Demo: **Object Migration Audit**

BiToolkits can be used to detect issues pertaining to object migrations. It enables the administrator to proactively address potential migration issues to avoid delay. The object migration audit should be performed before and after object migration. The following “Before Migration” tasks can be performed using BiToolkits:

1. Find dependency errors *(missing components in target environment)*
2. Find potentially wrong version of objects will be migrated
3. Find version collisions *(object to be migrated is older version than target environment)*
4. Find potentially missed object changes *(developers thought they made the change but the object has the same version as in the target environment)*
5. Find objects maybe inadvertently renamed
6. Save object’s original folder path (assuming objects were moved to a migration folder)

The following “After Migration” tasks can be performed using BiToolkits:

1. Verify objects migrated perfectly onto the target environment. This is particularly helpful when a migration involves multiple packages
2. Restore migrated objects back to their original folder

This demo showcases following BiToolkits features:

- *Compare Objects*
- *Object Finder*
- *Recursive Search Engine*
- *Command Manager Integration*

BiToolkits Steps

- Make sure BiToolkits is setup to contain both source and target migration environments
- Suppose we receive the following object migration list from user, from Dev to QA

Object_Name	Object_ID
Customer set	8827907011D3EB22C000B4B2D86C964F
by income	8827904B11D3EB22C000B4B2D86C964F
5. Customer Analysis	EE25FFCA439CDF149F4EEE8386DD1C4D
Customer Details (new v2)	9F1CAA534D67F893798C109D36D3EFF9
Churn Value by Region and Month	4F27049840E565C0B6752A9FE63D3ECB
Cox Churn Predictions (Imported)	005C42554ECD0451296B9CA9A7204E66

Before Migration Tasks:

1. Move objects to migration folder

   a. Bring migration objects into ***BiToolkits***

   1) Use ***Object Finder*** *tool*
   2) Click ***Object ID Search*** *tool*
   3) Copy and paste object IDs into the search box
   4) Click ***Search***
   5) A new batch is created (6 objects)

   b. Move objects to migration folder

   1) Click ***Check*** button on top left corner to select all objects
   2) Click ***Move Objects*** *(under* ***Command Manager Action*** *tab)*
   3) Note the confirmation box indicates only 5 objects are scripted to move. This is due to one object “Customer Details (new v2)” is already residing in the migration folder
   4) Navigate to “Public Objects” and tick “BiToolkits Migration Demo”
   5) Click ***OK***
   6) Command Manager scripts are executing in the background. When done, check the completion status to make sure there are no errors

   c. To verify objects are moved to migration folder

   1) Click ***Check*** button on top left corner to select all objects
   2) Click ***Create New Batch*** *(under* ***Batch Manipulation*** *tab)*
   3) Notice objects are now located in “BiToolkits Migration Demo” folder

2. Find dependency errors *(missing components in target environment)*

   d. Compare Components

   1) Click ***Check*** button on top left corner to select all objects
   2) Tick ***Component Details: Med*** under ***Options*** tab on bottom right
   3) Tick ***Nested Levels*** under ***Object Analysis*** tab on bottom left corner
   4) Click ***Components*** button to search for nested components
   5) Close the “***frmImpactSource***” popup screen if shown
   6) A new batch is created (159 objects)
   7) Click ***Check*** button on top left corner to select all objects
   8) Click ***QA & Migration*** tab on bottom left corner
   9) Untick “With First Level Components” and “All Level Components”
   10) Click ***Compare Objects*** button
   11) Select target environment, MySQL-qa : MicroStrategy Tutorial
   12) Click ***OK*** (note Compare Objects screen is displayed)
   13) Click ***Select Unmatched*** button on top left corner
   14) Note there are 2 objects exist in Dev but missing in QA

       a) The first missing object, “by payment method” is a ***missing component.*** It will cause dependency error unless it is included in the migration package

b) The second missing object, “Customer Details (new v2)”, will not cause dependency error because it is in the migration package (in Dev’s “BiToolkits Migration Demo” folder)

c) Note to find out which object is depending on “by payment method”, select this object and run a ***Dependents*** search

3. Find rest of the issues

   a. Compare Objects

   1) Close ***Compare Objects*** screen
   2) Select the “Folderfinder” batch
   3) Click ***Check*** button on top left corner to select all objects
   4) Click ***QA & Migration*** tab on bottom left corner
   5) Untick “With First Level Components” and “All Level Components”
   6) Click ***Compare Objects*** button
   7) Select target environment, MySQL-qa : MicroStrategy Tutorial
   8) Click ***OK*** (note Compare Objects screen is displayed)

   b. Find potentially wrong version of objects will be migrated *(object not found in target environment, by comparing Object ID)*

   - Dev: Customer Details (new v2)
   - QA: Missing

   With proper object naming convention, this mistake is easy to spot; otherwise, we need to perform object name search in Object Finder to find all matching object names in Dev and compare them with QA to find the correct object to migrate.

   c. Find version collisions *(object to be migrated is older version than target environment)*

   - Dev > QA: Customer set
   - Dev < QA: by income
   - Dev < QA: Cox Churn Predictions (Imported)

   d. Find potentially missed object changes *(developers thought they made the change but the object has the same version as in the target environment)*

   - Dev = QA: 5. Customer Analysis
   - Dev = QA: Churn Value by Region and Month

   e. Find objects maybe inadvertently renamed

   - Dev: by income
   - QA: new by income

After Migration Tasks:

1. Compare migrated objects

Select the same batch used in Before Migration Task and run Compare Objects tool again. Object in Dev and QA should match perfectly.

2. Restore objects back to the original folder
   - Select QA environment
   - Select the same batch used in Before Migration Task and click “Restore to Original Folders”
   - You need to manually move newly created objects to proper folder
   - Note due to limitations in Command Manager, some objects types cannot be scripted. For those objects, you need to manually move them to their original folder.

Conclusion:

With only few clicks, we are able to find several potential issues pertaining to object migrations. We were able to move objects to a migration folder and restore them back to the original folder all within BiToolkits using Command Manager Scripts.