folder name. These releases should be considered minor point release upgrades to the existing Data Abstraction Bes Practices folder. For example:

* /shared/BestPractices\_v81
  + The same folder is used for 8.1.1, 8.1.2, 8.1.3, 8.1.4, 8.1.5, 8.1.6, and 8.1.7
  + Each point release will overwrite the previous release.
* /shared/BestPractices\_v82 - this would represent a minor release change where the functionality was great enough to determine elicit a new folder while preserving the original folder.
  + This could be a result of structural changes to the code base or differences in the underlying Utilities parameters.

1. Project Maintenance Functional Tests

The project maintenance methods are found in the folder: **/shared/BestPractices\_vXX/\_ProjectMaintenance**

Assumptions:

* The Data Abstraction Sample was imported during installation.
  + BestPractices\_vX\_X\_X\_AS\_DataAbstractionSample81.car

The following are a list of functional tests that should be performed:

1. Get the Best Practices Version
   * **getBestPracticesVersion**()
   * **Result**: returns the current version. Example: 8.0
2. Update Impacted Resources
   * **updateImpactedBestPractices**()
   * **Result**: returns 1 for success
3. Generate Project
   * **generateProject**( “/shared/lab00”, 1, 1)
   * **Result**: Refresh Studio…generates the project specified on input.
4. Create data source metadata
   * Copy the entire folder of Metadata found at: /shared/DataAbstractionSampleXX/Physical/ Metadata to /shared/lab00/Physical
   * Copy the entire folder of Transformations found at: /shared/DataAbstractionSampleXX/Physical/ Formatting/Transformations to /shared/lab00/Physical/Formatting
   * Rebind the XSLT in Formatting/Transformations to test/Metadata…
5. Generate Configure Starting Folders
   * **generateConfigureStartingFolders**(“/shared/lab00”)
   * **Result**: Refresh Studio…generates a new ConfigureStartingFolders at /shared/lab00/\_scripts/Configure/ConfigureStartingFolders
6. Rename Project
   * **renameProject**(“/shared/lab00”, “lab00\_new”, 1, 1, 1)
   * **Result**: Refresh Studio…project has a new name of test\_new
     + **Verify /Constants/defaultValues** – basePath should be rebound to new name.
     + **Verify /Documentation/documentationTrigger\*** - parameters should be rebound to new name.
     + **Verify /Generate/generate\*** - all should be rebound to new name.
7. Move Project
   * **moveProject**(“/shared/lab00\_new”, “/shared/labs/test”, 1, 1, 1)
   * **Result**: Refresh Studio…project has been moved to a new folder
     + **Verify /Constants/defaultValues** – basePath should be rebound to new name.
     + **Verify /Documentation/documentationTrigger\*** - parameters should be rebound to new name.

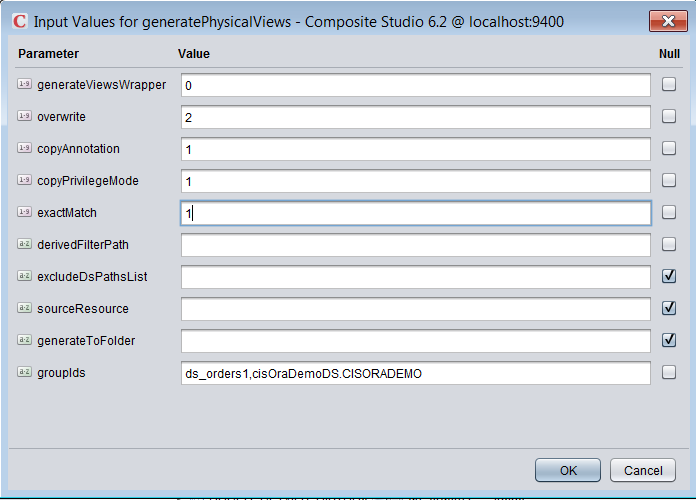
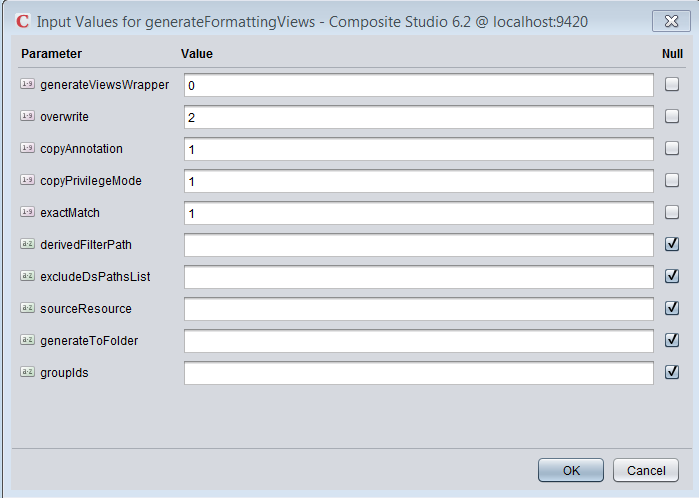
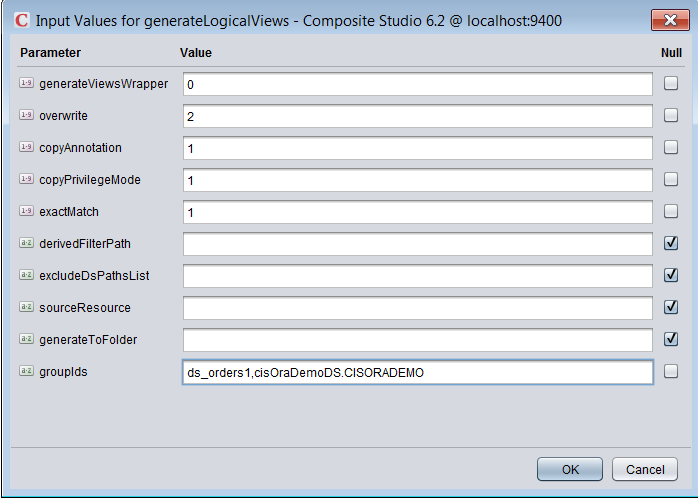
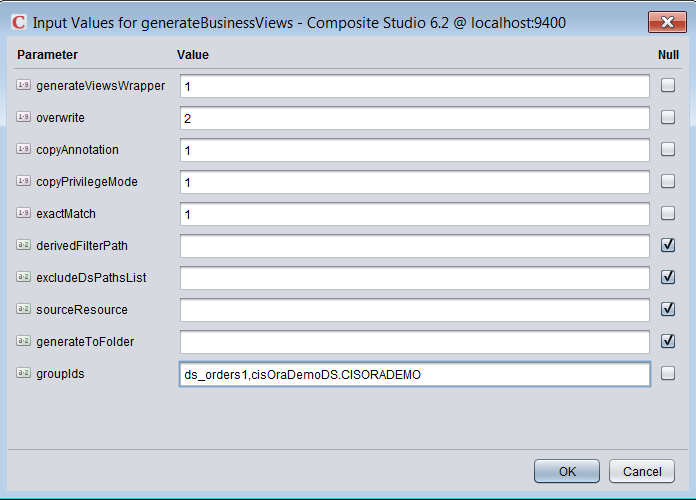
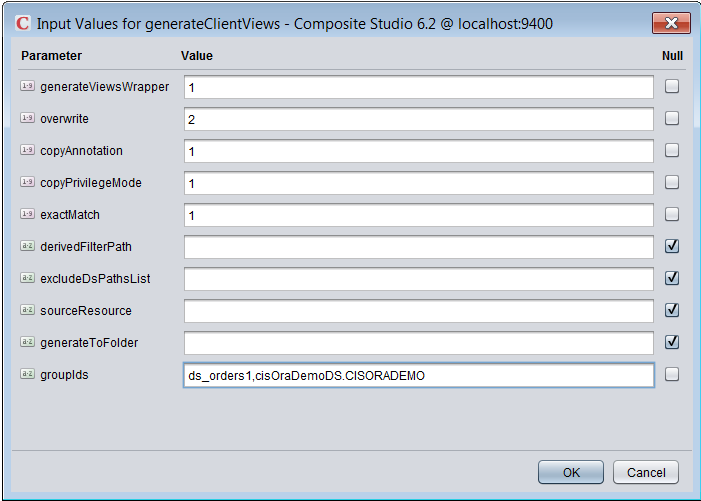
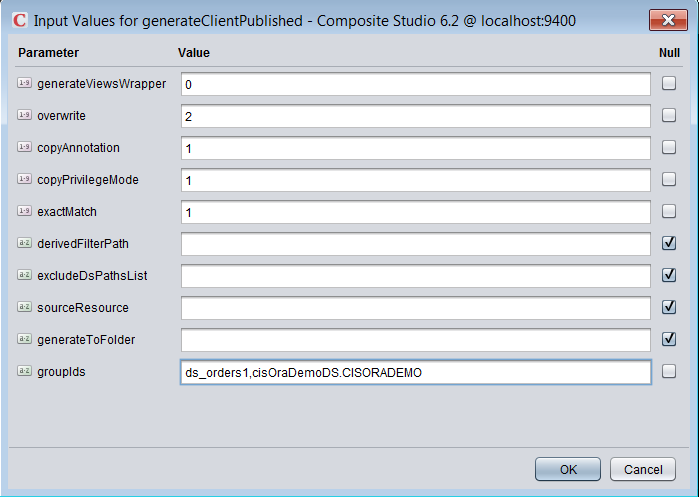
**Verify /Generate/generate\*** - all should be rebound to new name.

1. Upgrade Project Functional Tests

The upgrade project maintenance method is found in the folder: **/shared/BestPractices\_vXX/\_ProjectMaintenance**. Import sample projects are previous versions of the Best Practices from the RegressionTest folder. The following are a list of functional tests that should be performed:

1. Upgrade version 1.0
   1. Import (overwrite) sample project DataAbstractionSample10.car
   2. Import (overwrite) B.P. 3.0 – BestPractices\_generic\_v3-2010-09-30.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample10”, 1)
   4. **Result**: creates folder: /shared/DataAbstractionSample10\_vXX
2. Upgrade version 4.0
   1. Import (overwrite) sample project DataAbstractionSample40.car
   2. Import (overwrite) B.P. 4.0 – BestPractices\_v4-2011-02-08.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample40”, 1)
   4. **Result**: creates folder: /shared/DataAbstractionSample40\_vXX
   5. **Known Issues**:
      1. **/CRUD/isEmpty/isEmpty\_SupportingDocuments**
         1. **isEmptyString**(inVector[1].FileContentBlob, explicit, result);
            1. change to isEmptyBlob
         2. **isEmptyString**(inVector[1].FileContentClob, explicit, result);
            1. change to isEmptyClob
      2. /CRUD/RetrievePK/retrievePK\_SupportingDocuments
         1. **formatString**(inVector[i].FileContentBlob, 'FileContentBlob'…
            1. comment out or remove this line
         2. **formatString**(inVector[i].FileContentClob, 'FileContentClob'…
            1. change to formatClob
3. Upgrade version 5.0
   1. Import (overwrite) sample project DataAbstractionSample50.car
   2. Import (overwrite) B.P. 5.0 – BestPractices\_v5\_1-2012\_02\_15.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample50”, 1)
   4. **Result**: creates folder: /shared/DataAbstractionSample50\_vXX
4. Upgrade version 6.0
   1. Import (overwrite) sample project DataAbstractionSample40.car
   2. Import (overwrite) B.P. 6.0 – BestPractices\_v6\_0\_repaired\_ASAssets.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample60”, 1)
   4. **Result**: creates folder: /shared/DataAbstractionSample60\_vXX
   5. **Known Issues**:
      1. **/CRUD/isEmpty/isEmpty\_SupportingDocuments**
         1. **isEmptyString**(inVector[1].FileContentBlob, explicit, result);
            1. change to isEmptyBlob
         2. **isEmptyString**(inVector[1].FileContentClob, explicit, result);
            1. change to isEmptyClob
      2. /CRUD/RetrievePK/retrievePK\_SupportingDocuments
         1. **formatString**(inVector[i].FileContentBlob, 'FileContentBlob'…
            1. comment out or remove this line
         2. **formatString**(inVector[i].FileContentClob, 'FileContentClob'…
            1. change to formatClob
5. Upgrade version 6.6
   1. Import (overwrite) sample project DataAbstractionSample66.car
   2. Import (overwrite) B.P. 6.6 – BestPractices\_v6\_6\_repaired\_ASAssets.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample66”, 1)
   4. **Result**: creates folder: /shared/DataAbstractionSample66\_vXX
   5. **Known Issues**:
      1. **/CRUD/isEmpty/isEmpty\_SupportingDocuments**
         1. **isEmptyString**(inVector[1].FileContentBlob, explicit, result);
            1. change to isEmptyBlob
         2. **isEmptyString**(inVector[1].FileContentClob, explicit, result);
            1. change to isEmptyClob
      2. /CRUD/RetrievePK/retrievePK\_SupportingDocuments
         1. **formatString**(inVector[i].FileContentBlob, 'FileContentBlob'…
            1. comment out or remove this line
         2. **formatString**(inVector[i].FileContentClob, 'FileContentClob'…
            1. change to formatClob
6. Upgrade version 7.0
   1. Import (overwrite) sample project DataAbstractionSample70.car
   2. Import (overwrite) B.P. 7.0 – BestPractices\_v7\_0\_repaired\_ASAssets.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample70”, 1)
   4. **Result**: upgrades project in place: /shared/DataAbstractionSample70
   5. **Known Issues**:
      1. **/CRUD/isEmpty/isEmpty\_SupportingDocuments**
         1. **isEmptyString**(inVector[1].FileContentBlob, explicit, result);
            1. change to isEmptyBlob
         2. **isEmptyString**(inVector[1].FileContentClob, explicit, result);
            1. change to isEmptyClob
      2. /CRUD/RetrievePK/retrievePK\_SupportingDocuments
         1. **formatString**(inVector[i].FileContentBlob, 'FileContentBlob'…
            1. comment out or remove this line
         2. **formatString**(inVector[i].FileContentClob, 'FileContentClob'…
            1. change to formatClob
7. Upgrade version 7.1
   1. Import (overwrite) sample project DataAbstractionSample71.car
   2. Import (overwrite) B.P. 7.1 – BestPractices\_v7\_1\_repaired\_ASAssets.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample71”, 1)
   4. **Result**: upgrades project in place: /shared/DataAbstractionSample71
   5. **Known Issues**:
      1. **/CRUD/isEmpty/isEmpty\_SupportingDocuments**
         1. **isEmptyString**(inVector[1].FileContentBlob, explicit, result);
            1. change to isEmptyBlob
         2. **isEmptyString**(inVector[1].FileContentClob, explicit, result);
            1. change to isEmptyClob
      2. /CRUD/RetrievePK/retrievePK\_SupportingDocuments
         1. **formatString**(inVector[i].FileContentBlob, 'FileContentBlob'…
            1. comment out or remove this line
         2. **formatString**(inVector[i].FileContentClob, 'FileContentClob'…
            1. change to formatClob
8. Upgrade version 7.2
   1. Import (overwrite) sample project DataAbstractionSample72.car
   2. Import (overwrite) B.P. 7.2 – BestPractices\_v7\_2\_repaired\_ASAssets.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample72”, 1)
   4. **Result**: upgrades project in place: /shared/DataAbstractionSample72
9. Upgrade version 7.3
   1. Import (overwrite) sample project DataAbstractionSample73.car
   2. Import (overwrite) B.P. 7.3 – BestPractices\_v7\_3\_repaired\_ASAssets.car
   3. **upgradeProject**(X.X, “/shared/DataAbstractionSample73”, 1)
   4. **Result**: upgrades project in place: /shared/DataAbstractionSample73
10. Upgrade version 8.0
    1. Import (overwrite) sample project DataAbstractionSample80.car
    2. Import (overwrite) B.P. 8.0 – BestPractices\_v8\_0\_repaired\_ASAssets.car
    3. **upgradeProject**(X.X, “/shared/DataAbstractionSample80”, 1)
       1. **Result**: upgrades project in place: /shared/DataAbstractionSample80
11. Generate Views Functional Test

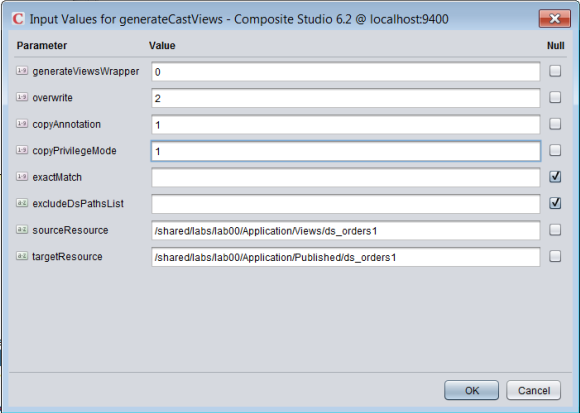
The following are a list of functional tests that should be performed. Refer to the project created in the first section. Open the scripts generate folder: **/shared/labs/lab00/\_scripts/Generate**.

1. Copy Common\_Model\_v3\_file3\_lab.xlxs to C:\CompositeSoftware\BestPractices\BestPractices\_vXX\Common\_Model\_v3\_file3.xlxs
2. Cache common\_model view before starting. Be sure that Common\_Model\_v3\_file3.xlxs contains “lab00” project entries.
3. Generate Physical Views
   * **generatePhysicalViews**(0, 2, 1, 1, 1, null, null, null, null, ds\_orders1,cisOraDemoDS.CISORADEMO )
   * 
   * **Result**: rows=152. Folders created:
     + /Physical/Physical/CISORADEMO
     + /Physical/Physical/ds\_orders1
4. Generate Formatting Views
   * **generateFormattingViews**(0, 2, 1, 1, 1, null, null, null, null, null)
   * 
   * **Result**: rows=455. All folders that match the folders in Metadata are created.
5. Generate Logical Views
   * **generateLogicalViews**(0, 2, 1, 1, 1, null, null, null, null, ds\_orders1,cisOraDemoDS.CISORADEMO )
   * 
   * **Result**: rows=153. Folders created:
     + /Business/Business/CISORADEMO
     + /Business/Business/ds\_orders1
6. Generate Business Views
   * **generateBusinessViews**(1, 2, 1, 1, 1, null, null, null, null, ds\_orders1,cisOraDemoDS.CISORADEMO )
   * 
   * **Result**: rows=153. Folders created:
     + /Business/Business/CISORADEMO
     + /Business/Business/ds\_orders1
7. Generate Client Views
   * **generateClientViews**(1, 2, 1, 1, 1, null, null, null, null, ds\_orders1,cisOraDemoDS.CISORADEMO )
   * 
   * **Result**: rows=153. Folders created:
     + /Application/Views/CISORADEMO
     + /Application/Views/ds\_orders1
8. Generate Client Published Views
   * **generateClientPublished**(0, 2, 1, 1, 1, null, null, null, null, ds\_orders1,cisOraDemoDS.CISORADEMO )
   * 
   * **Result**: rows=153. Folders created:
     + /Application/Published/CISORADEMO
     + /Application/Published/ds\_orders1
9. Generate Cast Views
   * Delete the folder /Application/Published/ds\_orders1
   * **generateCastViews**(0, 2, 1, 1, 1, null,

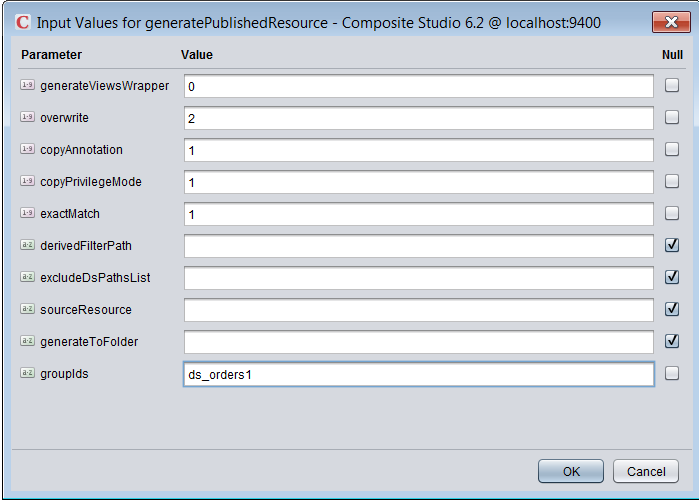
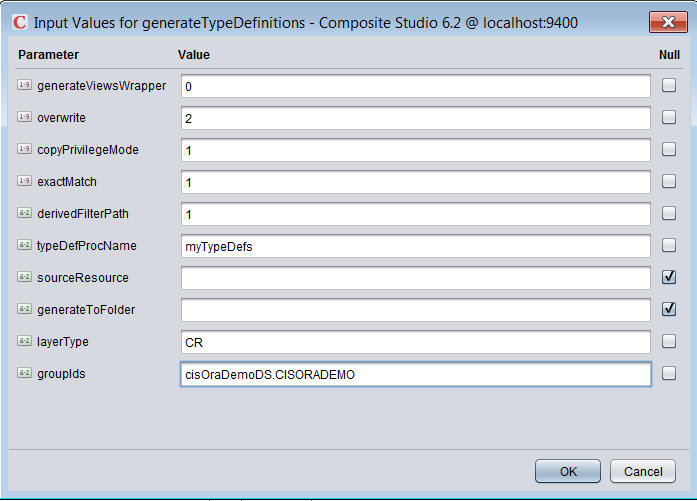
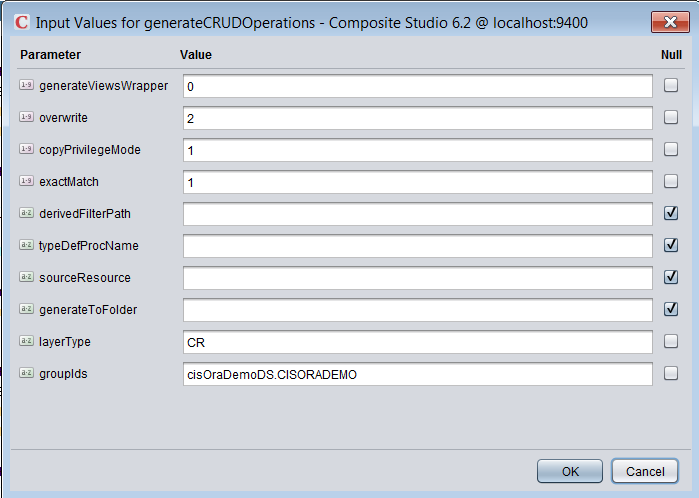
/shared/labs/lab00/Application/Views/ds\_orders1

/shared/labs/lab00/Application/Published/ds\_orders1

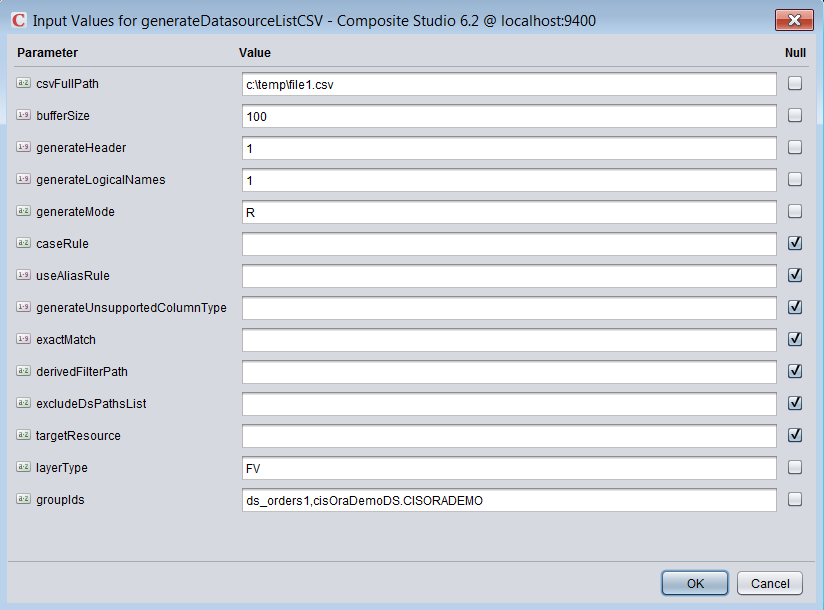
)

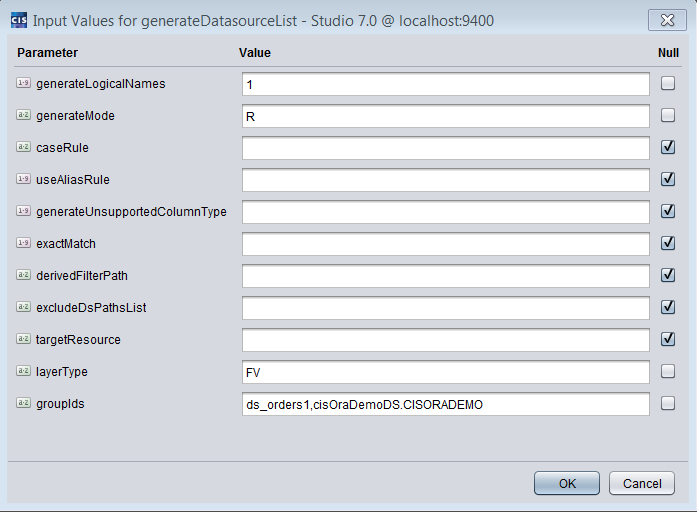


* + **Result**: rows=41. Folders created:
    - /Application/Published/ds\_orders1
    - Insure the views were created with cast statements.

1. Generate Published Database Resource
   * Create a published database called “**ds\_orders1**”
   * **generatePublishedResource**(0, 2, 1, 1, 1, null, null, null, null, ds\_orders1)
   * 
   * **Result**: rows=4. Published resources created:
     + /services/databases/ds\_orders1
       - /customers
       - /orderdetails
       - /orders
       - /shippingmethods
2. Generate Type Definitions
   * **generateTypeDefinitions**(0, 2, 1, 1, null, myTypeDefs, null, null, CR, cisOraDemoDS.CISORADEMO)
   * 
   * **Result**: rows=1. Procedure created:
     + /Application/Services/CRUD/Definitions/ myTypeDefs\_cisOraDemoDS.CISORADEMO
3. Generate CRUD Operations
   * Pre-requisite: generatePhysicalViews must have been executed with the “generateIndexes=1” variable set.
   * **generateCRUDOperations**(0, 2, 1, 1, null, null, null, null, null, CR, cisOraDemoDS.CISORADEMO)
   * 
   * **Result**: rows=103. Folders created:
     + /Application/Services/CRUD/…multiple folders
4. Generate Data Source List CSV File
   * **generateDatasourceListCSV**(

c:\temp\file1.csv, 100, 1, 1, R, null, null, null, null, null, null, null, FV, ds\_orders1,cisOraDemoDS.CISORADEMO)

* + 
  + **Result**: error=0. File is created:
    - A Common Model Best Practices CSV file gets created in the file system at C:\Temp\file1.csv

1. Generate Data Source List
   * **generateDatasourceList**(1, R, null, null, null, null, null, null, null, FV, ds\_orders1,cisOraDemoDS.CISORADEMO )
   * 
   * **Result**: rows=153.
2. Generate “Generic” Views
   * Create a custom copy of generateFormattingViews and place in the \_Custom folder
   * Modify the following parameter
     + DECLARE generateViews SMALLINT DEFAULT 2;
     + -- 0=Do not generate - (browse only) print out what will happen but don't perform the generation
     + -- 1=Do generate [DEFAULT] - Perform the VIEW Generation with a column projection.
     + -- 2=Do generate - Perform the VIEW Generation with a select \* projection.
   * **generateViews**(1, 2, 1, 1, 1, null, null, null, <S>, <T>, null)

S /shared/lab00/Physical/Formatting/Transformations/ds\_XML/productCatalog\_Transformation

T /shared/lab00/Physical/Formatting\_All/Transformations/ds\_XML/productCatalog\_Transformation

S /shared/lab00/Physical/Metadata/Excel/Common\_Model\_v3\_file2/Common\_Model\_v3\_file2.xlsx

T /shared/lab00/Physical/Formatting\_All/Excel/Common\_Model\_v3\_file2/Common\_Model\_v3\_file2.xlsx

S /shared/lab00/Physical/Metadata/OracleSource/cisOraDemoDS/CISORADEMO

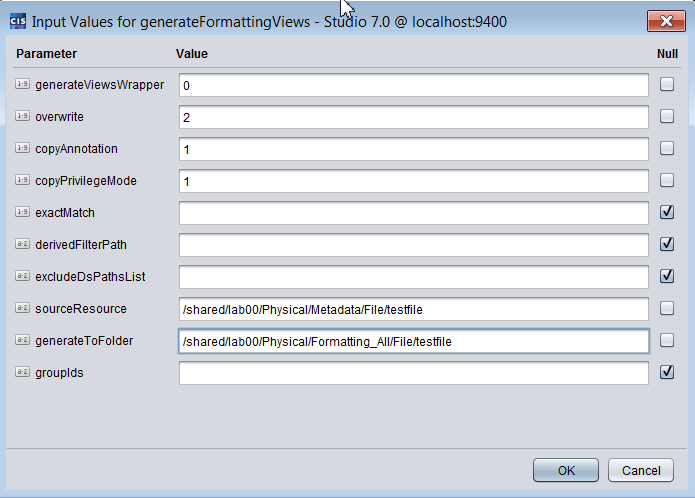
T /shared/lab00/Physical/Formatting\_All/OracleSource/cisOraDemoDS/CISORADEMO

S /shared/BestPracticesTestSmall/Physical/Metadata/OracleSource/cisOraDemoDS/CISORADEMO

T /shared/BestPracticesTestSmall/Physical/Formatting\_All/OracleSource/cisOraDemoDS/CISORADEMO

S /shared/lab00/Physical/Metadata/File/testfile

T /shared/lab00/Physical/Formatting\_All/File/testfile



* + **Result**: Views generated with SELECT \* projection in the /Formatting\_All folder.

1. Perform Round Trip Test

The following is a procedure to follow to execute a round trip test. This test will use the both the “lab00” folder previously created as well as the much larger test base from the “BestPracticesTestSmall” project. The objective of this test is to make sure the code is consistent between generating views and generating the CSV spreadsheet.

### Test Round Trip for “lab00”

1. Assumption: The cache has been refreshed.
2. Remove /shared/labs/lab00/Physical/Formatting/<all-subfolders-except-Transformations>
3. Generate Formatting Views
   * **generateFormattingViews**(0, 2, 1, 1, 1, null, null, null, null, null)
   * 
4. Generate Data Source List CSV File
   * **generateDatasourceListCSV**( C:\CompositeSoftware\BestPractices\roundtrip.csv, 1000, 1, 1, R, null, null, null, null, null, null, null, FV, null)
   * 
   * **Result**: error=0. File is created:
     + A Common Model Best Practices CSV file gets created in the file system at C:\CompositeSoftware\BestPractices\roundtrip.csv
5. Make a copy of the Formatting layer views Target=/shared/labs/lab00/Physical/Formatting\_Copy\_1
6. Copy the contents *roundtrip.csv* to *Common\_Model\_v3\_file4\_sample\_lab.xlsx*
   * Open Common\_Model\_v3\_file4\_sample\_lab.xlsx
   * Open roundtrip.csv in Excel
     + Select cell A2 and use the keystrokes: Ctrl-Shift-End (this will select Columns A-L and all rows except the header row
     + Do a Ctrl-C (copy)
   * Switch to Common\_Model\_v3\_file4\_sample\_lab.xlsx
     + Locate the cells for “lab00”
     + Place your cursor in the first cell for lab00
     + Do a Ctrl-V (Paste)
     + Save the spreadsheet
   * Refresh the cache of the spreadsheets: Common\_Model\_v3\_file[1-4].xlsx
     + Switch to Studio, Manager tab
     + Select "Cached Resources"
     + Select "common\_model"
     + Click Refresh Cache
7. Remove the views in the Formatting layer except the "/Transformations" folder which is considered a source folder for transformation procedures.
8. Generate the Formatting layer views generateFormattingViews(1, 2, 1, 0, null, null, null, null, null, null)
9. Validate the view generation process
   * validateGenerateViews(

/shared/labs/lab00/Physical/Formatting, /shared/labs/lab00/Physical/Formatting\_Copy\_1, null, 1, N, C, N)

* + **Result: There should be no rows displayed as this indicates that all views are equal and the round-trip was successful.**

### Test Round Trip for “BestPracticesTestSmall”

1. Import BestPracticesTestSmall.car.
2. Assumption: The cache has been refreshed.
3. Remove /shared/BestPracticesTestSmall/Physical/Formatting/<all-subfolders-except-Transformations>
4. Generate Formatting Views
   * **generateFormattingViews**(0, 2, 1, 1, 1, null, null, null, null, null)
   * 
5. Generate Data Source List CSV File
   * **generateDatasourceListCSV**( C:\CompositeSoftware\BestPractices\roundtrip.csv, 1000, 1, 1, R, null, null, null, null, null, null, null, FV, null)
   * 
   * **Result**: error=0. File is created:
     + A Common Model Best Practices CSV file gets created in the file system at C:\CompositeSoftware\BestPractices\roundtrip.csv
6. Make a copy of the Formatting layer views
   * Target=/shared/ BestPracticesTestSmall/Physical/Formatting\_Copy\_1
7. Copy the contents *roundtrip.csv* to *Common\_Model\_v3\_file3.xlsx*
   * Open Common\_Model\_v3\_file3.xlsx
   * Open roundtrip.csv in Excel
     + Select cell A2 and use the keystrokes: Ctrl-Shift-End (this will select Columns A-L and all rows except the header row
     + Do a Ctrl-C (copy)
   * Switch to Common\_Model\_v3\_file3.xlsx
     + Remove all cells with data for columns A-L
     + Place cursor in cell A2
     + Do a Ctrl-V (Paste)
     + Save the spreadsheet
   * Refresh the cache of the spreadsheets: Common\_Model\_v3\_file[1-4].xlsx
     + Switch to Studio, Manager tab
     + Select "Cached Resources"
     + Select "common\_model"
     + Click Refresh Cache
8. Remove the views in the Formatting layer except the "/Transformations" folder which is considered a source folder for transformation procedures.
9. Generate the Formatting layer views generateFormattingViews(1, 2, 1, 0, null, null, null, null, null, null)
10. Validate the view generation process validateGenerateViews(/shared/BestPracticesTestSmall/Physical/Formatting, /shared/BestPracticesTestSmall /Physical/Formatting\_Copy\_1, null, 1, N, C, N)
    * **Result: There should be no rows displayed as this indicates that all views are equal and the round-trip was successful.**