Data Abstraction Best Practices Installation and Release Notes

An Open Source Asset for use with TIBCO® Data Virtualization

TIBCO Software empowers executives, developers, and business users with Fast Data solutions that make the right data available in real time for faster answers, better decisions, and smarter action. Over the past 15 years, thousands of businesses across the globe have relied on TIBCO technology to integrate their applications and ecosystems, analyze their data, and create real-time solutions. Learn how TIBCO turns data—big or small—into differentiation at www.tibco.com.

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

www.tibco.com

Global Headquarters
3303 Hillview Avenue
Palo Alto, CA 94304

Tel: +1 650-846-1000
+1 800-420-8450
Fax: +1 650-846-1005

Revision History

Version	Date	Author	Comments
1.0	04/13/2010	Mike Tinius	Initial revision
2.0	07/20/2010	Mike Tinius	
3.0	08/25/2010	Mike Tinius	Added Create, Read, Update and Delete (CRUD)
4.0	02/08/2011	Mike Tinius	Improved instructions. Upgraded utilities.
5.0	11/05/2011	Mike Tinius	Simplified folder structure. Modified Excel files. Upgraded utilities.
5.1	02/15/2012	Mike Tinius	Allow for multiple schemas per data source. Modified Excel files. Upgraded utilities.
6.0	05/14/2012	Mike Tinius	Added new features. Upgraded utilities.
6.1	07/30/2012	Mike Tinius	Added controls for generating indexes.
6.2	8/6/2012	Mike Tinius	Fixed bugs
6.3	9/27/2012	Mike Tinius	Fixed issue where spaces/underscores preceding/following were trimmed.
6.4	10/01/2012	Mike Tinius	Added generateCastViews
6.5	10/30/2012	Mike Tinius	Added support for Utilties_2012Q4.car and Utilities_2012Q4_61+.car
6.6	11/15/2012	Mike Tinius	Added support for Utilties_2012Q401.car and Utilities_2012Q401_61+.car and documentation generation.
7.0	04/19/2013	Mike Tinius	Added generateDatasourceListCSV, synchronized with Utilities_2013Q104, change Common_Model_v2_file[1-3].csv spread sheet format and modified Best Practices from 4 major layers to 3.
7.1	06/04/2013	Mike Tinius	Fixed bugs found in 7.0. Added generatePublishedResource and upgradeProject procedures. Added parameters: overwrite, copyAnnotation, and copyPrivileges to all generate procedures
7.2	06/28/2013	Mike Tinius	Provided the ability to install Best Practices 7.2 in parallel with earlier releases to make migration easier.
7.3	08/30/2013	Mike Tinius	Synchronized with Utilities_2013Q301.car. Moved BestPractices_vXX folder to /shared/Utilities. Enhanced upgradeProject. Fixed header printout for generateDatasourceListCSV. Fixed performance issue with generateViews. Fixed an issue with CRUD procedure generation.
8.0	12/05/2013	Mike Tinius	Repackaged as /shared/PSAssets/BestPractices_8_0. Modified spreadsheet format.
8.1	02/21/2014	Mike Tinius	Updated with Utilities_2014Q1.car.

8.11	04/01/2014	Mike Tinius	Fixes for 8.1.1 Fixed issues with generateDatasourceList. Resolved upgrade from 8.0 to 8.1 and 8.1.1.
8.12	04/28/2014	Mike Tinius	Rebranding PS Assets as AS Assets. Requires Utilities_2014Q2.car.
8.13	08/08/2014	Mike Tinius	Updated generateViews methods with new generateCast variable features. Updated to use Utilities_2014Q3.car
8.14	08/25/2014	Mike Tinius	Updated to use Utilities_2014Q301.car and modified upgrade capabilities.
8.15	11/26/2014	Mike Tinius	Updated to use Utilities_2014Q4 + several other enhancements and fixes.
8.16	05/20/2015	Mike Tinius	Updated for Best Practices v8.1.6 – Powerpoint format only.
8.17	09/21/2015	Mike Tinius	Updated for Best Practices v8.1.7 – added generateViews=2 to allow generating views with a SELECT * projection. Requires Utilities_2015Q3.car
8.172	12/11/2015	Mike Tinius	Fixed "generateViews" to allow a derived filter path of LONGVARCHAR.
8.18	05/24/2017	Mike Tinius	Updated for Best Practices v8.1.8 – added Privilege scripts and manage annotations.
8.19	12/06/2017	Mike Tinius	Transitioned to Tibco for release 8.1.9
2018Q1	03/20/2018	Mike Tinius	Release 2018Q1 – updated to use Utilities 2018Q1. Added major capability: Dynamic File Framework. Changed references of mysql to postgres.
2019Q1	01/25/2019	Mike Tinius	Release 2019Q1 - Added the ability to handle resourceCaseRule, columnCaseRule, resourcePrefix, resourceSuffix and newColumnList for generateMode='G'.
2019Q101	01/29/2019	Mike Tinius	Release 2019Q101 – Fixed bug so derivedFilterPath could be used with generateToFolder.
2019Q200	06/13/2019	Mike Tinius	Release 2019Q200: 1. Added scriptsPath for flexibility of the location of the _scripts folder. 2. Fixed a bug in generateProject. 3. Removed all references to custom functions in favor of calling explicit paths so as to avoid name collisions at customer sites. 4. Converted as many vectors to XML as possible: ConfigParamVector and startingFolderVector. 5. Retained the original interfaces that contain vectors for backwards compatibility but the implementation procedures will use XML as input instead of vectors.

			6. Fix a bug with derivedFilterPath correlating with the list of groupIds.
2019.300	08/01/2019	Mike Tinius	Modified upgradeProject and generatProject. Fixed bug in generateDatasourceListXML(). Set minimum utilities to 2019.301. Modified all occurrences of getConstant to getConstantV2. Modified Documentation triggers, constants and added documentationDriverWrapper. Modified generateDatasourceListXMLInsertDB.
2020.200	03/12/2020	Mike Tinius	Requires Utilities 2020Q200. Fixed bug when generating a view for a PROCEDURE and generateViews=2. Changes in various procedures due to Utility changes in getBasicResourceCursor_SQL_TABLE and getBasicResourceCursor_SQL_PROCEDURE_CURSOR. Code changes DO NOT affect existing project resources.
2020.201	05/12/2020	Mike Tinius	Bug fix for pqCreate_postgres_cache_tables. Dynamic File Framework changes.
2020.300	08/20/2020	Mike Tinius	Fix for View Generation, Dynamic File and Manage Annotations.
2020.400	12/17/2020	Mike Tinius	Privilege Scripts: Fixed bugs with user creation. Added Deployment Manager to Group_List tab of spreadsheet and updated code to use it. View Generation: Fixed bugs with retrieval from commonModelCache. Updated “Learn” documentation and BestPracticesSample81 mysql to postgres data source.
2021.100	01/25/2021	Mike Tinius	Updated for changes to reintrospectDataSource
2021.101	02/18/2021	Mike Tinius	Fixed update privileges bug.
2021.200	04/08/2021	Mike Tinius	Update /shared/ASAssets/BestPractices_v81/_Installation
2021.200	05/27/2021	Mike Tinius	Added privilege procedures: getPrivilegesFormatted and getPrivilegesMultiPath.
2022.200	04/01/2022	Mike Tinius	Dynamic File modifications to pause between view generation to reduce the possibility of TDV freezing out other developers when generating large number of views.

Related Documents

Name	Version
How To Use Utilities.pdf	2021Q200

How To Use Data Abstraction Best Practices View Generation.pdf	2020Q201
How To Test Data Abstraction Best Practices View Generation.pdf	2020Q200
How To Learn Data Abstraction Best Practices View Generation.pdf	2020Q400
How To Use Data Abstraction Best Practices Manage Annotations.pdf	2020Q300
How To Use Data Abstraction Best Practices Privilege Scripts.pdf	2020Q200
How To Use Data Abstraction Best Practices Dynamic File Framework.pdf	2020Q300

Supported Versions

Name	Version
TIBCO® Data Virtualization	7.0 or later
AS Assets Utilities open source	2021Q200 or later

Table of Contents

1 Introduction

Purpose

The purpose of this document is to install the Data Abstraction Best Practices.

Audience

This document is intended to provide guidance for the following users:

1. **Architects** – Architects will achieve a deeper level of understanding of how the Best Practices scripts can be utilized from a project perspective.
2. **Developers** – Developers will learn how to use the scripts to generate views.

References

Product references are shown below. Any references to CIS or DV refer to the current TIBCO® Data Virtualization.

- TIBCO® Data Virtualization was formerly known as
  - Cisco Data Virtualization (DV)
  - Composite Information Server (CIS)

New Features

This version of the Best Practices provides the following new features:
Please visit the section "**V8.1 Modifications**" for a summary of new features.

The folder /shared/PSAssets was renamed to /shared/ASAssets. The change was to rebrand Professional Services (PS) Assets to Advanced Services (AS) Assets.

Versioning

Versioning for the Data Abstraction Best Practices is based has changed from major.minor.point release system to year.calendar_quarter such as 2018Q1. In the text below, any reference to "YYYYQnnn" represents Year+Q+quarter number. Year.Calendar Quarter [1-4]. If there is a fix/patch then it begins incrementing 401, 402, 403, etc. Any references to vXX represents the major minor version such as BestPractices_v81.

These releases should be considered minor point release upgrades to the existing Data Abstraction Best Practices folder. For example:

- /shared/BestPractices_v81

    - The same folder is used for 8.11, 8.12, 8.13, 8.14, 8.15, 8.16, 8.17, 8.18, 8.19, 2018Q*, 2019Q*,2020Q*.
    - Each point release will overwrite the previous release because changes are considered a minor improvement for the overall release with no structural changes and no parameter name or type changes.

- /shared/BestPractices_v82 - this would represent a minor release change where the functionality was great enough to determine that a new folder should be created while preserving the original folder.
    - This could be a result of structural changes to the code base or differences in the underlying Utilities parameters. Another example would when the parameters names or types change for the generation scripts.
    - In this case, we want to preserve the previous version while allowing a completely new version to be installed thus allowing the customer the chance to migrate off the previous version.
- /shared/BestPractices_v91 - this would represent a major release change where the functionality was drastically different. For example, if the structure of the spreadsheet changes, it would precipitate a drastic amount of changes that would be incompatible with any other previous release then it is prudent to have a major release.
    - One caveat to this is that if the parameters of the underlying Utilities are modified thus affecting the previous version, then we cannot guarantee the backward compatibility. In this situation, the customer must upgrade to the most current version of the Data Abstraction Best Practices that supports the Utilities.

2 Installation

How to Install

This section provides information on how to install the Best Practices scripts.

Best Practices Installation Summary

1. Download a supported version of the Utilities and Best Practices /Release zip files
2. Backup all data sources
3. Install Utilities and Best Practices Archive files
4. Install Best Practices Spreadsheets
5. Restore previously backed up and modified data sources.
6. Update all Data Source Root Paths
7. Upgrade Best Practices Scripts and Folder Structure
   7.1. Required when upgrading version 2018Q1 or prior as interfaces have changed starting with 2019Q1.
      7.1.1./shared/ASAssets/BestPractices_v81/_ProjectMaintenance/upgradeProject
   7.2. New installations bypass this step.
8. Upgrade (pre-Data Abstraction v8.0) Best Practices Spreadsheets
9. Create/Update the Spreadsheet Cache [CommonModelCache]
10. Install the Data Abstraction Sample (Optional)

Best Practices Installation

1. **Download** a supported version of the Utilities
   1.1.1.**REQUIRED**: The Utilities must be downloaded separately from Tibco Open Source Github: https://github.com/TIBCOSoftware/ASAssets_Utilities/tree/master/Release
      1.1.1.1. The supported Utilities version for the current Best Practices release is: **Utilities_2021Q200.car** or higher.
2. **Unzip** – Unzip the most current BestPractices_YYYYQnnn_Customer.zip anywhere in your file system.
3. **Backup All Data Sources/Triggers**
   3.1. **New install – bypass**.
   3.2. Backup all data sources and triggers prior to installing an upgraded version

3.3. Make a backup copy of any configured data sources and triggers if there is an existing folder: /shared/ASAssets/BestPractices_v81

3.3.1. Execute /shared/ASAssets/BestPractices_v81/_Installation/**backupAllCustomResources**

3.3.1.1. If it does not exist, then import \BestPractices_SourceCode\patches\patch_backupAllDatasources.car

3.3.1.2. Input Parameters:

3.3.1.2.1. performBackup: Y=perform the backup. N=do not perform backup

3.3.1.2.2. exportCarFilePath: Leave null for no export. Provide a full path on the TDV server to export a .car file.

3.3.1.2.3. exportEncryptionPassword: For 8.x and above. The car file encryption password.

3.3.1.2.4. Export list includes: /services/databases/ASAssets/BestPractices_v81||/shared/ASAssets/BestPractices_v81

3.4. The following is a list of resources that get backed up. There are also triggers but too many to list here.

3.4.1. View Generation Configuration:
3.4.2. /shared/ASAssets/BestPractices_v81/_ProjectMaintenance/defaultValues
3.4.3. /shared/ASAssets/BestPractices_v81/DataSource/CommonModelCache
3.4.4. /shared/ASAssets/BestPractices_v81/DataSource/CommonModelCSVSources
3.4.5. /shared/ASAssets/BestPractices_v81/DataSource/CommonModelExcelSources
3.4.6. Dynamic File Configuration:
3.4.7. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/DYNAMIC_FILE_LOCAL_LOOPBACK
3.4.8. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/DYNAMIC_FILE_QUEUE_DS
3.4.9. Manage Annotation Configuration:
3.4.10. /shared/ASAssets/BestPractices_v81/ManageAnnotations/Metadata/ManageAnnotations_EXCEL
3.4.11. Privilege Management Configuration:
3.4.12. /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_LLE_ORA
3.4.13. /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_LLE_SS
3.4.14. /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_PROD_ORA
3.4.15. /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_PROD_SS
3.4.16. /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DS_EXCEL

4. **Install Composite Utilities and Best Practices Archive files** – Install CAR files using DV Studio.

4.1. Install Utilities Archive

4.1.1. STOP: This is a new installation procedure. Read carefully.

4.1.1.1. The Utilities "**MUST**" be installed in "/shared/**ASAssets**/Utilities". Read the scenarios below to determine your situation.

4.1.1.2. Any prior versions of the Utilities (/shared/Utilities) must be moved first as per **Option 2** below.

4.1.1.3. Any prior versions of the Utilities (/shared/PSAssets/Utilities) must be renamed first as per **Option 4** below.

4.1.2. For CIS 6.0 and lower, the Best Practices are no longer supported as those releases have reached end-of-life.

4.1.3. Installation Procedure:

New Folder Structure

Many Advanced Services (AS) assets are being consolidated under a single folder, /shared/ASAssets. From this point forward, the Utilities will be distributed in a CAR file that expects this structure.

To facilitate the management of the Utilities and other AS assets moving forward, the following are some guidelines for fresh and existing installations:

[Optoin 1]: For **NEW** installs of the Utilities where no other AS assets have been installed:

- Create a folder in /shared called "ASAssets". Spelling and capitalization are important here so please use this exact spelling and case.
- Create the **published data source** folder /services/databases/ASAssets.
- Right click on the **"computername (/)" root folder** and select "Import …", choose the Utilities distribution CAR file in the resulting dialog, and click the "Import>" button.

[Option 2]: For **EXISTING** installs of the Utilities where an ASAssets folder **IS** desired:

- Create a folder in /shared called "ASAssets". Spelling and capitalization are important here so please use this exact spelling and case.
- Create the **published data source** folder /services/databases/ASAssets.
- Cut the existing Utilities folder from /shared and paste it into ASAssets. Cut and paste will ensure that any resources using the Utilities will be rebound to use the new location (this does not rebind references embedded in character string values, however.) **DO NOT COPY** the Utilities folder into ASAssets as this will not rebind dependent resources.
- Right click on the **"computername (/)" root folder** and select "Import …", choose the Utilities distribution CAR file in the resulting dialog, check the "Overwrite" checkbox, and click the "Import>" button.

- Using the "**Overwrite**" option should only overwrite the Utilities folder and leave everything else in ASAssets intact.

[Option 3]: For **EXISTING** installs of the Utilities where an **ASAssets** folder **already exists**:

- Create the **published data so**urce folder /services/databases/ASAssets if it does not already exist.
- Right click on the **"computername (/)" root folder** and select "Import …", choose the Utilities distribution CAR file in the resulting dialog, check the "Overwrite" checkbox, and click the "Import>" button.
- Using the "**Overwrite**" option should only overwrite the Utilities folder and leave everything else in "ASAssets" intact.

[Option 4]: For **EXISTING** installs of the Utilities where the former **PSAssets** folder **already exists**:

- Rename PSAssets to ASAssets
  - /shared/PSAssets → /shared/ASAssets
- Create the **published data source** folder /services/databases/ASAssets if it does not already exist.
- Right click on the **"computername (/)" root folder** and select "Import …", choose the Utilities distribution CAR file in the resulting dialog, check the "Overwrite" checkbox, and click the "Import>" button.
- Using the "**Overwrite**" option should only overwrite the Utilities folder and leave everything else in "ASAssets" intact.
- Locate the CJP data sources and perform a reintrospection on each.

4.2. Install Data Abstraction Best Practices Composite Archive file (.car)

4.2.1. The Best Practices scripts "**MUST**" be installed in "**/shared/ASAssets/BestPractices_v81**" or they will not work.

- Right click on the **"computername (/)" root folder** and import the car file with the format of ***BestPractices_YYYYQnnn.car***
- Check the "**overwrite**" box

4.2.2. Update Impacted Resources (if needed)

4.2.2.1. If any of the resources are impacted (red) then execute "**updateImpactedBestPractices**" found in "/shared/ASAssets/BestPractices_v81/_ProjectMaintenance".

4.2.2.2. No input is required as the defaults are already set for the correct parameters. Simply execute the procedure and refresh Studio when the procedure has completed.

5. Install Best Practices Spreadsheets (mandatory)

5.1. **Required**: The Best Practices v81 "must" use the new formatted spreadsheets.

5.2. Create the following directory if it does not exist.

5.2.1. Note: ***The folder location is only a suggestion. You may use any location that you want. In any of the instructions below that reference [c:/DV or /opt/DV] simply replace the suggested path with your path.***

5.2.2. WINDOWS: c:/DV

5.2.3. UNIX: /opt/DV

5.2.4. Copy the directory "**BestPractices**" from the distribution directory "\BestPractices_YYYYQnnn_Customer\BestPractices_SourceCode**BestPractices**" to the directory you created in the previous step.

5.3. Folders should look similar to this:

5.3.1. WINDOWS: c:/DV/BestPractices/**BestPractices/BestPractices_v80**

5.3.2. UNIX: /opt/DV/**BestPractices/BestPractices_v80**

5.3.3. The root path can be anything the customer wants but the /BestPractices/BestPractices_v80 is mandatory and cannot be changed. In the example above the root path for Windows is c:/DV and root path for UNIX is /opt/DV.

6. Restore data sources that were copied

6.1. **New install – bypass**.

6.2. Location: "/shared/ASAssets/BestPractices_v81/_Installation/restoreAllCustomResources"

6.3. Input Parameters:

6.3.1. performRestore: Y=perform the restore. N=do not perform restore

6.3.2. excludeTargetPaths: A comma-separated list of paths to exclude from the restore.

6.3.3. Example:
"/shared/ASAssets/BestPractices_v81/_ProjectMaintenance/defaultValues,/shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_LLE_ORA,/shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_LLE_SS,/shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_PROD_ORA,/shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DB_PROD_SS"

6.3.4. Reason: If there were updates to defaultValues or data sources, it would be required to handle these changes manually and thus they should be exluded from the restore.

6.3.5. Note: As of 2020Q402, the database schema and some columns have changes for the following resources. Therefore, it is advised not to copy over these.

/shared/ASAssets_Backup/BestPractices_Backup/PrivilegeScripts/Metadata/Privileges_DB_LLE_ORA

/shared/ASAssets_Backup/BestPractices_Backup/PrivilegeScripts/Metadata/Privileges_DB_LLE_SS

/shared/ASAssets_Backup/BestPractices_Backup/PrivilegeScripts/Metadata/Privileges_DB_PROD_ORA

/shared/ASAssets_Backup_/BestPractices_Backup/PrivilegeScripts/Metadata/Privileges_DB_PROD_SS

7. **Upgrade Best Practices Scripts and Folder Structure** (optional)

7.1. **New install – bypass**.

7.2. For a Data Abstraction Best practices version 7.3 (08/28/2013) or less go to this section otherwise bypass this section:

7.2.1. TIBCO recommends engaging with Professional Services to assist

7.2.2. Upgrade Best Practices Scripts

8. **Upgrade (pre 8.0) Best Practices Spreadsheets** (optional)

8.1. **New install – bypass**.

8.2. If the folder /BestPractices/BestPractices_v80 does not have "_v80" then go to this section otherwise bypass this section:

8.2.1. TIBCO recommends engaging with Professional Services to assist

8.2.2. Upgrade to v8.x Excel/CSV files

9. **Create the Spreadsheet Cache** (mandatory)

9.1. **Modify CommonModelCache data source** connection settings using Studio

/shared/ASAssets/BestPractices_v81/DataSource/CommonModelCache

Note: Use equivalent settings for your environment for the connection information.

- Host: localhost (default)
- **Port:** 9404 (whatever your TDV port is +4)
- Database Name: ciscache (the default cache database)
- Login: root (default)
- **Password**: (whatever your postgres cache root password is)

9.1.1. Save and Test the connection.

9.2. **Create the tables**:
/shared/ASAssets/BestPractices_v81/DataSource/CacheInstructions/pqCreate_postgres_cache_tables

9.2.1. Reintrospect CommonModelCache

9.3. **Load common_model view**

9.3.1. Load the common_model table using the following:
/shared/ASAssets/BestPractices_v81/DataSource/**common_model_load_cache**

9.3.2. Execute a common_model view to see the data

9.3.3. Configure the trigger schedule "common_model_load_trigger" for periodic loads or leave it disabled and perform this step on-demand (manually).

10. **Install the Data Abstraction Sample** (Optional)

10.1. The Data Abstraction Best Practices sample has been separated from the main code line and made an “OPTIONAL” install.

10.2. The sample is (**mandatory**) for the labs in “***How To Learn** Data Abstraction Best Practices View Generation.pdf*”

10.3. The sample “**MUST**” be installed in “/shared/DataAbstractionSample81” or it will not work.

- Right click on the **“computername (/)” root folder** and import the car file with the format of **BestPractices_YYYYQnnn_DataAbstractionSample81.car**
- Check the “**overwrite**” box

10.4. Update all Data Source Root Paths

10.4.1. Execute the procedure “**updateDatasourcePaths**“ located here:

/shared/DataAbstractionSample81/_scripts/_Custom/updateDatasourcePaths

10.4.2. Input Parameters: newRootPath

10.4.2.1. WINDOWS example: c:/DV

10.4.2.2. UNIX example: /opt/DV

10.4.3. The Excel datasource is automatically re-introspected. This is especially necessary when the server is UNIX.

10.4.4. The following paths are automatically updated to reflect the new root path as per the example path:

Data Source Resource Path	Default Root Path location
/shared/ASAssets/BestPractices_v81	
/Physical/Metadata/Excel/Common_Model_v3_file2	c:/DV/BestPractices/BestPractices/examples
/Physical/Metadata/File/Common_Model_v2	c:/DV/BestPractices/BestPractices/examples
/Physical/Metadata/File/testfile	c:/DV/BestPractices/BestPractices/examples
/Physical/Metadata/File/testfile3	c:/DV/BestPractices/BestPractices/examples
/Physical/Metadata/XML/ds_XML	c:/DV/BestPractices/BestPractices/examples

11. **Install Privilege Scripts** (Optional)

11.1. This is an optional step. Follow instructions found in “***How To Use Data Abstraction Best Practices Privilege Scripts.pdf***” to install the Privilege Scripts.

12. **Install Manage Annotations Scripts** (Optional)

    This is an optional step. Follow instructions found in "***How To Use Data Abstraction Best Practices Manage Annotation.pdf***" to install the Annotation Scripts.

3 Appendix A – Upgrade Details

How to Upgrade the Best Practices Scripts

Upgrade Best Practices Scripts and Folder Structure (optional)

Upgrade Best Practices scripts from any version between 1.0 and 7.3. Upgrade the folder structure for 1.0 through 6.6.

Version Differences and Mappings

Review the section "**Best Practices Version Differences**" for more details on what the differences are between a version and the 7.x/8.x baseline.

Review the section "**Best Practices Version Mappings**" for more details on what the mappings are between a version and the 7.x/8.x baseline.

Upgrade Project to Current Version

Any Best Practices projects starting at 1.0 or higher can be automatically upgraded to 8.0 and higher using the "upgradeProject" procedure.

For versions 1.0 through 6.6, the upgrade procedure upgrades to a copy of the original project leaving the original project and name intact. The caveat to this is that it does not repoint any of the /services/databases or /services/webserivces to the new project. This must be done manually. For /services/databases, it can be done in bulk using the "generatePublishedResource" procedure which will "re-publish" a folder of views to a target folder. It does not re-create any cached view settings. It does not set any custom procedures.

For versions 1.0 through 5.2, it will generate a new ConfigureStartingFolders() procedure based on the data sources discovered in the Physical/Metadata folder. It does not migrate the existing ConfigureStartingFolders() forward. You may need to manually tweak the procedure once the upgrade has completed.

For versions 6.0 through 6.6, the upgrade procedure will migrate the existing ConfigureStartingFolders() procedure to the new project. It is always recommended to verify that the folders are consistent.

For versions 7.0 through 7.3, it will upgrade the project in place by upgrading the "_scripts" folder. It will create a backup copy of the original folder "_scripts_Copy_1". It wil migrate the existing ConfigureStartingFolders() procedure to the new group format. The upgrade will iterate through lthe layer folders to upgrade any references to the /shared/Utilities and change to /shared/ASAssets/Utilities. Lastly, the CRUD services will be upgraded as well. Since the project is updated in-place there are no issues with cached-view settings or custom procedures as all of those remain in-place.

Instructions

These are the upgrade instructions. Note: Any place where a capital “XX”, “X_X” or “X.X” is referenced, it refers to the current Best Practices version. For example if the current version is 8.1 then the references would be “81”, “8_1” or “8.1”.

1. **Update to the Common Model v3 spreadsheet**

   1.1. For 1.0 through 7.3…

      1.1.1. Refer to the section “Upgrade to v8.x Excel/CSV files” for detailed instructions on how to migrate the excel spreadsheets.

2. **Backup project**

   2.1. For all versions…Export the project as a .car file in order to preserve the original resources and settings.

3. **Upgrade Project**

   3.1. Pre-requisites:

      3.1.1. When upgrading projects to 8.0 and higher it is required to have only the version that the project being upgraded is based upon present in Composite.

      3.1.2. Import (overwrite) all necessary Best Practices car files.

         3.1.2.1. For 6.0, import “**BestPractices_v6_0_repaired_ASAssets.car**” from \BestPractices_SourceCode\For_Upgrade_Only_to_8.1

         3.1.2.2. For 6.6, import “**BestPractices_v6_6_repaired_ASAssets.car**” from \BestPractices_SourceCode\For_Upgrade_Only_to_8.1

         3.1.2.3. For 7.0, import “**BestPractices_v7_0_repaired_ASAssets.car**” from \BestPractices_SourceCode\For_Upgrade_Only_to_8.1

         3.1.2.4. For 7.1, import “**BestPractices_v7_1_repaired_ASAssets.car**” from \BestPractices_SourceCode\For_Upgrade_Only_to_8.1

         3.1.2.5. For 7.2, import “**BestPractices_v7_2_repaired_ASAssets.car**” from \BestPractices_SourceCode\For_Upgrade_Only_to_8.1

         3.1.2.6. For 7.3, import “**BestPractices_v7_3_repaired_ASAssets.car**” from \BestPractices_SourceCode\For_Upgrade_Only_to_8.1

         3.1.2.7. For 8.0, import “**BestPractices_v8_0_repaired_ASAssets.car**” from \BestPractices_SourceCode\For_Upgrade_Only_to_8.1

   3.2. Execute the procedure “**upgradeProject**” found in the folder: /shared/ASAssets/BestPractices_vXX/_ProjectMaintenance.

      3.2.1. upgradeToVersion – specify the version to upgrade to or leave null to upgrade to the latest version. Example: 8.13

      3.2.2. projectPath - the full project path that is to be upgraded

         e.g. /shared/PROJECT1

3.2.3. generateTestFolder - determine whether to generate the Test folder or not

1=generate Test folder,

0 or null=do not generate Test folder (default)

3.3. The ConfigureStartingFolders() procedure:

3.3.1. For versions 1.0 through 5.2, you may need to manually tweak the procedure once the upgrade has completed.

3.3.2. For versions 6.0 through 6.6, it is recommended to verify that the folders are consistent from the old and new project.

4. **Click refresh in Composite Studio** when finished and review the project "_scripts" folder for changes.

5. **Upgrading CRUD procedures prior to 7.2 (Known Issues)**

5.1. This is only necessary if there are CRUD (Create, Read, Update, and Delete) procedures. Generally, they are located in the project at /Applications/Services/CRUD. Look for any sub-folders with procedures in them or impacted procedures. If none, then proceed to step 6.

5.2. RetrievePK procedures – look for any impacted procedures. You will need to modify any RetrievePK procedures containing CLOB or BLOB column types.

5.2.1. For CLOB column types, change the procedure formatString to formatClob.

5.2.2. For BLOB column types, comment out or remove the line of code referring to the BLOB column. There is no formatting for BLOB column types.

5.3. Update procedures – you may need to modify any Update procedures containing CLOB or BLOB column types. In general, most of the Best Practices versions have resolved any issues by providing a separate section of code for updating CLOB and BLOB.

5.3.1. For both CLOB and BLOB, if there are any lines starting with "set updateClause = formatString(…" and are referring to CLOB or BLOB, comment out or remove the lines.

5.4. isEmpty procedures – look for any impacted procedures. You will need to modify any isEmpty procedures containing CLOB or BLOB column types.

5.4.1. For CLOB column types, change the procedure isEmptyString to isEmptyClob.

5.4.2. For BLOB column types, change the procedure isEmptyString to isEmptyBlob.

6. **Spot check the following procedures to insure the upgrade was successful:**

6.1. Open **<project_path>**/_scripts/Documentation/constants

You should see "DECLARE utilitiesRootPath …"

6.2. Open **<project_path>**/_scripts/Constants/defaultValues and execute

You should see: bestPracticesVersion_ X.X *←your current version*

6.3. Open /_scripts/Generate/generateFormattingViews – the following line should have your project path where you see <project_path>:

set basePath = **<project_path>**/"_scripts"/Constants/defaultValues.basePath;

Also, various procedures in the script should be rebound to /shared/ASAssets/BestPractices_vXX

6.4. All custom scripts moved to **<project_path>**/_scripts/Custom

7. Finishing the upgrade

7.1. Test the generate scripts to insure they are working properly

7.2. If you are satisfied that the upgrade is working correctly:

7.3. For 1.0 through 6.6…

7.3.1. Export the original Project as a backup

7.3.2. Remove the original project folder

7.3.3. Rename the new project folder to the old name so that the published views will be rebound to the new project.

7.4. For 7.0 and higher….

7.4.1. Remove /_scripts_Copy_1

8. Reverting a failed upgrade

8.1. For 1.0 through 6.6…

8.1.1. The original project is left untouched so there is no need to revert.

8.2. For 7.0 and higher….

8.2.1. Option 1: re-import the backup that you took or…

8.2.2. Option 2: If a copy of "**_scripts**" folder was created, then remove _scripts and rename the "first" copy "**_scripts_Copy_1**" back to "_scripts".

9. What gets updated in this release:

9.1. Make a copy of $PROJECT_PATH_DST/_scripts

9.2. Copy DataAbstraction_GENERIC_Template/_scripts/Constants to $PROJECT_PATH_DST/_scripts/Constants

9.3. Copy DataAbstraction_GENERIC_Template/_scripts/Display to $PROJECT_PATH_DST/_scripts/Display

9.4. Copy DataAbstraction_GENERIC_Template/_scripts/Generate to $PROJECT_PATH_DST/_scripts/Generate

9.5. Update PROJECT_PATH/_scripts/Constants/defaultValues

9.6. Update PROJECT_PATH/_scripts/Documentation/constants

9.7. UpdateTrigger PROJECT_PATH/_scripts/Documentation

9.8. Configure starting Folders PROJECT_PATH/_scripts/Configure

9.9. Rebind all resources starting at PROJECT_PATH/_scripts from Previous Utilities path (/shared/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.10. Rebind all resources starting at PROJECT_PATH/_scripts from New Utilities path (/shared/ASAssets/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.11. Rebind all resources starting at PROJECT_PATH/Application from Previous Utilities path (/shared/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.12. Rebind all resources starting at PROJECT_PATH/Application from New Utilities path (/shared/ASAssets/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.13. Rebind all resources starting at PROJECT_PATH/Business from Previous Utilities path (/shared/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.14. Rebind all resources starting at PROJECT_PATH/Business from New Utilities path (/shared/ASAssets/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.15. Rebind all resources starting at PROJECT_PATH/Physical from Previous Utilities path (/shared/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.16. Rebind all resources starting at PROJECT_PATH/Physical from New Utilities path (/shared/ASAssets/Utilities/BestPractices_v73) to Best practices 8.1 (/shared/ASAssets/BestPractices_v81)

9.17. Update CRUD procedures within the destination project path

How to Upgrade the Excel Spreadsheets

Upgrade to v8.x Excel/CSV files

This section describes how to upgrade the Excel files from any Best Practices version to Best practices 8.x baseline. The basic idea is to first upgrade the project and then generate the spreadsheet. This will be relatively easy as v7.0 introduces the capability to generate the spreadsheet from the existing Formatting sub-layer.

Instructions

These are the upgrade instructions.

1. **Upgrade the project**

   1.1. Refer to the section "How to Upgrade the Best Practices Scripts" for detailed instructions on upgrading any version of the Best Practices to the current 7.x baseline.

2. **Update Excel Files**

   2.1. Refer to the section "***Physical to Logical Mappings***" in the document "***How To Use Data Abstraction Best Practices View Generation.pdf***" for editing the Common_Model_v3_file[1-3].xlsx files.

   2.2. Copy Excel templates from the Best Practices distribution zip file found in /BestPractices_V8_x_x_Customer/BestPractices_SourceCode/BestPractices.

   2.3. Generate the Best Practices CSV file from the existing Formatting Views. Execute the procedure "generateDatasourceListCSV" with the following parameters:

      2.3.1. csvFullPath=C:/Temp/Common_Model_v3_file.csv

      2.3.2. bufferSize=1000

      2.3.3. generateHeader=1

      2.3.4. generateLogicalNames=1

      2.3.5. generateMode=R

      2.3.6. targetResource=/shared/<project_name>/Physical/Formatting

      2.3.7. layerType=check the null box

      2.3.8. groupIds=check the null box

      2.3.9. derivedFilterPath=check the null box

   2.4. Transfer the results of the generated CSV to your Excel (.xlsx) file

      2.4.1. Take a backup of your BestPractices folder before beginning

      2.4.2. Open Common_Model_v3_file1.xlsx

      2.4.3. Open C:/Temp/Common_Model_v3_file.csv

         2.4.3.1. Except for the header row, select all non-empty rows in the range of columns A-L

            2.4.3.1.1. Place your cursor in cell A2

            2.4.3.1.2. Press Control-Shift-End key to go to the bottom

2.4.3.1.3. All non-empty rows and columns A-L should now be selected

2.4.3.1.4. Press Control-C to copy

2.4.4. Switch over to the Common_Model_v3_file1.xlsx spreadsheet

2.4.4.1. Position the cursor in A2

2.4.4.2. Press Control-V to paste the contents of the copied buffer

2.4.4.3. Press Control-End to go to the end and then scroll left (Home key).

2.4.4.4. If your rows have exceeded the default 1040 rows then you will need to extend the spreadsheet formulas which exist in columns J through R.

2.4.4.4.1. Copy columns M-Z for row 1040

2.4.4.4.2. Paste this column until you reach the end of the pasted values

2.4.5. Save the Excel Common_Model_v3_file1.xlsx spreadsheet

2.4.6. If Save as CSV is required then follow these steps otherwise bypass this section.

2.4.6.1. Select “Save As” and choose “Save as type” CSV (Comma delimited) (*.csv) →

2.4.6.2. click Save to save as Common_Model_v3_file1.csv

2.4.6.3. Answer yes to the prompt to overwrite

2.4.6.4. Answer yes to the prompt regarding compatible features

2.4.6.5. Close the spreadsheet

2.4.6.6. Answer “Don’t Save’ to the prompt

2.5. Cache the spreadsheet results into the Postgres database.

4 Appendix B – Release Notes

Best Practices Script Modifications

2022Q200 Modifications – Apr 1, 2022

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Dynamic File modifications to pause between view generation to reduce the possibility of TDV freezing out other developers when generating large number of views.

    1.1. [MOD] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileCreate

    1.2. [MOD] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/constants

2021Q200 Modifications – May 27, 2021

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

2. Privileges

    2.1. [NEW] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/getPrivilegesFormatted

    2.2. [NEW] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/getPrivilegesMultiPath

2021Q200 Modifications – April 8, 2021

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

3. General updates

    3.1. [MOD] /shared/ASAssets/BestPractices_v81/_Installation

2021Q101 Modifications – February 18, 2021 [updated 4/6/21]

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

4. General updates

    4.1. Fixed bug with updatePrivilegesDriver to set ancestor folder correctly when encountering a datasource.

4.2. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Helpers/updateResourcePrivilegesDriverImplV4

2021Q100 Modifications – January 25, 2021

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

5. General updates

5.1. Modified for changes to /shared/ASAssets/Utilities/repository/reintrospectDataSource

5.2. Requires ASAssets Utilities 2021.100

5.3. [MOD] /shared/ASAssets/BestPractices_v81/_Installation/updateDatasourcePaths

5.4. [MOD] /shared/ASAssets/BestPractices_v81/_ProjectMaintenance/defaultValues

2020Q400 Modifications – December 17, 2020

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

6. Privilege Scripts

6.1. Modified to add Deployment_M column to the spreadsheet tab Group_List.

6.2. Removed the UserName_lowercase from the User_List tab. Not needed for anything.

6.3. Required Modifications to existing project:

6.3.1. 1. Required to add "Deployment_M" column to the Group_List tab after the "User_M" tab.

6.3.2. 2. Reintrospection is required for /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DS_EXCEL

6.4. [MOD] /shared/ASAssets/BestPractices_v81/_ProjectMaintenance/defaultValues

6.4.1. DECLARE PUBLIC minVersion CONSTANT DOUBLE DEFAULT 2020.402;

6.5. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Helpers/TypeDefinitions

6.6. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/ModifyDBPrivileges - all procedures were modified

6.7. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/DDL - all procedures were modified

6.8. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DS_EXCEL/Resource_Privileges_LOAD_DB.xlsx/Group_List

6.9. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Metadata/Privileges_DS_EXCEL/Resource_Privileges_LOAD_DB.xlsx/User_List

6.10. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/Formatting - all user and group related views and procedures

6.11. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/getGroups

6.12. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/updateGroupsDriver

6.13. [MOD] /shared/ASAssets/BestPractices_v81/PrivilegeScripts/validateGroupsDriver

7. View Generation

7.1. How To Learn Data Abstraction Best Practices View Generation.pdf

7.2. [MOD] /shared/ASAssets/BestPractices_v81/Procedures/retrieveName/getNamesRetrievedContainer

7.3. [MOD] /shared/ASAssets/BestPractices_v81/Procedures/retrieveName/getNameRetrieved

7.4. [MOD] /shared/ASAssets/BestPractices_v81/Procedures/retrieveName/retrieveNewColumnList

2020Q300 Modifications – August 28, 2020

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

8. View Generation

8.1. MOD] /shared/ASAssets/BestPractices_v81/Procedures/projectMaintenance/upgradeProjectProcedures/copyResources

8.2. [NEW] /shared/ASAssets/BestPractices_v81/_Installation/backupAllDatasources

8.3. Changed postgres cache CommonModelCache from cache to regular postgres datasource.

8.4. [MOD] /shared/ASAssets/BestPractices_v81/DataSource/CommonModelCache

8.5. [MOD] /shared/ASAssets/BestPractices_v81/DataSource/common_model

8.6. [MOD] /shared/ASAssets/BestPractices_v81/DataSource/CacheInstructions/pqCreate_postgres_cache_tables

8.7. [NEW] /shared/ASAssets/BestPractices_v81/DataSource/common_model_load_trigger

8.8. [NEW] /shared/ASAssets/BestPractices_v81/DataSource/common_model_load_cache

9. Dynamic File – focus on fixes for SQL Server blocking

9.1. [MOD] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/01_pqCreateDrop_dynamic_file_tables

9.2. [MOD] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/constants.EXECUTE_DML_PACKAGE_PATH

9.3. [MOD] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/02_display_DYNAMIC_FILE_QUEUE

9.4. [NEW] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/02_display_DYNAMIC_FILE_QUEUE_pq

9.5. [MOD] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileQueueAutoPublish

9.6. [MOD] /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileQueueInsert

10. Manage Annotations

10.1. [MOD] /shared/ASAssets/BestPractices_v81/ManageAnnotations/Metadata/ManageAnnotations_EXCEL

10.1.1. Eliminated non-essential columns from the spreadsheet.

10.2. [MOD] /shared/ASAssets/BestPractices_v81/ManageAnnotations/Formatting/ResourceAnnotations

10.2.1. Eliminated non-essential columns from the select statement.

10.3. [MOD] /shared/ASAssets/BestPractices_v81/ManageAnnotations/Helpers/generateResourceList

10.3.1. Simplified logic.

10.4. [MOD] /shared/ASAssets/BestPractices_v81/ManageAnnotations/generateResourceListToCSV

10.4.1. Modified logic to only output ResourcePath, PhysicalName, PhysicalType, ResourceAnnotation. Simplified the output.

10.5. [MOD] /shared/ASAssets/BestPractices_v81/ManageAnnotations/updateAnnotations

10.5.1. Fixed bug where columnAnnotations were not being concatenated. Added SUCCESS status.

10.6. [DEL] /shared/ASAssets/BestPractices_v81/ManageAnnotations/Helpers/generateResourceListToCSV

2020Q201 Modifications – May 12, 2020

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

11. 1. Bug Fix

11.1. /shared/ASAssets/BestPractices_v81/DataSource/CacheInstructions/pqCreate_postgres_cache_tables [compositeTransformation to logicalTransformation].

12. Dynamic File Framework

12.1. [MOD] Modified status to allow for different scenarios with published, metadata and file. See documentation.

12.2. [NEW] Auto-delete files when removed from file system.

12.3. [NEW] ODATA fix - publish views with underscores and no special characters like "-" or ".". Only alpha-numeric and underscore are allowed.

12.4. [NEW] Added ENVIRONMENT_NAME to constants instead of getting from getEnvName().

12.5. [NEW] Added PROJECT_DEFAULT_EMAIL to project constants procedure for exception messages.

12.6. [MOD] Changed call logDebugMessage() to call PRINT().

12.7. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/Example/example_Create_or_Remove

12.8. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/Helper/deleteFile

12.9. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/constants [added ENVIRONMENT_NAME]

12.10. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileCleanup

12.11. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileCreate

12.12. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileList [Modified status message]

12.13. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileQueueAutoPublish

12.14. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileQueueInsert

12.15. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileQueueProcess

12.16. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/dynamicFileRemove

12.17. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/initCreateResources

12.18. /shared/ASAssets/BestPractices_v81/DynamicFileFramework/initRemoveResources

2020Q200 Modifications – March 12, 2020

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

13. Requires Utilities 2020.200

    13.1. /shared/ASAssets/BestPractices_v81/_ProjectMaintenance/defaultValues

14. Added "columnNum" into cursor for TypeDefinitions.columnListRow and generateDatasourceListXML:

    14.1. The "columnNum" is being published from the following now:

        14.1.1. /shared/ASAssets/Utilities/repository/getBasicResourceCursor_SQL_TABLE

        14.1.2. /shared/ASAssets/Utilities/repository/getBasicResourceCursor_PROCEDURE_CURSOR

15. Modified where clause for the following:

    15.1. /shared/ASAssets/BestPractices_v81/Procedures/generateViewsLoopXML

    15.2. /shared/ASAssets/BestPractices_v81/Procedures/validategenerateViews

    15.3. /shared/ASAssets/BestPractices_v81/Procedures/sqlParser/utility/retrieveColumnList

    15.4. /shared/ASAssets/BestPractices_v81/ManageAnnotations/updateAnnotations

16. Fixed bug when generating a view for a PROCEDURE and generateViews=2.

    16.1. Before it would generate a SELECT * but that is not valid for procedures.

    16.2. It must generate a projection list when the source resource is a procedure regardless of generateViews=2

2019Q300 Modifications – August 01, 2019

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

17. Fixed bugs in generateDatasourceListXML:

    17.1. Modified to account for null "inDerivedFilterPath" parameter which through an exception.

    17.2. Fixed bug where duration was only calculated when debug2=Y. Now it is calculated all the time.

    17.3. Fixed bug where new columns were generated when the resource was not a valid used resource.

18. Modified all occurrences of getConstant to getConstantV2. Deprecating getConstant.

19. Set minimum Utilities requirement to 2019.301 due to getConstant() and modifyConstnt() modification.

20. Modified upgradeProject() and determineBestPracticesVersion() to more accurately detect version.

21. Added upgradeRestoreDefaultValues() called by updateProject() to restore original values from the _scripts_Copy_nn/Constants/defaultValues() to the current defaultValues().

22. Added upgradeRestoreDocumentationConstants() called by updateProject() to restore original values from the _scripts_Copy_nn/Documentation/constants() to the current constants().

23. Added an entry to upgrade 2019.200 to 2019.300 in vector_masterUpgradeVector() and vector_masterUpgradeVector().

24. Modified generateProject to work with new documentation triggers in Best Practices Template.

25. Modified all procedures to place the comments above the variable instead of below.

26. Modified generateDatasourceListInsertDB() to add an input parameter "performDeleteProjectRows" to allow deleting all project rows before inserting.

27. Modified generateDatasourceListXMLInsertDB() base code to allow for new parameter "performDeleteProjectRows".

28. Modified the Documentation Triggers to reduce the number of parameters and remove path dependencies. Modified Documentation constants. Added documentationDriverWrapper to resolve paths dynamically.

2019Q200 Modifications – Jun 13, 2019

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

29. Added scriptsPath for flexibility of the location of the _scripts folder.

30. Fixed a bug in generateProject.

31. Removed all references to custom functions in favor of calling explicit paths so as to avoid name collisions at customer sites.

32. Converted as many vectors to XML as possible: ConfigParamVector → ConfigParamXML and startingFolderVector → startingFolderXML.

33. Retained the original interfaces that contain vectors for backwards compatibility but the implementation procedures will use XML as input instead of vectors.

34. Fix a bug with derivedFilterPath correlating with the list of groupIds.

2019Q101 Modifications – Jan 29, 2019

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

35. Fixed bug so derivedFilterPath could be used with generateToFolder.

    Allows for the ability to pass in a comma-separated list of tables to generate.

    The issue before was that it only generated the first resource in the list and not the rest of them.

    Correlate the groupId position with the derivedFilterPath position

    ```
    derivedFilterPath="customers,orders",shippingmethods
                                  |              |
                                  v              v
            groupId=ds_inventory.tutorial,ds_orders.tutorial
    ```

2019Q1 Modifications – Jan 25, 2019

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

36. Added the ability to handle resourceCaseRule [formerly caseRule], columnCaseRule, resourcePrefix, resourceSuffix and newColumnList for generateMode='G'.

    This functionality is supported for both generatViews and its various derivatives and generateDatasourceList and its derivatives.

    resourceCaseRule - case rule for tables.

    columnCaseRule - case rule for columns.

    resourcePrefix - prefix for tables.

    resourceSuffix - suffix for tables.

    newColumnList - inject a new columns at the end of each table being generated.

        newColumnList Format:
        column1&&type1&&value1//column2&&type2&&value2//column3&&type3&&value3

The delimiters were chosen strategically because commas and pipes may appear as a transformation within the value.

37. Added the ability to read physical to logical mappings from the postgres cache database.

/shared/ASAssets/BestPractices_v81/_ProjectMaintenance/defaultValues

/shared/ASAssets/BestPractices_v81/DataSource/CommonModelCache

/shared/ASAssets/BestPractices_v81/DataSource/pCommon_Model_Union

/shared/ASAssets/BestPractices_v81/DataSource/pkg_create_tables

/shared/ASAssets/BestPractices_v81/DataSource/common_model

38. Added the ability to generate physical to logical mappings and save into the postgres cache database.

Implementation:

/shared/ASAssets/BestPractices_v81/Procedures/generateDatasourceListInsertDB

Interface:

/shared/ASAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Generate/generateDatasourceListInsertDB

4. Upgrade Projects:

a. If any pre-existing data abstraction best practices folders have been created the installer "MUST" execute the upgrade project on each of them as the interface procedures have changed and the current installed ones will be impacted. Additionally, new variables have been added to several

generate procedures. Resources affected:

/shared/<project>/_scripts/Configure/ConfigureParams - structure changes causing impact in old version.

/shared/<project>/_scripts/Generate/generateViews - parameter changes causing impact in old version.

/shared/<project>/_scripts/Generate/generateDatasourceList - parameter changes causing impact in old version.

/shared/<project>/_scripts/Generate/generateDatasourceListCSV - parameter changes causing impact in old version.

/shared/<project>/_scripts/Generate/generateDatasourceListInsertDB - new procedure. no impact to old version.

/shared/<project>/_scripts/Generate/generateBusinessViews - internal variables added for new features. no impact to old version

/shared/<project>/_scripts/Generate/generateCastViews - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generateClientPublished - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generateClientViews - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generateCRUDOperations - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generateFormattingViews - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generateLogicalViews - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generatePhysicalViews - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generatePublishedResource - internal variables added for new features. no impact to old version.

/shared/<project>/_scripts/Generate/generateTypeDefinitions - internal variables added for new features. no impact to old version.

b. Execute /shared/ASAssets/BestPractices_v81/_ProjectMaintenance/upgradeProject()

2018Q1 Modifications – March 20, 2018

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Add new major capability: Dynamic File Framework
2. Removed getEnv and setEnv and called the /lib/util functions directly.
3. Replaced RegexSplit with extractDelimitedText throughout.
4. Replaced /shared/ASAssets/Utilities/repository/resourceExists with /lib/resource/ResourceExists throughout.
5. Replaced references to mysql with postgres
6. Requires Utilities 2018Q1.
7. Changed release number to YYYYQnnn. Year.Calendar Quarter [1-4]. If there is a fix/patch then it begins incrementing 401, 402, 403, etc

V8.19 Modifications – December 22, 2017

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Transitioned to TIBCO Software Open Source.

V8.18 Modifications – May 26, 2017

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Added Privilege Scripts and privilege spreadsheet to the Best Practices.

2. Added Manage Annotations and annotation spreadsheet to the Best Practices.

V8.172 Modifications – December 11, 2015

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Fixed "generateViews" to allow a derived filter path longer than VARCHAR(1024). Changed to LONGVARCHAR.

V8.17 Modifications – September 21, 2015

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Procedure: generatViews

    1.1. Parameter: Added generateViews=2 to allow generating views with a SELECT * projection list instead of a column projection list.

        1.1.1. 0=Do not generate - (browse only) print out what will happen but don't perform the generation

        1.1.2. 1=Do generate [DEFAULT] - Perform the VIEW Generation with a column projection.

        1.1.3. 2= Do generate - Perform the VIEW Generation with a select * projection.

2. Validated with CIS 7.0.x

3. Fixed procedures to work with Utilities_2015Q3.car or higher.

V8.16 Modifications – May 20, 2015

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Validated with CIS 7.0.x

2. Updated Powerpoint format to Cisco format

3. Works with Utilities_2014Q4.car or higher

V8.15 Modifications – November 26, 2014

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Works with Utilities_2014Q4.car or higher

1.1. Updated to work with new updateTrigger procedure in Utilities which has a new signature.

2. Fixed output message to print the line instead of null when view is SKIPPED.

3. Increased size of logicalStatus from 255 to 1024 and fixed it to output a SKIPPED message no greater than 1024 characters.

4. Fixed generateDatasourceListCSV parsing

   4.1. parseColumnExpression() goes into infinite recursion trying to parse the expression: extract(day from SNAPSHOT_MTH + 32)

   4.2. Lines commented using single-line comments (--) are still being processed. If commented line happens to contain an invalid resource/table, thegetBasicResourceXML() fails with "Resource does not exist" message

   4.3. getUsedResourcesXML() throws an exception when the resource being analyzed has a cache which is not properly configured. A case is open with support to fix getUsedResources admin API (CIS-50706).

   4.4. When the 'from' keyword in view definition is in lower-case, extractSQLParts() script throws "Error evaluating function substring" error.

   4.5. Improved the handling of special characters within column names and paths to eliminate failures. All non-alpha-numeric columns were tested.

5. Improvements in view generation

   5.1. Improved performance of generate views "R" (retrieve name from spreadsheet). Changed methodology to retrieve all column metadata on first invocation as opposed to doing a database lookup for each column. Large views that would take several minutes to create have dropped down to under 5 seconds. This timing is consistent no matter how many columns.

   5.2. CRUD - ConfigureStartingFolders: Modified CT_FOLDER (CRUD Target) to use the groupPath so that there is a unique path for each data source in the target CRUD folder.

6. Update to generateProject() and upgradeProject() to resolve defaultValues() version number.

V8.14 Modifications – August 25, 2014

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Works with Utilities_2014Q301.car or higher

   1.1. Fixed issue in Utilities getBasicResourceCursor_ActionAttributes() where it cut off the attribute value response to 255 characters. Increased size of attribute value.

2. Upgrade Modifications

   2.1. Added ability to upgrade a project folder that has no previous versions of Data Abstraction Best Practices. It detects the absence and creates "_scripts" with the latest version of the Data Abstraction Best Practices.

   2.2. Added ability to upgrade a project folder that only has "_scripts" with or without the subfolders. The folders may not contain any procedures however. It can only be an empty shell structure.

2.3. Added the ability to upgrade a project and copy any data abstraction template folders/resources that do not exist in the target project folder but don't overwrite any pre-existing resources such as "_Custom", "Configure" and "Documentation".

2.4. Fixed the upgrade for /Documentation folder to correctly modify the "keyVerifyText" when $PROJECT_PATH_SRC or $PROJECT_PATH_DST was being used in the upgrade vector "vector_masterUpgradeVector".

V8.13 Modifications – August 8, 2014

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Works with Utilities_2014Q2.car or higher

   1.1. Validated with Utilities_2014Q3.car

2. View Generation Modifications

   2.1. Added check for dependency views in target path. If either the actual view or a different view name with a dependency to the source exists then don't generate when 0 or 1.

   0=throw exception,

   1=skip generating if exists. Changed default to 1=overwrite instead of 2=overwrite if exists. Added an output message when "SKIPPED". The qualifying message will indicate the following scenarios:

   SKIPPED: Dependent resource exists. targetResourcePath=<path>

   SKIPPED: Target resource exists. targetResourcePath=<path>

   SKIPPED: Dependent resource exists. Target resource exists. targetResourcePath=<path>

   2=overwrite if exists

3. Modifications to "generateCast" variable for the various generateView... methods and is applicable when used when generateMode='G' or 'R'. Previously, a cast statements were only available when generateMode='G'. Now this feature is being expanded to include generateMode='R' and providing additional cast options. The recommended value would be to use generateCast=2 which would generate cast statements only around non-index columns. The CIS optimizer will be more efficient with query push-down on some databases that have indexes. The cast statements prevent the optimizer from choosing the correct push-down query.

   3.1. Added generateCast=2,3,4,5 allowing for "NO" CAST around index columns and adding a CAST display column for index columns. This will help with the CIS optimizer to be able to push down SQL to databases that support indexes. The optimizer will be able to make better choices.

   3.2. 0=Do not generate CAST statement. Pass through column as is. Default behavior.

   3.3. 1=Generate the CAST statement around the column

   3.4. 2=Generate the CAST statement around the non-index columns only (No CAST on index columns)

   3.5. 3=Generate the CAST statement around the non-index columns only and generate a "display" column for each index column. (No CAST on index columns)

3.6. 4=Generate the CAST statement around the non-index columns and non-primary key index columns only (No CAST on primary key index columns)

3.7. 5=Generate the CAST statement around the non-index columns and non-primary key index columns only and generate a "display" column for each primary key index column. (No CAST on primary key index columns)

4. CIS 6.1 has reached end of life and is no longer supported in this release.

V8.12 Modifications – May 12, 2014

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Rebranding PS Assets to AS Assets
   1.1. Changed folder from /shared/PSAssets to /shared/ASAssets
   1.2. Requires Utilities_2014Q2 which uses /shared/ASAssets
   1.3. Fixed issues with upgrading from different versions of the Best Practices starting with 6.0. Provided a new repaired version of the Best Practices which use /shared/ASAssets instead of /shared/PSAssets for the following Best Practices versions: 6.0, 6.6, 7.0, 7.1, 7.2, 7.3 and 8.0. Also provided a sample project for each of the versions named above.
2. Removed common_policy as it was a left-over artifact from testing.
3. Moved DataAbstractionSample to /ASAssets/DataAbstractionSample81.
4. Removed SQL Server and LDAP sources
   4.1. Leave, file, XML, MySQL, Excel and Oracle (those have been there in the past)
   4.2. The SQL Server source brings in the sqljdbc4.jar by default and overwrites existing capabilities files.
5. Changed from MySQL cache to the default file cache to make it easier from a startup configuration perspective.
   5.1. Provided instructions on how to create a relational database cache. Any DB may be used.
6. Added a call to re-introspect the Excel data source in the updateDatasourcePaths.
   6.1. Excel source in a UNIX environment must be re-introspected upon import.
7. Fixed lab answer - Lab00-Answer_v8_1_3.car.

V8.11 Modifications – April 1, 2014

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Requires an upgrade of Utilities_2014Q101.car
   1.1. Fixed getResourceLineageRecursive where a resource has a circular reference to itself.
   1.2. Utility resources affected:
      1.2.1. Utilities/repository/getDependentResourcesRecurseCursor
      1.2.2. Utilities/repository/getResourceLineageRecursive
      1.2.3. Utilities/repository/getResourceLineageDatasources

1.2.4. Utilities/repository/getResourceListUnpublished
1.2.5. Utilities/documentation/modules/getDocDataSourceLineage
1.2.6. Utilities/documentation/test/test_getDocDataSourceLineage
1.2.7. Utilities/pdtool/helpers/getDeployableResourceListByLineage
1.2.8. Utilities/pdtool/template_generatePDToolDeployableResourcePlan
1.2.9. Utilities/getUtilitiesVersion

2. Import .car file options:
   2.1. Patch .car (BestPractices_v8_1_2014_02_21_patch_20140327.car) – use this to upgrade an existing 8.1 version.
      2.1.1. To import using Composite Studio:
         2.1.1.1. Right-click on Desktop (<username>) or localhost:9400 (/)
         2.1.1.2. Select import
         2.1.1.3. Browse to the patch .car file
         2.1.1.4. Check the "overwrite" box
         2.1.1.5. Note: the other boxes may be left checked that were already checked.
         2.1.1.6. Click Import
   2.2. Full .car (BestPractices_v8_1_1_2014_03_27.car) – use this to replace the existing BestPractices_v8_1_2014_02_21.car file from the previous distribution. Import with "overwrite" checked.
3. Provide an upgrade from 8.0 to 8.1
4. Generate Datasource CSV throws an exception when there are foreign keys defined for a datasource table.
5. Ability to Generate Datasource CSV file with generateMode='G' and layerType='PM' and different caseRule settings
   5.1. Datasource is being skipped. Resolved the "WARNING" issue where it couldn't find the data source. This only happened when the folder being pointed to was of type "DATA_SOURCE". Resolved by getting resourceType of folder path. Example of issue:
      5.1.1. [10:35:29 26] generateDatasourceList : WARNING: The target container does not exist: path=/shared/PSAssets/BestPractices_v81/DataAbstractionSample/Physical/Metadata/File/testfile
   5.2. When generateMode='G' and the caseRule=C and 2 source columns Section and _Section are in the source and botha are reserved word so the need double quotes.
      5.2.1. When caseRule=C both Columns will resolve to "Section" resulting in duplicate columns.
      5.2.2. What happens with the above combo is that _Section is turned in to a duplicate column "Section" which is double quoted. There is code that detects duplicate column and adds a sequential number to it. It should have been Section1 but it was being generated as "Section"1 because Section is a reserved word and gets double quoted. That scenario is resolved.
      5.2.3. Additionally, the column that is generated 'Section1' may already exist so the code will make sure that a unique column is generated even when the generated column exists.

6. The user will be required to manually modify CIS_HOME/conf/customjars/RepoUtils.properties and add the word 'offset' to the list with a pipe separator. This will be resolved in the next relase of the PSAsset Utilities.

Patch Changes: (Release changes)

/shared/PSAssets/BestPractices_v81/_Help/_RELEASE_NOTES
/shared/PSAssets/BestPractices_v81/_ProjectMaintenance/defaultValues
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Constants/defaultValues
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Display/displayLineageTree
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Documentation/documentationTrigger_Application_Published
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Documentation/documentationTrigger_Application_Services
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Documentation/documentationTrigger_Application_Views
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Documentation/documentationTrigger_Business_Business
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Documentation/documentationTrigger_Business_Logical
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Documentation/documentationTrigger_DATABASE
/shared/PSAssets/BestPractices_v81/DataAbstraction_GENERIC_Template/_scripts/Generate/generateDatasourceList
/shared/PSAssets/BestPractices_v81/DataAbstractionSample/_scripts/Display/displayLineageTree
/shared/PSAssets/BestPractices_v81/DataAbstractionSample/Application/Services/CRUD (entire folder + subfolders)
/shared/PSAssets/BestPractices_v81/DataSource/common_model
/shared/PSAssets/BestPractices_v81/DataSource/search_common_model
/shared/PSAssets/BestPractices_v81/Procedures/checkDuplicateColumn
/shared/PSAssets/BestPractices_v81/Procedures/checkDuplicateColumnExists
/shared/PSAssets/BestPractices_v81/Procedures/crudProcedures/generateCRUD_TypeDefinitions
/shared/PSAssets/BestPractices_v81/Procedures/generateDatasourceList
/shared/PSAssets/BestPractices_v81/Procedures/generateDatasourceListCSV
/shared/PSAssets/BestPractices_v81/Procedures/generateName/applyWordRule
/shared/PSAssets/BestPractices_v81/Procedures/generateName/parseWord
/shared/PSAssets/BestPractices_v81/Procedures/generateViewsLoop
/shared/PSAssets/BestPractices_v81/Procedures/getName
/shared/PSAssets/BestPractices_v81/Procedures/projectMaintenance/generateConfigurationStartingFolders
/shared/PSAssets/BestPractices_v81/Procedures/projectMaintenance/vector_masterUpgradeVector
/shared/PSAssets/BestPractices_v81/Procedures/projectMaintenance/vector_upgradeVersionVector
/shared/PSAssets/BestPractices_v81/Procedures/sqlParser/parseSqlScriptComplex
/shared/PSAssets/BestPractices_v81/Procedures/sqlParser/parseSqlScriptSimple
/shared/PSAssets/BestPractices_v81/Procedures/sqlParser/utility/parseFromClause
/shared/PSAssets/BestPractices_v81/Procedures/sqlParser/utility/retrieveColumnList

V8.1 Modifications – February 21, 2014

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Support for Utilities_2014Q1.car.
2. Migrated procedures to Utilities including cachedProjectResources and updateImpactedResources.

V8.0 Modifications – December 05, 2013

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Support for Utilities_2013Q402.car.
2. Modified Common Model Spreadsheet by adding columns:
   - 2.1. Project Name – Provides the ability to handle unique entries across named projects, data sources, catalogs, schemas and resources
   - 2.2. Great Grand Parent Name – equates to the Composite data source name for an object.
   - 2.3. Grand Parent Name – equates to the catalog name for a physical data source object.
3. Added a new column model spreadsheet “Common_Model_v3_file4_sample_lab.xlsx” which contains the sample mappings for the sample project “DataAbstractionSample” and the lab projects “lab00-lab10”.
   - 3.1. The other three files are defaulted to blank out of the box.
4. Read and Cache from Excel spreadsheet
   - 4.1. Read directly from .xlsx instead of .csv file. Provide a flag to read from .csv if necessary.
   - 4.2. Cache contents of .xlsx or .csv in mysql for faster access.
5. Configure Starting Folders – modified generateConfigureStartingFolders to include data source name, catalog and schema in the group name. This will uniquely identify the group within the project folder. The name parts are separated by periods. For example dsname.catalog.schema.
   - 5.1. Existing projects are upgraded to use the new group ID format.
6. Generate Views Defaults
   - 6.1. “generateIndexes” - Changed default for the various generate…Views variable from 1 to 0.
     - 6.1.1. This includes generateFormattingViews, generateLogicalViews, generateBusinessViews, generateClientViews, generateClientPublished, and generateCastViews
     - 6.1.2. Except for the generatePhysicalViews which needs to generate indexes specifically for use by generateCRUDOperations.
   - 6.2. “excludeDatasourcePathList” – Added the ability to inject a data source paths to exclude when retrieving resources for generating.
   - 6.3. “generateViewsWrapper” – changed paramers: 0=output cursor, 1=output table information, 2=output table + column information. Default = 1.
   - 6.4. generateWithSourceColumn Default = 1. used when generateMode='R'
     - 6.4.1. 1=Generate the view with the source column (pass through)-logical status is UNCHANGED
     - 6.4.2. 0=Do NOT generate the view with the source column (no pass through)-logical status is DROPPED
7. Generate CRUD Procedures
   - 7.1. Default requirement requires layerType=”CR” when groupIds are being used to generate CRUD procedures. The configuration folders defined by the CR grouping must point to source views that do not have any derived columns. Create, Read, Update, Delete does not work when there are derived columns in the views.
   - 7.2. The best practice for creating CRUD procedures are as follows:
     - 7.2.1. Generate physical views using generatePhysical()
     - 7.2.2. Configure CR source to point to the physical views.
     - 7.2.3. Configure CR target to point to the /Application/Services/CRUD folder
     - 7.2.4. Generate CRUD using generateCRUDOperations()

V7.3 Modifications – August 30, 2013

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Utilities_2013Q301.car. Deprecated use of Utilities and Best Practices with CIS 6.0 and lower since it has reached end-of-life as of August 15, 2013.
2. Upgrade Project Changes
   2.1. Added the ability to automatically upgrade previous versions of the Best Practices starting at 1.0. All versions between 1.0 and 6.6 will be upgraded to a copy of the existing project but in the format of the current version of the best practices. The new copy will contain the latest best practices scripts. Additionally, a best effort is made to migrate the ConfigureStartingFolders() procedure which identifies the data source groups.
   2.2. Added a 7.2 to 7.3 script upgrade path.
   2.3. Moved BestPractices_vXX folder to /shared/PSAssets/BestPractices_v73.
3. General View Generation Changes
   3.1. generateDatasourceListCSV – Fixed issue with generating when there is a header.
   3.2. generateViews – Added printout of begin and end time and duration. Improved console window output with only blank lines after each view grouping instead of every line.
   3.3. generateViewsLoop – Fix generateViewsLoop performance issue by only invoking updateImpactedResources when a view is created and not on the entire folder for each resource.
4. Utility Changes
   4.1. Promoted several procedures to /shared/Utilities including: copyResourcesPrivileges, createAllFoldersPrivileges, createResourceCopy, and extractTextList.
   4.2. Changed utility /repository/copyResources commit block to an exception block to resolve a commit error.
   4.3. Changed utility /repository/rebind/getRebindableResources and /repository/getResourceListRecursive to contain BEGIN INDEPENDENT TRANSACTION blocks around the procedures getting resources. There were a lot of random runtime exceptions being thrown.
   4.4. Changed utility /repository/searchResources to allow resources with impactLevel=SYNTAX_ERROR to pass through as the source can and needs to be read for the best practices scripts. Added impactLevel to the output so filtering can still be done with other programs.
5. CRUD procedure generation changes
   5.1. For getFormatProcedure(), changed the defaults from formatString to return null if the column type is not found. This prevents the issue where an unsupported type is accidently configured with string. The procedures generateCRUD_RetrievePK() and generateCRUD_Update() will create a warning message in the log and console window. An example warning is shown below:

```
##########################################################################################

# WARNING: NO formatType PROCEDURE FOUND FOR columnType=OTHER  columnName=MY_OTHER_COL

##########################################################################################
```

   5.2. For getIsEmptyProcedure(), changed the defaults from isEmptyString to return null if the column type is not found. This prevents the issue where an unsupported type is accidently configured with string. The procedure generateCRUD_isEmpty() will create a warning message in the log and console window. An example warning is shown below:

```
##########################################################################################

# WARNING: NO isEmptyType PROCEDURE FOUND FOR columnType=OTHER  columnName=MY_OTHER_COL

##########################################################################################
```

6. Upgrading CRUD procedures prior to 7.2

   6.1. RetrievePK procedures – look for any impacted procedures. You will need to modify any RetrievePK procedures containing CLOB or BLOB column types.

      6.1.1. For CLOB column types, change the procedure formatString to formatClob.

      6.1.2. For BLOB column types, comment out or remove the line of code referring to the BLOB column. There is no formatting for BLOB column types.

   6.2. Update procedures – you may need to modify any Update procedures containing CLOB or BLOB column types. In general, most of the Best Practices versions have resolved any issues by providing a separate section of code for updating CLOB and BLOB.

      6.2.1. For both CLOB and BLOB, if there are any lines starting with "set updateClause = formatString(…" and are referring to CLOB or BLOB, comment out or remove the lines.

   6.3. isEmpty procedures – look for any impacted procedures. You will need to modify any isEmpty procedures containing CLOB or BLOB column types.

      6.3.1. For CLOB column types, change the procedure isEmptyString to isEmptyClob.

      6.3.2. For BLOB column types, change the procedure isEmptyString to isEmptyBlob.

V7.2 Modifications – June 28, 2013

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Utilities_2013Q204_60.car and Utilities_2013Q204_61+.car support
2. Parallel Installation – this gives the user the ability to install Best Practices 7.2 in parallel with previous instances of the Best Practices so that it allows the developers to migrate projects only when they want to. All dependencies on custom functions has been eliminated in this release in favor if explicit paths to allow this capability. Additionally the defaultValues (/shared/BestPractices_v72/Procedures/defaultValues) contains a constant which specifies the root path of the Best Practices so that any dynamically scripted references use this parameter.

   2.1. Installation folder: /shared/BestPractices_v72

3. Fixed a view generation issue when an oracle data source contains stored procedures which don't reveal the cursor metadata. This resulted in an error being thrown during view generation.
4. Fixed issues with CRUD procedure generation:

   4.1. For RetrievePK procedures, the formatBlob is not generated as it is not applicable.

   4.2. For isEmpty procedures, the isEmptyBlob and isEmptyClob are now being properly generated.

V7.1 Modifications – June 04, 2013

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Utilities_2013Q202_60.car and Utilities_2013Q202_61+.car support
2. Added "UpgradeProject" which will automatically upgrades a project starting at 7.0 to 7.1.
3. Added "logicalColumnType" to the output cursor for all of the generateViews method and its variation methods.
4. Added "generatePublishedResource" which can generate a resource LINK or folder of resource LINKS to a Composite Database or Schema.

5. Added parameters to each of the generate…View scripts for performing the following functionality:
   5.1. overwrite - allows user to decide whether they want to overwrite an existing view or not.
      5.1.1. 0="FAIL_IF_EXISTS"=do not overwrite the resource. If the resource exists, raise an exception.
      5.1.2. 1="SKIP_IF_EXISTS"=skip the resource if it exists and continue processing
      5.1.3. 2="OVERWRITE_IF_EXISTS"=do overwrite the resource if it exists.
   5.2. copyAnnotation - allows user to decide whether they want to copy annotations or not form both resource and columns.
      5.2.1. 0=false=do not copy the annotation from the target resource
      5.2.2. 1=true=do copy the annotation from the target resource
   5.3. copyPrivilegeMode – flag indicating the mode in which to copy privileges. Privileges are only copied from the parent when creating new resources including folders
      5.3.1. null (default) - do not set any privileges at all.
      5.3.2. 0 - set mode to "OVERWRITE_APPEND" - merges and does not update privileges for users or groups not mentioned.
      5.3.3. 1 - set the mode to "SET_EXACTLY" - makes privileges look exactly like those provided in the call.
   5.4. exactMatch – specified how the source resource will be matched against the target resource
      5.4.1. 0=fuzzy match - sourcePath + derivedFilterPath must simply be contained within resourcePath
      5.4.2. 1 (default)=exact match - sourcePath + derivedFilterPath must match exactly in resourcePath
6. Synchronized all generate…. procedures to have the same consistent interface.
7. Fixed generateDatasourceList to output logicalColumnType when generateMode='G'
8. Fixed generateDatasourceList and generateDatasourceListCSV to output correct format when a physical column name is repeated via derived columns. For example, if the Formatting view contained the column "id" and another column "id || '_text' id2, the output would incorrectly list the physical name twice. The result would be the spreadsheet gets generated incorrectly and then the generateFormattingViews cannot interpret the difference between a physical to logical mapping and a new derived column.
9. Fixed generation scripts to not generate when column type='OTHER'. This occurs when Composite introspects data types like Oracle SDO spatial that do not have mappings to composite.
10. Removed the use of CHR(13) (carriage return) when creating/updating view text and <CR> for Logical Definitions in the spreadsheet as they result in
 or
 showing up in views and annotations in 6.0 and above.
11. Modified the signature of rebindAllResources to include rebindFromFolder.

V7.0 Modifications – April 19, 2013

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Utilities_2013Q104_60.car and Utilities_2013Q104_61+.car support
2. Added the following automation procedures:
   2.1. generateProject() to automate the creation of a project.
   2.2. renameProject() to rename a project and dynamically modify any views or procedures containing the project path in the text.
   2.3. moveProject() to move a project from one folder to another folder structure and dynamically modify any views or procedures containing the project path in the text.

2.4. generateConfigureStartingFolders() to auto-generate the ConfigureStartingFolders from data sources and transformations.Provide the ability to generate the Common_Model_v2_file[1-3].csv spreadsheets from either the Formatting Layer or Physical Metadata tables.

3. Modified ConfigureStartingFolders to remove "sourceName" from the result cursor. Simplified the configuration by allowing the source folder to point to any FOLDER/CONTAINER or DATA_SOURCE at any level. Therefore, the sourceName qualifier is no longer needed. Previously, the generation scripts threw an error if the source folder pointed to the actual data source instead of a FOLDER/CONTAINER. No restrictions now.

4. Modified the layers to 3 major layers. This was a major shift in the Best Practices.

4.1. Removed the _Layer from the end to shorten the name

4.2. Removed the L1_, L2_, L3_, L4_ in front of the major layers to shorten the name

4.3. The following has occurred in order to position the Best Practices layers more in alignment with the Forrester report.

4.3.1. Removed the Physical Layer sub-layer “Physical” as its functionality was limited in scope and the “/Physical/Formatting” sub-layer has evolved to take over its functionality.

4.3.2. Removed the L2_Formatting_Layer by moving the sub-layer “Formatting” down to the Physical Layer. The formatting views provide a transition from physical to logical and are one-to-one mapping of physical entities. This shift is in alignment with what Forrester talks about.

4.3.3. Renamed L4_Mapping_Layer to simply “Application” whereby Layer is implied

4.3.4. Renamed L3_Business_Layer to simply “Business” whereby Layer is implied

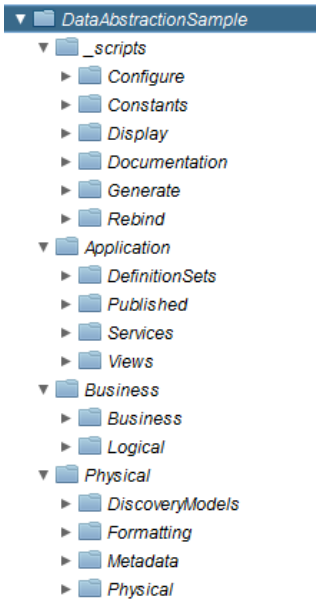
4.3.5. Renamed L1_Physical_Layer to simply “Physical” whereby Layer is implied

4.3.6. All three names above are in alignment with what Forrester discusses in terms of the three major layers.

4.4. Removed Federated_Views as they are really simply Logical Views. All base views with no “where-clauses” whether they are regular joins, federated joins or federated unions should go in the “Business/Logical” sub-layer

4.5. Added Application/Published contract to the “Application” layer as it has been determined through practice that some client tools will break unless an explicit contract of type-casted views are provided to the BI client.

4.6. Moved all script, documentation and constants folders to the “_scripts”. Renamed scripts folder to “_scripts” so that it would occur above the three layers which results in “Application”, “Business” and “Physical” naturally occurring in top-down, alphabetic occurrence which aligns with the Best Practices visuals and Forrester’s representation. Visually, the folders are arranged as follows in CIS:



5. Round-trip Synchronization – Added generateDatasourceListCSV which generates the exact format as the spreadsheet. This allows for executing a round-trip between the Formatting Layer and the Spreadsheet. This resolves two things.

    5.1. (1) A developer may update the Formatting Layer and not the spreadsheet. Easy to synchronize the two. The one manual step is the developer needs to copy columns A-I from the generated CSV back into one of the Common_Model_v2_file[1-3].xlsx spreadsheets. Exporting directly to .xlsx format is not in scope for the Best Practices scripts.

    5.2. (2) Upgrading the Best Practices spreadsheet. Allows user to updgrade and re-generate the new spreadsheet format in order to re-synchronize with any changes to the spreadsheet format when the Best Practices version changes.

        5.2.1. SCOPE: See scope in #3 above.Added generateProjectCopy and generateProject to automate the creation of a project.

6. Removed generateFormattingViewsWrapper() and generatePhysicalViewsWrapper() and added a variable “generateViewsWrapper” to each generation script method.

Code Changes

This section contains the code changes.

1. Synchronized this release with Utilities_2013Q104_60.car and Utilities_2013Q104_61+.car
2. Complete overhaul of Excel spreadsheet. Modified columns and formulas in the spreadsheets Common_Model_v2_file[1-3].xlsx. Spreadsheet Format:

    COL A: Data Source - The CIS Data Source Name

    COL B: Parent Name - The parent name of the Container (a.k.a grandparent of the physical name)

    COL C: Container Name - The contain (a.k.a. parent) of the physical name)

    COL D: Physical Name - They physical resource name which may be a TABLE or COLUMN

    COL E: Physical Type - The physical or native type of the physical resource

    COL F: Logical Name - The logical name of the VIEW/TABLE or COLUMN

    COL G: Logical Type - The logical type

    COL H: Logical Transformation - The Composite transformation (excluding outer cast statements)

    COL I: Logical Definition - The logical description or annotation of the TABLE/VIEW or COLUMN

3. Overhauled **generateDatasourceList** to generate a similar format of the spreadsheet.

It actually generates the same format except that it adds a "Logical Path" column that is not in the spreadsheet as the last column in the cursor.

SCOPE:

3.1. Provides the ability to generate Formatting Views in order to export to the Common_Model_v2_file[1-3].xlsx spreadsheet.

3.2. Provides the ability to generate Physical Views or Physical Metadata Tables to the Common_Model_v2_file[1-3].xlsx spreadsheet in order to establish a baseline spreadsheet which can then be modified to add logical names, types, transformations and definitions.

3.3. It will **NOT** generate Business Layer or Application Layer views with complex joins. It is only targeted at generating views containing simple one-table mappings such as those found in the Formatting Layer and below.

4. Added **generateDatasourceListCSV** which generates the exact format as the spreadsheet. This allows for executing a round-trip between the Formatting Layer and the Spreadsheet. This resolves two things.

   (1) A developer may update the Formatting Layer and not the spreadsheet. Easy to synchronize the two. The one manual step is the developer needs to copy columns A-I from the generated CSV back into one of the Common_Model_v2_file[1-3].xlsx spreadsheets. Exporting directly to .XLSX format is not in scope for the Best Practices scripts.

   (2) Upgrading the Best Practices spreadsheet. Allows user to upgrade and re-generate the new spreadsheet format in order to re-synchronize with any changes to the spreadsheet format when the Best Practices version changes.

   SCOPE: See scope in #3 above.

5. Modified **generateViewsLoop** to have the flexibility to use CONTAINER or DATA_SOURCE for the resource type.

6. Modified gernateCRUD_Operations and generateCRUD_TypeDefinitions to have the flexibility to use CONTAINER or DATA_SOURCE for the resource type. Set the default to 0 for “ConfigParamsVector[1].useAliaseRule” which means do not apply aliases.

V6.6 Modifications – November 15, 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. 2012Q401 Utilities support

Code Changes

This section contains the code changes.

1. Synchronized this release with Utilities_2012Q401 and Utilities_2012Q401_61+
   Note: This release requires Utilities_2012Q401.car and Utilities_2012Q401_61+.car

2. Added examples of generating documentation as part of the Best Practices

   2.1. Utilizes /shared/Utilities/documentation/documentationTrigger and /constants

   2.2. When a project is created from the "DataAbstraction_GENERIC_Template", the user will also have the ability to schedule the documentation generation.

V6.5 Modifications – November 05, 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. 2012Q4 Utilities support

Code Changes

This section contains the code changes.

1. Synchronized this release with Utilities_2012Q4 and Utilities_2012Q4_61+
   Note: This release requires Utilities_2012Q4.car and Utilities_2012Q4_61+.car

Added resourceType parameter to the invocation of getResourceListRecursive() in several procedures.
1.1. Procedure affected: generateViewsLoop(), generateCRUD_Operations(), generateDatasourceList(), rebindAllResources(), rebindGenerationScripts(), and displayResourceTree()

V6.4 Modifications – October 1, 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Enhancement – added “generateCastViews”

Code Changes

This section contains the code changes.

1. Added a new feature for generating views with CAST statements without relying on the ConfigureStartingFolders. This provides a more adhoc and flexible capability.
   1.1. “generateCastViews” – interface method
   1.2. “generateCastViews” – core procedure with all the logic
   1.3. “generateViewsLoop – split out the original loop processing from generateViews
   1.4. “generateViews” – split out the setup code from the loop processing code
   1.5. “isDerivedPathMatch” – provided ability to match exactly or do fuzzy match
   1.6. Modified other functions that invoke “isDerivedPathMatch” due to method signature change.

V6.3 Modifications – September 29, 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Bug fixes.

Code Changes

This section contains the code changes.

1. Fixed an issue where column names contain spaces or underscores either in the beginning or end of the column name.
   1.1. Code changes for generating physical views (removed TRIM statements):
      1.1.1. /shared/BestPractices/Procedures/generateName/parseWord
         1.1.1.1. Added use of RegexSplit function with -1
      1.1.2. /shared/BestPractices/Procedures/generateName/getNameGenerated

2. Fixed an issue where column names contain spaces at the beginning or end of the column name when retrieving formatting view columns.
   2.1. Code changes for generating formatting views (removed TRIM statements):
      2.1.1. /shared/BestPractices/Procedures/retrieveName/getNameRetrieved

3. Data source /File modifications
   3.1. Added a column to testfile.txt to test the new capabilities.
   3.2. Added Common_Model_file[1,2,3].xls to the File folder in order to test columns containing XML characters.

4. Code changes for generating views using the “derivedFilterPath” so that it is not as rigid. Allow for any part of the path to be present instead of an exact match. The derivedFilterPath would contain any text that represents the path of the “source” resource starting after what is provided by the based folder path in ConfigureStartingFolders. The “derivedFilterPath” is applicable for all the generation scripts

   4.1. For example, if the ConfigureStartingFolders contains

set groupId = 'DELIMITED';
SET PM_FOLDER=physicalMetadataPath||'/File';

Therefore the “sourceFolderPath” would resolve to
/shared/BestPractices/DataAbstractionSample/Physical/Physical_Metadata/File

The two data sources under /File include Common_Model and testfile. Additionally, Common_Model contains three files which are Common_Model_file1.xls, Common_Model_file2.xls and Common_Model_file3.xls.

Therefore if the user sets “derivedFilterPath” = testfile, then only testfile is generated.

If the user sets “derivedFilterPath” = Common_Model then all three Common_Model_file[1,2,3].xls are generated.

If the user sets “derivedFilterPath” = Common_Model/Common_Model_file2 then only Common_Model_file2.xls is generated.

5. Code changes for escaping column names containing XML characters:
   5.1. /shared/BestPractices/Procedures/getReservedWord
   5.2. Escape any XML characters in the column names like
      5.2.1. apostrophe (') '
      5.2.2. ampersand (&) &
      5.2.3. less than (<) <

Also requires a fix to the Utilities: /shared/Utilities/repository/updateResourcesSqlTable line 121
Before: <resource:sqlText>'||scripttext||'</resource:sqlText>
After: <resource:sqlText>'||CAST(XMLTEXT(scripttext) AS LONGVARCHAR)||'</resource:sqlText>

V6.2 Modifications – August 6, 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Bug fixes.

Code Changes

This section contains the code changes.

1. Fixed an issue where double quotes were not being put around physical column names containing spaces.
2. Fixed an issue where generated views with CAST statements that included data types with commas "DECIMAL(12,2)" forced the portion of the data type following the comma onto a separate line.
3. Fixed an issue where reserved words are applied to generation of CRUD methods.

V6.1 Modifications – July 30, 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Ability to control generating indexes.

Code Changes

This section contains the code changes.

1. Changes were made to the ConfigureParams vector

1.1. Added generateIndexes variable

V6.0 Modifications – May 19, 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. Ability to generate the CAST statement from underlying data types when mode='G' (Generate) and not 'R' (Retrieve).
2. Ability to generate the CAST statement for generateDatasourceList
3. Ability to pass in a comma separated list of groupId filters for the ConfigurStartingFolder when generating any layer. This allows the user to be more specific about what they want to generate views for.
4. Ability to generate physical views from XSLT procedures in order to support XML data sources.
5. Ability to search a resource tree "searchResourceTree" for the passed in resource path was documented.
6. A wrapper for generatePhysicalViews so that developers can execute without hitting the Studio Fetch Row Size limit.
7. A wrapper for generateFormattingViews so that developers can execute without hitting the Studio Fetch Row Size limit.
8. A display procedure that returns the Best Practices current version.
9. Ability to set a filter path to only generate views based on a comma separate list of derived filter paths.
10. A new procedure to generateClientPublished views with casting defaulted.

Code Changes

This section contains the code changes.

1. Changes were made to the ConfigureParams vector
   1.1. Added generateCast variable
2. Changes were made to the output cursor for ConfigureStartingFolders
   2.1. Added groupId to each insert statement
3. Changes were made to each of the procedures in /generationScripts/generate
   3.1. Added groupId input parameter
4. Wide spread changes were made to the procedures in /procedures and are too numerous to list here.

V5.1 Modifications – Jan 2012

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. **Multiple Schemas** – Ability to generate views where the data source contains multiple schemas.

Code Changes

This section contains the code changes.

1. generateViews – Moved the code to "SET firstTimeInLoop=FALSE" inside the loops for childRes2 and childRes3. This fix allows the loops to accurately print out the various tables found at these levels. Previously, it was only printing out the last table/view in the list of tables from the data source.

   1.1. Path: /shared/BestPractices/Procedures/generateViews

1.2. Explanation: The code was only getting the last Table in the list when the top level folder contains multiple data sources which in turn contains multiple schema container folders which contain tables. An example is shown below:

Physical_Metadata
- Datasource1
  - SCHEMA1
    - Table1
    - Table2
    - TableN
- Datasource2
  - SCHEMA1
    - Table1
    - Table2
    - TableN
  - SCHEMA2
    - Table1
    - Table2
    - TableN
  - SCHEMA3
    - Table1
    - Table2
    - TableN
- Datasource3
  - SCHEMA1
    - Table1
    - Table2
    - TableN
  - SCHEMA2
    - Table1
    - Table2
    - TableN

V5.0 Modifications – Nov 2011

This section contains a list of features and code modifications.

Features

This section contains the new feature descriptions.

1. **Bug Fixes** – general bug fixes .

Code Changes

This section contains the code changes.

1. DataAbstractionSample / DataAbstraction_GENERIC_Template – removed the second level layer numbering schema from the folders to shorten the path. For example, L1_1_Client_Services was shortened to Client_Services. This was done for all the second level folders in both the sample and the template.
   a. Path: /shared/BestPractices/DataAbstraction_GENERIC_Template/constants/defaultValues
   b. Path: /shared/BestPractices/DataAbstraction_GENERIC_Template/constants/defaultValues
2. defaultValues – removed second level layer numbering schema for variables [clientServicesPath, clientViewsPath, genUniqueIDPath]. Added base-path variables for each layer to be used in “ConfigureStartingFolders”. This makes the entire project more portable and easier to change base names.
   a. Path: /shared/BestPractices/DataAbstraction_GENERIC_Template/constants/defaultValues
   b. Path: /shared/BestPractices/DataAbstraction_GENERIC_Template/constants/defaultValues

3. ConfigureStartingFolders – made use of variables for base path names. Organized examples in a better way. Provided better documentation.
    a. Path: /shared/BestPractices/DataAbstractionSample/generationScripts/Configure/ConfigureStartingFolders
    b. Path: /shared/BestPractices/DataAbstraction_GENERIC_Template/generationScripts/Configure/ConfigureStartingFolder
4. LogicalInterfaceNameUnion – fixed reference to the 3rd file from Common_Model_file2.csv to Common_Model_file3.csv.
    a. Path: /shared/BestPractices/Procedures/retrieveName/LogicalInterfaceNameUnion
5. CommonModelSources – Added additional fields to the 3 data source files
    a. Path: /shared/BestPractices/Procedures/retrieveName/CommonModelSources
6. generateViewsCommon – Put brackets around output to detect spaces more easily
    a. changed transformResourceName to compositeTransformation
    b. Path: /shared/BestPractices/Procedures/generateViewsCommon
7. generateViews – added an insert statement for output prior to the column output for each table/view that is encountered.
    a. Path: /shared/BestPractices/Procedures/generateViews
8. Common_Model_file1,2,3.xls – Modified format of excel and csv files
    a. Path: C:/CiscoSystems/BestPractices/BestPractices
    b. Inserted “Data Source” column before column A
    c. Inserted “Parent Container” column before Table Name
    d. Inserted “Composite Transformation” column after Table Column Name
9. CommonModelSource – Re-instrospected the data source to pick up new columns
    a. Path: /shared/BestPractices/Procedures/retrieveName/CommonModelSources
10. LogicalInterfaceNameRetrieval – Column names changed in csv file so re-adjusted them from resourceName to physicalName and transformationResourceName to compositeTransformation
    a. Path: /shared/BestPractices/Procedures/retrieveName/LogicalInterfaceNameRetrieval
11. getNameRetrieved – adjusted column names from LogicalInterfaceNameRetrieval
    a. Added a debug statement at the beginning and end of the procedure to see the input parameters
    b. Path: /shared/BestPractices/Procedures/retrieveName/getNameRetrieved
12. retrieveNewColumnList – adjusted column names from LogicalInterfaceNameRetrieval
    a. Path: /shared/BestPractices/Procedures/retrieveName/retrieveNewColumnList

5 Appendix C – Version Differences

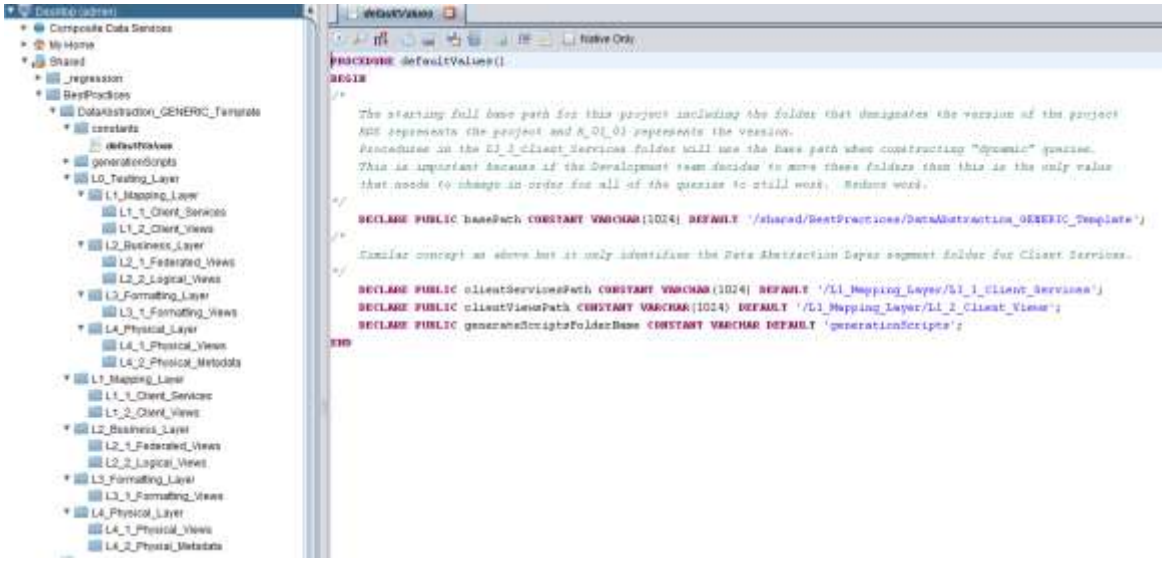
Best Practices Version Differences

The following section provides a description of the differences between the various Best Practices versions.

Best Practices 1.0

The following are highlights of this release

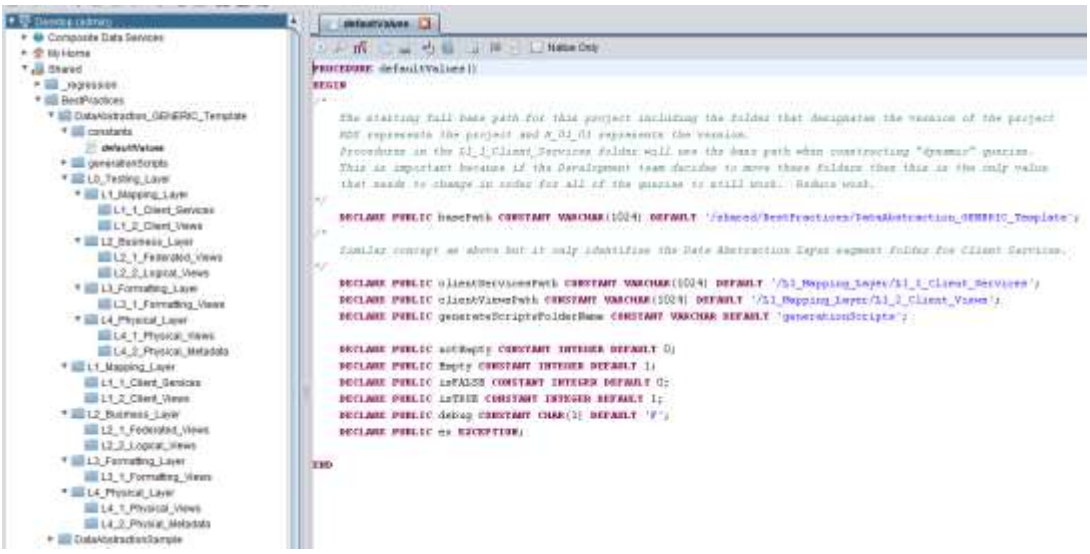
- Scripts are found in /generationScripts.
- Constants are found in /constants/defaultValues
- Folder spelled wrong /L4_Physical_Layer/L4_2_**Physial**_Metadata
- Graphic of directory structure and constants/defaultValues



Best Practices 2.0 (same structure as 1.0)

The following are highlights of this release

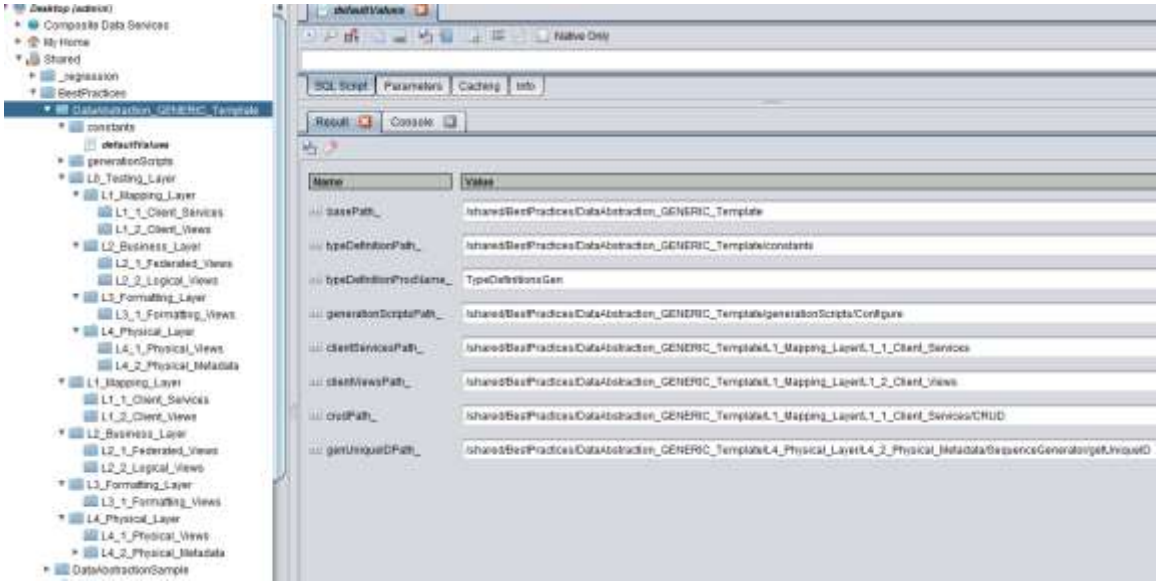
- Scripts are found in /generationScripts.
- Constants are found in /constants/defaultValues
    - Added public variables
- Folder spelled wrong /L4_Physical_Layer/L4_2_**Physial**_Metadata
- Graphic of directory structure and constants/defaultValues



Best Practices 3.0 (same structure as 1.0)

The following are highlights of this release

- Scripts are found in /generationScripts.
- Constants are found in /constants/defaultValues
    - Added OUT variables for the path type variables
- Folder spelled correctly /L4_Physical_Layer/L4_2_**Physical**_Metadata
- Graphic of directory structure and constants/defaultValues

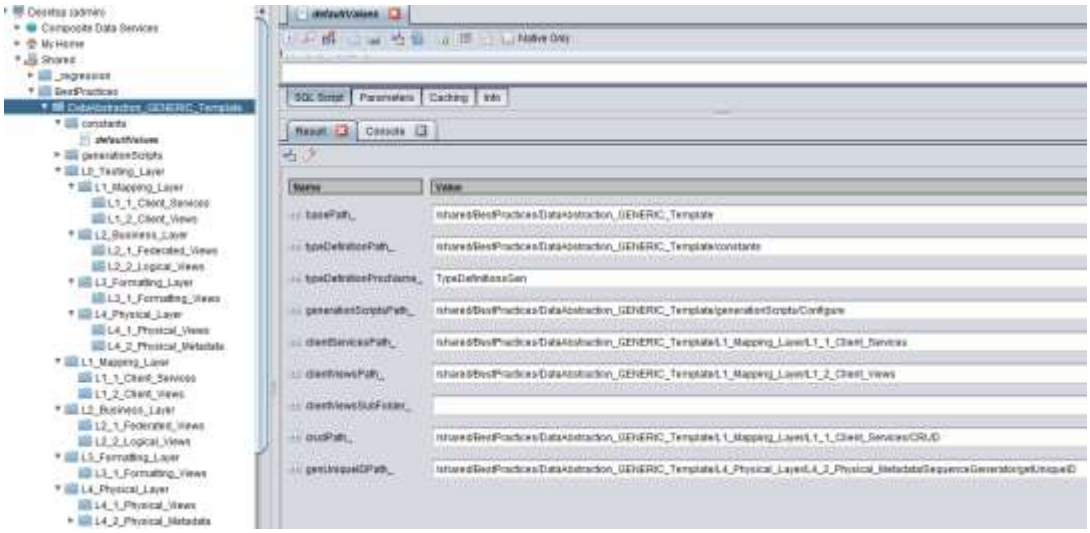


Best Practices 4.0 (same structure as 1.0)

The following are highlights of this release

- Scripts are found in /generationScripts.

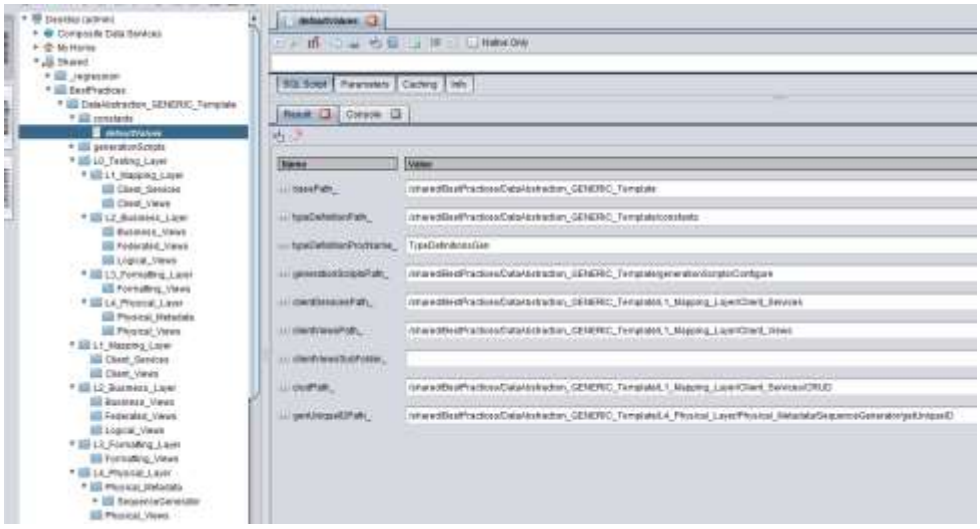
- Constants are found in /constants/defaultValues
    - Added OUT variable for clientViewsSubFolder_
- Folder spelled correctly /L4_Physical_Layer/L4_2_**Physical**_Metadata
- Graphic of directory structure and constants/defaultValues



Best Practices 5.0 (new structure for 5.1 through 5.2)

The following are highlights of this release

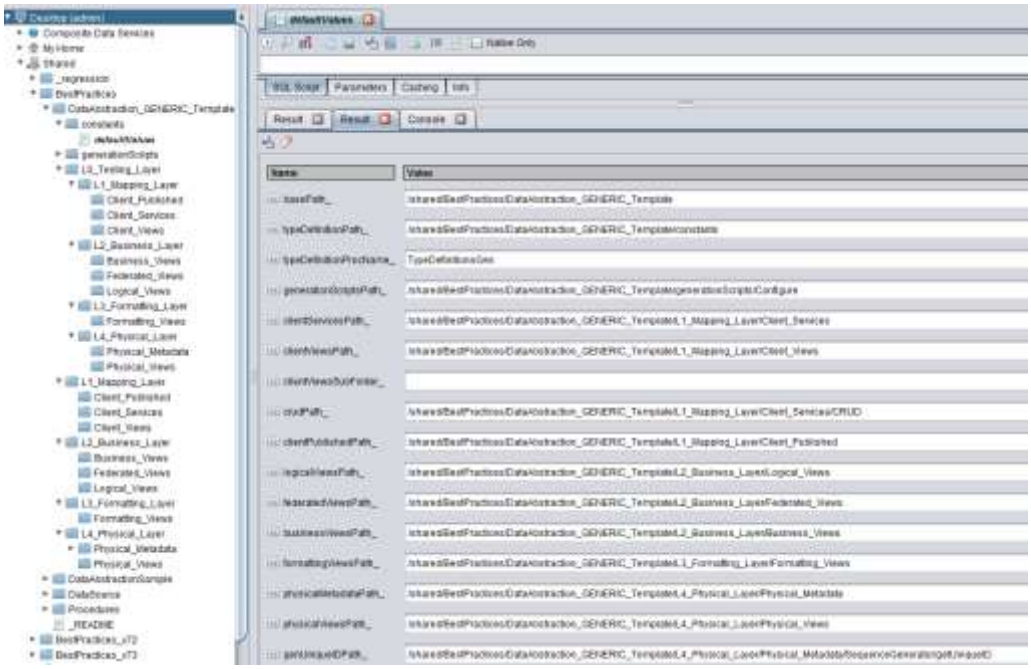
- Scripts are found in /generationScripts.
- Constants are found in /constants/defaultValues
- Structure changes
    - Removed the level prefix from the 2nd level views (e.g. /L1_Mapping_Layer/Client_Views)
    - Added Business_Views to /L2_Business_Layer/Business_Views
- Graphic of directory structure and constants/defaultValues



Best Practices 6.0 (new structure for 6.0 through 6.5)

The following are highlights of this release

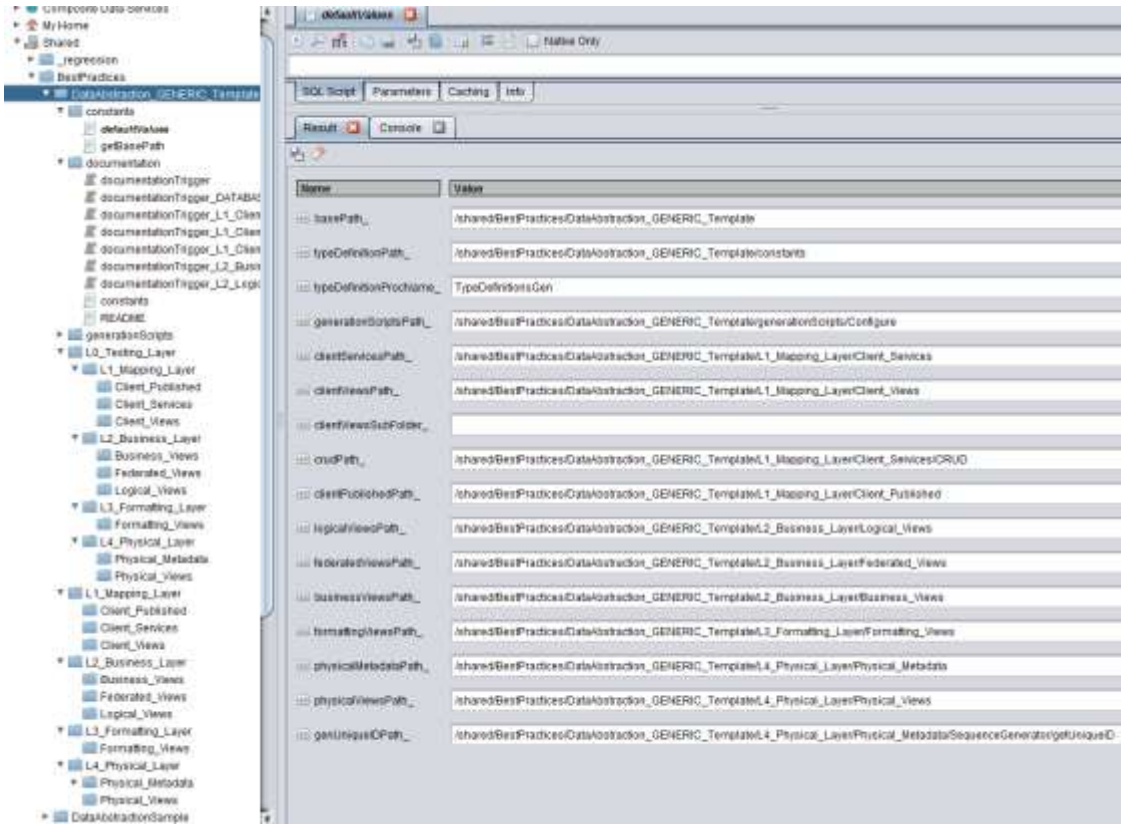
- Scripts are found in /generationScripts.
- Constants are found in /constants/defaultValues
    - Added 7 more OUT variables for the path types
- Structure changes
    - Added Client_Published to /L2_Mapping_Layer/Client_Published
- Graphic of directory structure and constants/defaultValues



Best Practices 6.6 (same as 6.0, added /documentation)

The following are highlights of this release

- Scripts are found in /generationScripts.
- Constants are found in /constants/defaultValues (same as 6.0)
- Structure changes
    - Best Practices folders are the same as 6.0
    - Added /documentation folder
- Graphic of directory structure and constants/defaultValues



Best Practices 7.0 (complete structure and script change)

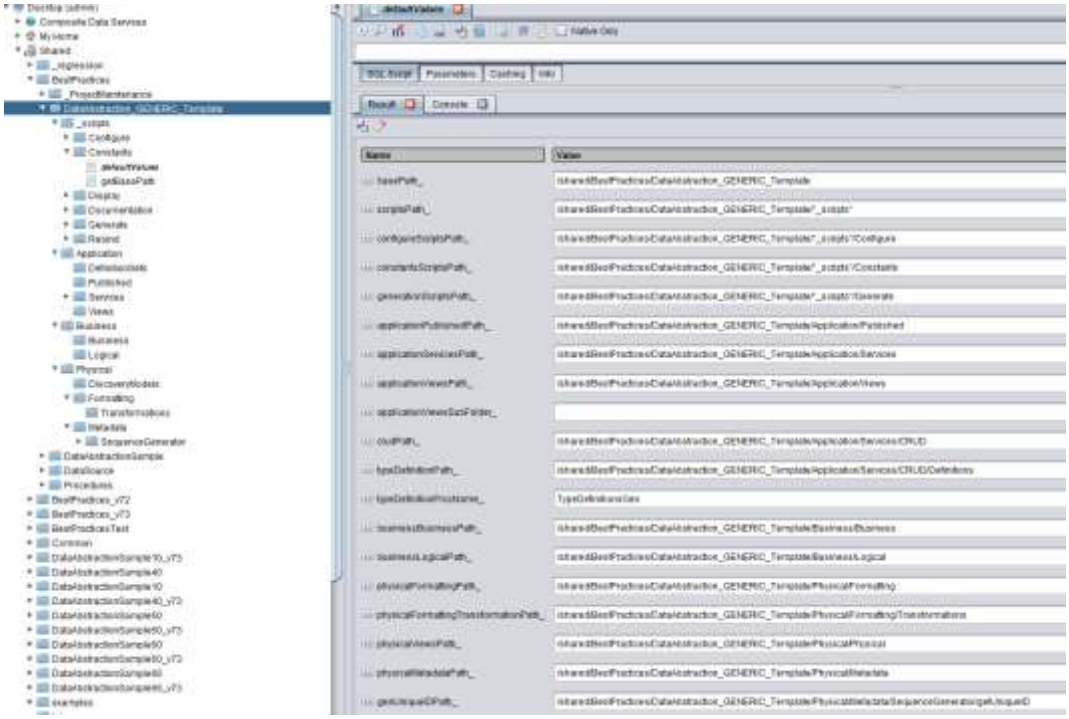
The following are highlights of this release

- Scripts are found in /_scripts.
- Constants are found in /_scripts/Constants/defaultValues
    - Added scriptsPath_, configureScriptsPath_, constantsScriptsPath_
    - Changed names of variables to reflect folder name changes.
- Structure changes (all new)
    - Changed to 3 major levels instead of 4
    - Changed folder names for major layers and sub-layers and added sub-layers
        - L1_Mapping_Layer → Application
        - L2_Business_Layer → Business
        - L4_Physical_Layer → Physical
    - Moved Formatting views to the physical layer
        - L3_Formatting_Layer → /Physical/Formatting
    - Deprecated Federated and Physical views
        - Will still migrate the folders when upgrading projects for 6.6 and lower
    - Changed script location and names
        - /generationScripts → /_scripts

- /generationScripts/Configure → /_scripts/Configure
- /generationScripts/Display → /_scripts/Display
- /generationScripts/Generate → /_scripts/Generate
- /generationScripts/Rebind → /_scripts/Rebind

  - /constants → /_scripts/Constants
  - /documentation → /_scripts/Documentation

- Graphic of directory structure and constants/defaultValues

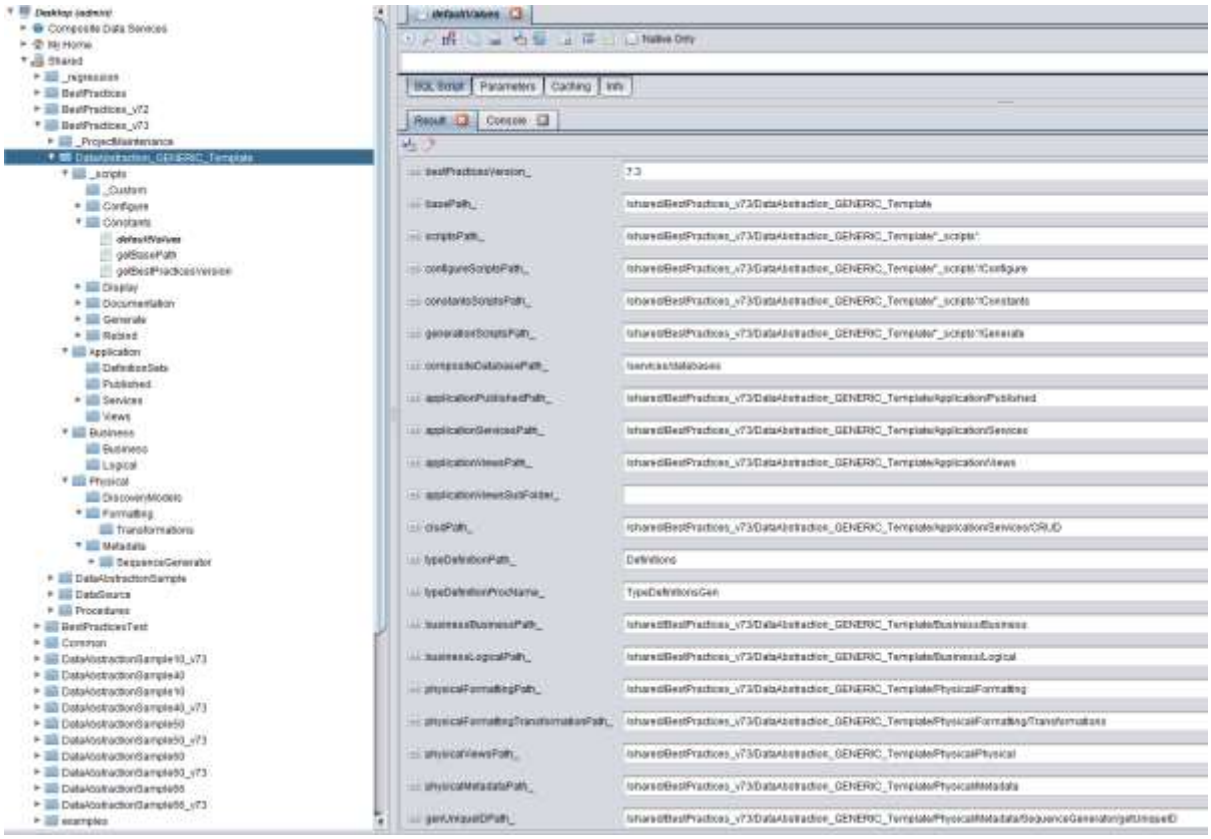


Best Practices 7.1 thru 7.3 (same as 7.0, minor tweaks to defaultValues)

The following are highlights of this release

- Best Practices scripts 7.2 and forward introduced versioning of the Best Practices scripts so that they can be installed in parallel with previous releases. This was done to lesson the impact of upgrading. However, there is still a reliance on specific versions of the Utilties in order to execute the best practices scripts. An example of the best practices folders is shown below:
  - /shared/BestPractices – B.P. 7.1 and lower
  - /shared/BestPractices_v72 – Best Practices v7.2 (requires Utilities_2013Q204_61+.car)
  - /shared/PSAssets/BestPractices_v73 – Best Practices v7.3 (requires Utilities_2013Q301.car)
- Scripts are found in /_scripts.
- Constants are found in /_scripts/Constants/defaultValues
  - Added OUT variables for bestPracticesVersion_ and compositeDatabasePath_
  - For 7.3, added federatedViewsPath as an internal variable that is not published as an OUT variable for the purposes of upgrading prior projects.

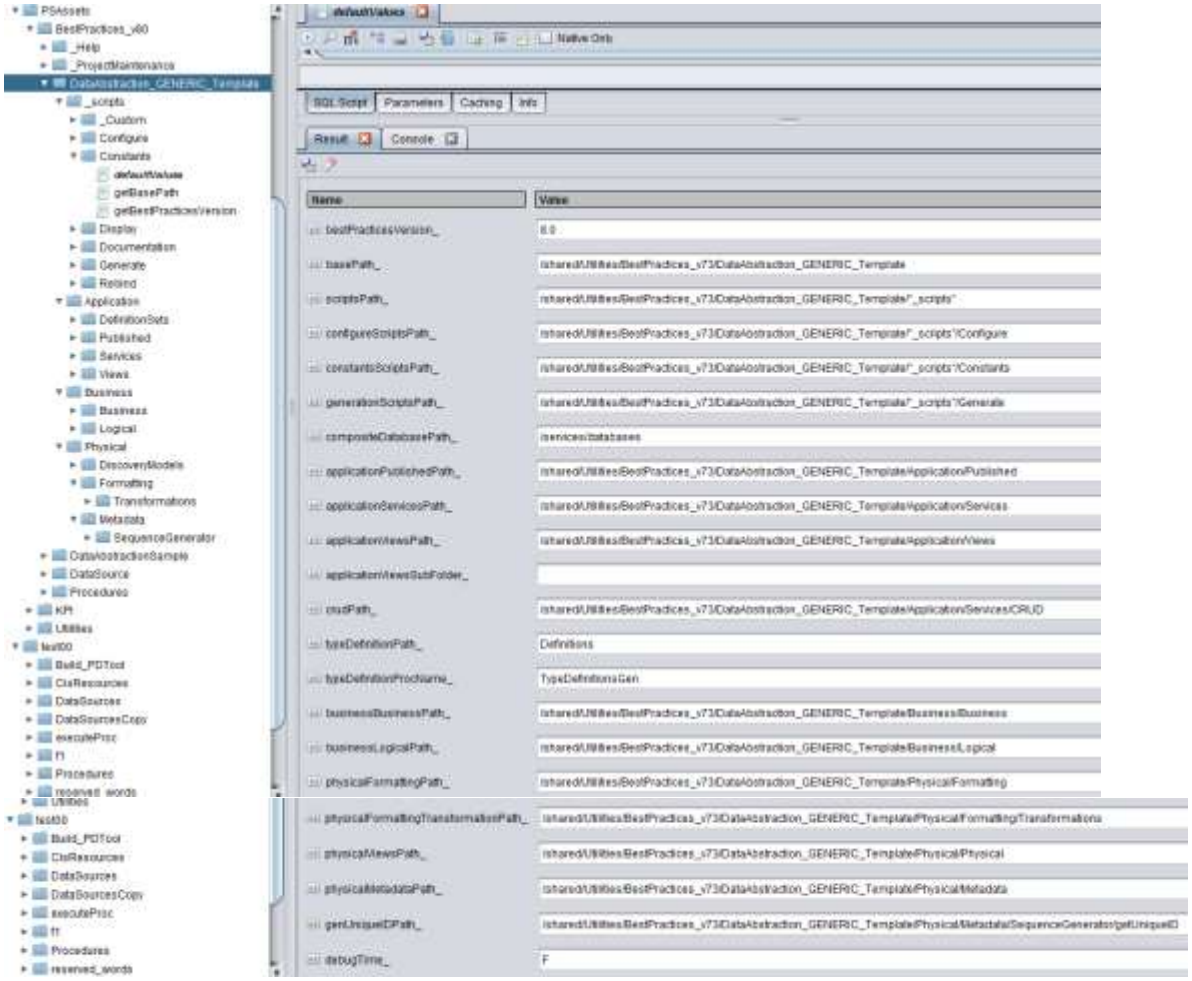
    - Added a procedure “getBestPracticesVersion” to be able to determine exactly what the version the scripts are in.
- Structure changes (same as 7.0)
- Graphic of directory structure and constants/defaultValues



Best Practices 8.0 (same as 7.0, minor tweaks to defaultValues)

The following are highlights of this release

- Scripts are found in /_scripts.
- Constants are found in /_scripts/Constants/defaultValues
    - Added OUT variables for debugTime_
- Structure (same as 7.0)
- Graphic of directory structure and constants/defaultValues



Best Practices Version Mapping

The following section provides a description of how the various versions are mapped to 7.x/8.x baseline. These mappings are performed automatically when using the “upgradeProject” procedure.

Best Practices 1.0 thru 4.0 mapped to 7.x/8.x

The following section provides a description of how 1.0 thru 4.0 is mapped to 7.x/8.x.

```
-- 1. Copy Physial Metadata (Spelled wrong in 1.0 and 2.0)

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/L4_2_Physial_Metadata','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata','CONTAINER')], null)

-- 2. Copy Physical Metadata (Spelled correctly in 3.0 and 4.0)

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/L4_2_Physical_Metadata','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata','CONTAINER')], null)

-- 3. Copy Physical Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/L4_1_Physical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Physical','CONTAINER')], null)

-- 4. Copy Formatting Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L3_Formatting_Layer/L3_1_Formatting_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Formatting','CONTAINER')], null)

-- 5. Copy Logical Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/L2_2_Logical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Logical','CONTAINER')], null)

-- 6. Copy Federated Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/L2_1_Federated_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Federated','CONTAINER')], null)

-- 7. Copy Client Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/L1_2_Client_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Views','CONTAINER')], null)

-- 8. Copy Client Services

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/L1_1_Client_Services','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services','CONTAINER')],          null)
```

```
-- 9. Copy constants to Definitions used by CRUD

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null, VECTOR[('$PROJECT_PATH_SRC/constants','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions','CONTAINER')],              null)

-- 10. Copy Test Physical Metadata (Spelled correctly in 1.0 through 4.0)

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/L4_2_Physical_Metadata','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Metadata','CONTAINER')], null)

-- 11. Copy Test Physical Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/L4_1_Physical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Physical','CONTAINER')], null)

-- 12. Copy Test Formatting Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L3_Formatting_Layer/L3_1_Formatting_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Formatting','CONTAINER')], null)

-- 13. Copy Test Logical Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/L2_2_Logical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Logical','CONTAINER')], null)

-- 14. Copy Test Federated Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/L2_1_Federated_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Federated','CONTAINER')], null)

-- 15. Copy Test Client Views

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/L1_2_Client_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Views','CONTAINER')], null)

-- 16. Copy Test Client Services

(1.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/L1_1_Client_Services','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Services','CONTAINER')],   null)

-- 17. Rebind Physial Metadata (Spelled wrong in 1.0 and 2.0)

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/L4_2_Physial_Metadata'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata'||'/','CONTAINER')], null)

-- 18. Rebind Physical Metadata (Spelled correctly in 3.0 and 4.0)

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/L4_2_Physical_Metadata'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata'||'/','CONTAINER')], null)
```

```
-- 19. Rebind Physical Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/L4_1_Physical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Physical'||'/','CONTAINER')], null)

-- 20. Rebind Formatting Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L3_Formatting_Layer/L3_1_Formatting_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Formatting'||'/','CONTAINER')], null)

-- 21. Rebind Logical Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/L2_2_Logical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Logical'||'/','CONTAINER')], null)

-- 22. Rebind Federated Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/L2_1_Federated_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Federated'||'/','CONTAINER')], null)

-- 23. Rebind Client Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/L1_2_Client_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Views'||'/','CONTAINER')], null)

-- 24. Rebind Client Services

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/L1_1_Client_Services'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services'||'/','CONTAINER')], null)

-- 25. Rebind Test Physical Metadata (Spelled correctly in 1.0 through 4.0)

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/L4_2_Physical_Metadata'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Metadata'||'/','CONTAINER')], null)

-- 26. Rebind Test Physical Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/L4_1_Physical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Physical'||'/','CONTAINER')], null)

-- 27. Rebind Test Formatting Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L3_Formatting_Layer/L3_1_Formatting_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Formatting'||'/','CONTAINER')], null)

-- 28. Rebind Test Logical Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/L2_2_Logical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Logical'||'/','CONTAINER')], null)
```

-- 29. Rebind Test Federated Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/L2_1_Federated_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Federated'||'/','CONTAINER')], null)

-- 30. Rebind Test Client Views

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/L1_2_Client_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Views'||'/','CONTAINER')], null)

-- 31. Rebind Test Client Services

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/L1_1_Client_Services'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Services'||'/','CONTAINER')], null)

-- 32. Rebind /shared/BestPractices

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('/shared/BestPractices'||'/','CONTAINER')], VECTOR[(bestPracticesRootPath||'/','CONTAINER')], null)

-- 33. Rebind constants

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/constants'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions'||'/','CONTAINER')], null)

-- 34. Rebind defaultValues

(1.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions/defaultValues','PROCEDURE')],
VECTOR[('$PROJECT_PATH_DST/"_scripts"/Constants/defaultValues','PROCEDURE')], null)

-- 35. Update CRUD procedures within the destination project path

(1.0, bestPracticesVersionDefault, 'updateCrud', null, null, null, VECTOR[('$PROJECT_PATH_DST','CONTAINER')], null)

Best Practices 5.0 thru 5.2 mapped to 7.x/8.x

The following section provides a description of how 5.0 thru 5.2 is mapped to 7.x/8.x.

-- 1. Copy Physical Metadata

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Metadata','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata','CONTAINER')], null)

-- 2. Copy Physical Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Physical','CONTAINER')], null)

-- 3. Copy Formatting Views

```
(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L3_Formatting_Layer/Formatting_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Formatting','CONTAINER')], null)

-- 4. Copy Logical Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Logical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Logical','CONTAINER')], null)

-- 5. Copy Federated Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Federated_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Federated','CONTAINER')], null)

-- 6. Copy Business Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Business_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Business','CONTAINER')], null)

-- 7. Copy Client Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Views','CONTAINER')], null)

-- 8. Copy Client Services

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Services','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services','CONTAINER')],          null)

-- 9. Copy constants to Definitions used by CRUD

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null, VECTOR[('$PROJECT_PATH_SRC/constants','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions','CONTAINER')],              null)

-- 10. Copy Test Physical Metadata

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Metadata','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Metadata','CONTAINER')], null)

-- 11. Copy Test Physical Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Physical','CONTAINER')], null)

-- 12. Copy Test Formatting Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L3_Formatting_Layer/Formatting_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Formatting','CONTAINER')], null)

-- 13. Copy Test Logical Views
```

```
(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Logical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Logical','CONTAINER')], null)

-- 14. Copy Test Federated Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Federated_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Federated','CONTAINER')], null)

-- 15. Copy Test Business Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Business_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Business','CONTAINER')], null)

-- 16. Copy Test Client Views

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Views','CONTAINER')], null)

-- 17. Copy Test Client Services

(5.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Services','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Services','CONTAINER')],   null)

-- 18. Rebind Physical Metadata

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Metadata'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata'||'/','CONTAINER')], null)

-- 19. Rebind Physical Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Physical'||'/','CONTAINER')], null)

-- 20. Rebind Formatting Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L3_Formatting_Layer/Formatting_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Formatting'||'/','CONTAINER')], null)

-- 21. Rebind Logical Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Logical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Logical'||'/','CONTAINER')], null)

-- 22. Rebind Federated Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Federated_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Federated'||'/','CONTAINER')], null)

-- 23. Rebind Business Views
```

```
(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Business_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Business'||'/','CONTAINER')], null)

-- 24. Rebind Client Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Views'||'/','CONTAINER')], null)

-- 25. Rebind Client Services

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Services'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services'||'/','CONTAINER')], null)

-- 26. Rebind Test Physical Metadata

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Metadata'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Metadata'||'/','CONTAINER')], null)

-- 27. Rebind Test Physical Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Physical'||'/','CONTAINER')], null)

-- 28. Rebind Test Formatting Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L3_Formatting_Layer/Formatting_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Formatting'||'/','CONTAINER')], null)

-- 29. Rebind Test Logical Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Logical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Logical'||'/','CONTAINER')], null)

-- 30. Rebind Test Federated Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Federated_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Federated'||'/','CONTAINER')], null)

-- 31. Rebind Test Business Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Business_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Business'||'/','CONTAINER')], null)

-- 32. Rebind Test Client Views

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Views'||'/','CONTAINER')], null)

-- 33. Rebind Test Client Services
```

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Services'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Services'||'/','CONTAINER')], null)

-- 34. Rebind /shared/BestPractices

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('/shared/BestPractices'||'/','CONTAINER')], VECTOR[(bestPracticesRootPath||'/','CONTAINER')], null)

-- 35. Rebind constants

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/constants'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions'||'/','CONTAINER')], null)

-- 36. Rebind defaultValues

(5.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions/defaultValues','PROCEDURE')],
VECTOR[('$PROJECT_PATH_DST/"_scripts"/Constants/defaultValues','PROCEDURE')], null)

-- 37. Update CRUD procedures within the destination project path

(5.0, bestPracticesVersionDefault, 'updateCrud', null, null, null, VECTOR[('$PROJECT_PATH_DST','CONTAINER')], null)

Best Practices 6.0 thru 6.6 mapped to 7.x/8.x

The following section provides a description of how 6.0 thru 6.6 is mapped to 7.x/8.x.

-- 1. Copy Physical Metadata

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Metadata','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata','CONTAINER')], null)

-- 2. Copy Physical Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Physical','CONTAINER')], null)

-- 3. Copy Formatting Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L3_Formatting_Layer/Formatting_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Formatting','CONTAINER')], null)

-- 4. Copy Logical Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Logical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Logical','CONTAINER')], null)

-- 5. Copy Federated Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Federated_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Federated','CONTAINER')], null)

```
-- 6. Copy Business Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Business_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Business','CONTAINER')], null)

-- 7. Copy Client Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Views','CONTAINER')], null)

-- 8. Copy Client Services

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Services','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services','CONTAINER')],          null)

-- 9. Copy Client Published

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Published','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Published','CONTAINER')], null)

-- 10. Copy constants to Definitions used by CRUD

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null, VECTOR[('$PROJECT_PATH_SRC/constants','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions','CONTAINER')],              null)

-- 11. Copy Test Physical Metadata

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Metadata','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Metadata','CONTAINER')], null)

-- 12. Copy Test Physical Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Physical','CONTAINER')], null)

-- 13. Copy Test Formatting Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L3_Formatting_Layer/Formatting_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Formatting','CONTAINER')], null)

-- 14. Copy Test Logical Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Logical_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Logical','CONTAINER')], null)

-- 15. Copy Test Federated Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Federated_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Federated','CONTAINER')], null)
```

```
-- 16. Copy Test Business Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Business_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Business','CONTAINER')], null)

-- 17. Copy Test Client Views

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Views','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Views','CONTAINER')], null)

-- 18. Copy Test Client Services

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Services','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Services','CONTAINER')],   null)

-- 19. Copy Test Client Published

(6.0, bestPracticesVersionDefault, 'copyChildren', null, null,
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Published','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Published','CONTAINER')], null)

-- 20. Rebind Physical Metadata

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Metadata'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Metadata'||'/','CONTAINER')], null)

-- 21. Rebind Physical Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L4_Physical_Layer/Physical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Physical'||'/','CONTAINER')], null)

-- 22. Rebind Formatting Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L3_Formatting_Layer/Formatting_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Physical/Formatting'||'/','CONTAINER')], null)

-- 23. Rebind Logical Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Logical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Logical'||'/','CONTAINER')], null)

-- 24. Rebind Federated Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Federated_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Federated'||'/','CONTAINER')], null)

-- 25. Rebind Business Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L2_Business_Layer/Business_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Business/Business'||'/','CONTAINER')], null)
```

```
-- 26. Rebind Client Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Views'||'/','CONTAINER')], null)

-- 27. Rebind Client Services

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Services'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services'||'/','CONTAINER')], null)

-- 28. Rebind Client Published

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L1_Mapping_Layer/Client_Published'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Published'||'/','CONTAINER')], null)

-- 29. Rebind Test Physical Metadata

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Metadata'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Metadata'||'/','CONTAINER')], null)

-- 30. Rebind Test Physical Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L4_Physical_Layer/Physical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Physical'||'/','CONTAINER')], null)

-- 31. Rebind Test Formatting Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L3_Formatting_Layer/Formatting_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Physical/Formatting'||'/','CONTAINER')], null)

-- 32. Rebind Test Logical Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Logical_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Logical'||'/','CONTAINER')], null)

-- 33. Rebind Test Federated Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Federated_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Federated'||'/','CONTAINER')], null)

-- 34. Rebind Test Business Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L2_Business_Layer/Business_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Business/Business'||'/','CONTAINER')], null)

-- 35. Rebind Test Client Views

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Views'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Views'||'/','CONTAINER')], null)
```

```
-- 36. Rebind Test Client Services

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Services'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Services'||'/','CONTAINER')], null)

-- 37. Rebind Test Client Published

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/L0_Testing_Layer/L1_Mapping_Layer/Client_Published'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Test/Application/Published'||'/','CONTAINER')], null)

-- 38. Rebind /shared/BestPractices

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('/shared/BestPractices'||'/','CONTAINER')], VECTOR[(bestPracticesRootPath||'/','CONTAINER')], null)

-- 39. Rebind constants

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_SRC/constants'||'/','CONTAINER')],
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions'||'/','CONTAINER')], null)

-- 40. Rebind defaultValues

(6.0, bestPracticesVersionDefault, 'rebind', '$PROJECT_PATH_DST', 'CONTAINER',
VECTOR[('$PROJECT_PATH_DST/Application/Services/CRUD/Definitions/defaultValues','PROCEDURE')],
VECTOR[('$PROJECT_PATH_DST/"_scripts"/Constants/defaultValues','PROCEDURE')], null)

-- 41. Update CRUD procedures within the destination project path

(6.0, bestPracticesVersionDefault, 'updateCrud', null, null, null, VECTOR[('$PROJECT_PATH_DST','CONTAINER')], null)
```