

# Advanced Data Migration Content Packaging

## User Guide

Contents

|  |           |
|--|-----------|
| <b>Introduction</b>  | <b>1</b>  |
| <b>Set Up the Content Extraction Utility</b>                           | <b>1</b>  |
| <b>Install the Utility</b>   | 1         |
| <b>Set Up the Utility</b>  | 2         |
| Add Utility to Stewardship Tier Site Menu                              | 2         |
| Add Users to WebApp  | 2         |
| <b>Configure Migration Content Pack</b>                                | 3         |
| <b>Create Content Pack DBVT Package from Template</b>                  | <b>5</b>  |
| ChangeContentPackName  | 5         |
| ChangeDGDatabaseName   | 6         |
| ChangeDSWDatabaseName  | 6         |
| ChangeSDBDatabaseName  | 7         |
| cMap   | 7         |
| Console  | 8         |
| Cranport   | 8         |
| Cransoft   | 8         |
| DSW  | 9         |
| <b>Step 1: Set Up Content Pack Folder</b>                              | 9         |
| <b>Step 2: Create Migration Content Pack from Template</b>             | 10        |
| <b>Step 3: Customize Target Extract Database Folder</b>                | 11        |
| <b>Step 4: Customize DSW Wave/Process Area Staging Database Folder</b> | 13        |
| <b>Step 5: Customize Source Database Folder</b>                        | 13        |
| <b>Step 6: Customize Export Batch File</b>                             | 14        |
| <b>Step 7: Customize Install Batch File</b>                            | 16        |
| <b>Step 8: Create Content Pack ReadMe.txt File</b>                     | 17        |
| <b>Analyze Migration Content Pack Database SQL</b>                     | <b>18</b> |
| Views  | 18        |
| Functions  | 18        |
| Common SQL Problems that Require Analysis/Cleanup                      | 18        |
| Misaligned SQL Object Internal versus External Name                    | 18        |
| Missing SQL Object Dependencies  | 19        |

|  |           |
|--|-----------|
| Cross Database Objects   | 19        |
| <b>Export Migration Content Pack Data and SQL</b>  | <b>20</b> |
| <b>Create System Type Extract Database from Template</b>   | <b>20</b> |
| <b>Step 1: Set up System Type Extract Folder</b>   | 21        |
| <b>Step 2: Customize System Type Extract Package from Template</b>   | 21        |
| <b>Step 3: Customize Install Batch File</b>  | 23        |
| <b>Step 4: Create System Type Extract Database</b>   | 23        |
| <b>Step 5: Select Required System Type for Extract</b>   | 25        |
| <b>Export Migration Content Pack System Type</b>   | <b>25</b> |
| <b>Install Migration Content Pack System Types</b>   | <b>26</b> |
| <b>Install Migration Content Pack Data</b>   | <b>26</b> |
| <b>Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier Application Tables Extracted</b> | <b>26</b> |
| <b>Appendix B: DBVT MS SQL 2008 Compatibility Fix</b>  | <b>28</b> |
| <b>Appendix C: DBVT Extract of SQL Objects with Special Characters Fix</b>                                     | <b>31</b> |
| <b>Appendix D: SQL View and Procedure Analysis SQL</b>   | <b>31</b> |

## Introduction

This document provides a step-by-step procedure that explains how a user can export Advanced Data Migration Content from one Stewardship Tier instance and install it on another instance. This document is used in conjunction with the following files:

- CONTENT\_EXTRACTION\_UTILITY.zip
- SYSTEM\_TYPE\_CONTENT\_PACK\_TEMPLATE.zip
- MIGRATE\_CONTENT\_PACK\_TEMPLATE.zip

These files, along with this manual, can be obtained by opening a ticket with support at <https://support.syniti.com> and requesting the **AdvancedDataMigrationContentPackagingUtility.zip**.

Refer to [Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier Application Tables Extracted](#) for details on which tables are included in an Advanced Data Migration Content Pack.

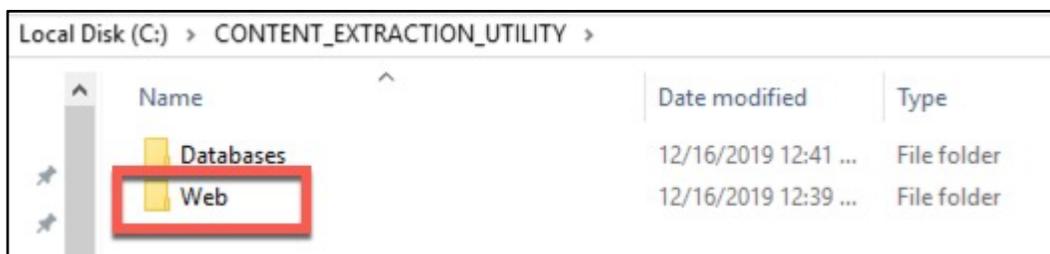
## Set Up the Content Extraction Utility

The Migration Content Pack configuration WebApp enables users to define a Content Pack by selecting which Waves, Process Areas and Data Source to include. These selections are then used by Content Pack DBVT templates, which can easily be configured to pull out the related data into a package. The package can then be installed on other Stewardship Tier instances.

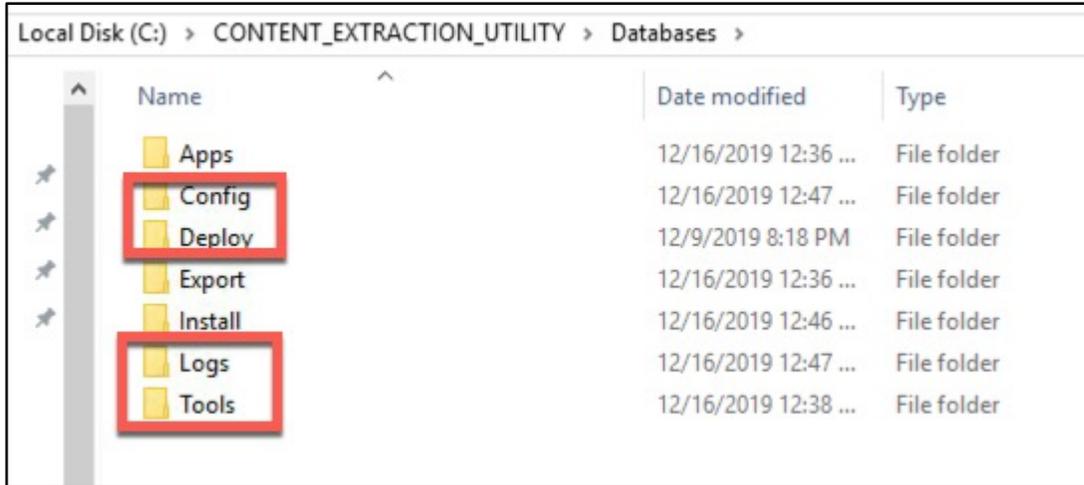
## Install the Utility

Perform the following steps to install the Utility:

1. Unzip the **CONTENT\_EXTRACTION\_UTILITY.zip** file.
2. Navigate to the Stewardship Tier installation folder (e.g., C:/Program Files (x86)/BOA/DSP).
3. Copy the web folder into the newly extracted Content\_Extraction\_Utility folder (e.g., creating C:/Content\_Extraction\_Utility/web). It should then look like the following screenshot:



4. Navigate to the **Stewardship Tier installation/Databases** folder.
5. Copy the **Config, Deploy, Logs** and **Tools** folders to the **Content\_Extraction\_Utility/Databases** folder. It should then look like the following screenshot:



6. Navigate to the **Content\_Extraction\_Utility/Databases/Install** folder.
7. Run the following .bat file **INSTALL\_PKG\_CONTENT\_EXTRACTION\_UTILITY.bat**.

### Set Up the Utility

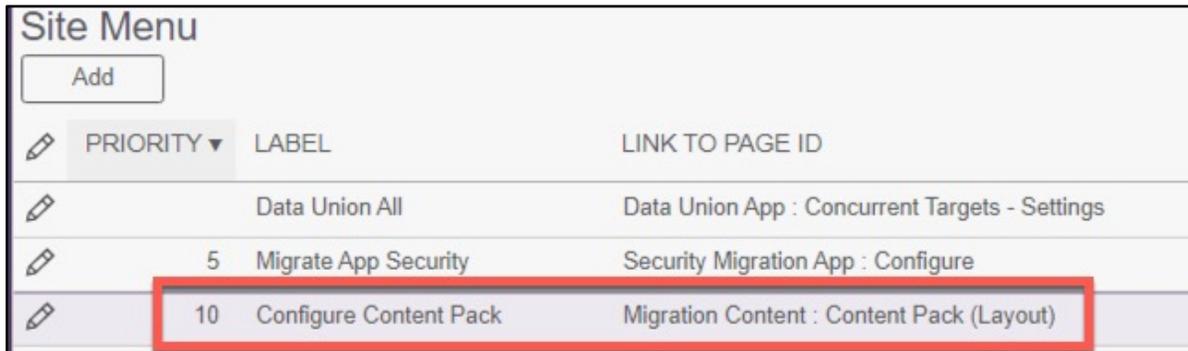
Once the Utility has been installed, the following steps must be performed before the utility can be used. These steps must be performed in the Stewardship Tier by an Administrator:

- Update Data Source Credentials
- Add Utility to Stewardship Tier Site Menu
- Add Users to WebApp

### Add Utility to Stewardship Tier Site Menu

To add the Utility to the Stewardship Tier site menu:

1. Select **Admin > Configuration > Site Menu** in the *Navigation* pane.
2. Add the **Content Pack Extract Utility** home page to the site menu.



### Add Users to WebApp

By default, the Stewardship Tier Administrator User has access to the Content Pack Extract Utility. If other users require access to perform the migration to the Content Pack application, they must be added to the WebApp and associated WebApp groups.

To add users to the WebApp:

1. Select **Admin > Security > WebApp Security** in the *Navigation* pane.
2. Click the **Users** icon for the Migration Content WebApp.
3. Click **Add** to add the user to the WebApp.
4. Select the user from the **User ID** list box.
5. Click **Save**.
6. Click **Groups** for the User ID.
7. Click **Add**.
8. Select **Power User** from **GROUP ID** list box.
9. Click **Save**.

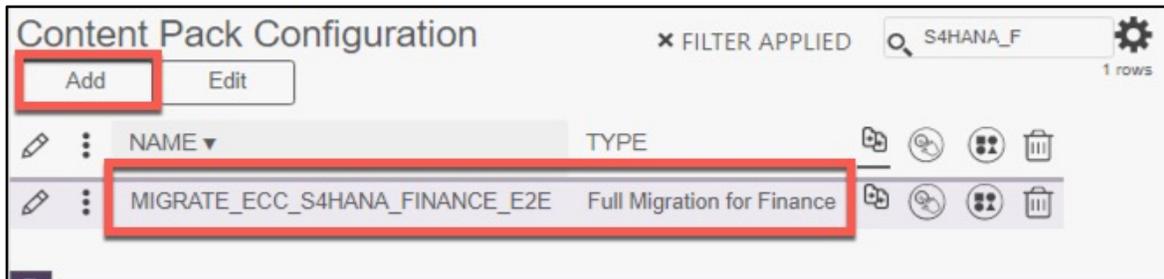
**NOTE:** After completing these steps, if the Configure Content Pack link does not display on the site menu, either refresh the browser, log out of the Stewardship Tier and log back in, or clear cache.

## Configure Migration Content Pack

Define a Content Pack and add relevant Waves, Process Areas and Data Sources.

To configure the Migration Content Pack:

1. Create the 'Content Pack' Header Record. Enter the name of the Content Pack. Do not use special characters other than underscore ( `_` ) in the name.



**NOTE:** If no record exists, the application prompts the user to enter the details for a new record, even without the user clicking Add.

2. Add Wave(s) that are in scope for the Content Pack.

**Content Pack Configuration** × FILTER APPLIED  1 rows

Add Edit

| NAME ▼                         | TYPE                       |  |  |  |  |
|--------------------------------|----------------------------|--|--|--|--|
| MIGRATE_ECC_S4HANA_FINANCE_E2E | Full Migration for Finance |  |  |  |  |

---

**Content Pack Wave**  1 rows

Add Edit

| CONTENT PACK ID ▼              | WAVE   |  |
|--------------------------------|--------|--|
| MIGRATE_ECC_S4HANA_FINANCE_E2E | S4HANA |  |

3. Add Process Areas that are in scope for the Content Pack.

**Content Pack Configuration** × FILTER APPLIED  1 rows

Add Edit

| NAME ▼                         | TYPE                       |  |  |  |  |
|--------------------------------|----------------------------|--|--|--|--|
| MIGRATE_ECC_S4HANA_FINANCE_E2E | Full Migration for Finance |  |  |  |  |

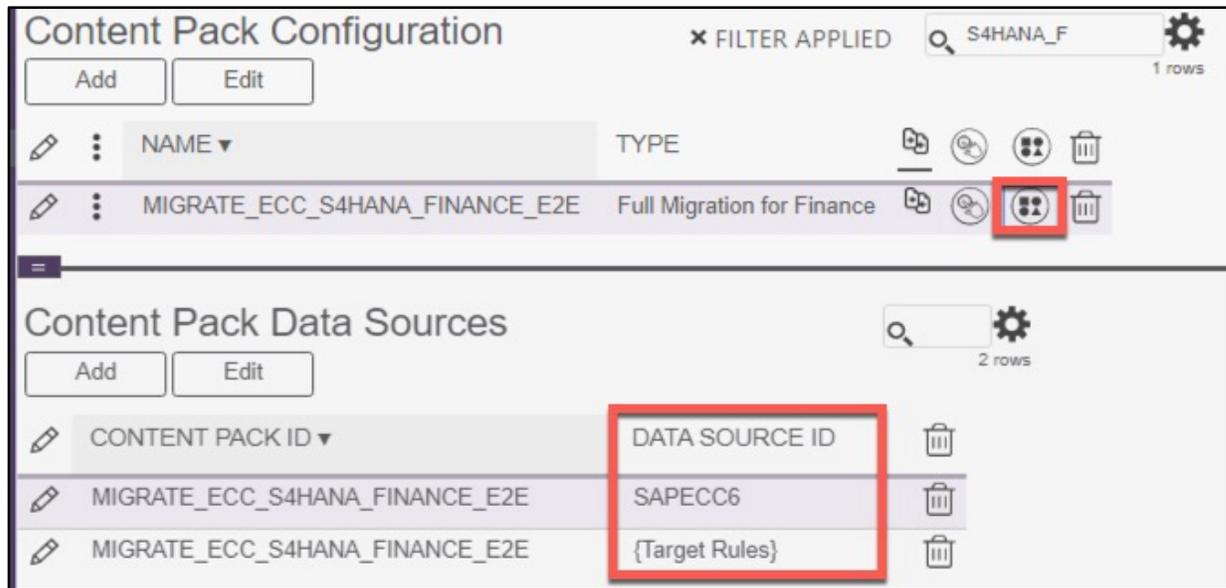
---

**Content Pack Process Areas**  2 rows

Add Edit

| CONTENT PACK ID ▼              | PROCESS AREA |  |
|--------------------------------|--------------|--|
| MIGRATE_ECC_S4HANA_FINANCE_E2E | PA_FI        |  |
| MIGRATE_ECC_S4HANA_FINANCE_E2E | PA_CO        |  |

4. Add Data Sources that are in scope for the Content Pack.



## Create Content Pack DBVT Package from Template

The Migrate Content Pack Template contains a series of folders and files that have been designed to interact with the Content Pack Extract utility to simplify the extraction of content from the Stewardship Tier.

The **Migrate\_Content\_Pack\_Template/Databases/Apps** folder contains 4 folders that contain files that need to be tailored to the specific content pack being created:

- ChangeContentPackName
- ChangeDGDatabaseName
- ChangeDSWDatabaseName
- ChangeSDBDatabaseName

## ChangeContentPackName

The files in the **ChangeContentPackName** folder are used to control the extraction (and installation) of Content from the different Stewardship Tier WebApps. Instructions for these processes are contained within the Database XML files and Operations Config file. The extracted data is also stored in the component's folder as JSON files. The ChangeContentPackName folder and Database XML file Instance variables must be aligned to the real content pack name.

The ChangeContentPackName folder contains these subfolders.

MIGRATE\_CONTENT\_PACK\_TEMPLATE > Databases > Apps > ChangeContentPackName >

| Name       | Date modified        | Type        |
|------------|----------------------|-------------|
| Components | 10/17/2019 11:31 ... | File folder |
| Databases  | 10/17/2019 11:51 ... | File folder |
| Operations | 10/17/2019 11:51 ... | File folder |

The ChangeContentPackName folder should exist only ONCE within a Content Pack.

### ChangeDGDatabaseName

The **ChangeDGDatabaseName** folder represents the Target Extract database e.g., dgSAP. This is a self-contained folder that is used to define and control the extraction (and installation) of SQL definitions of tables, views, procedures and functions. Instructions for these processes are contained within the Definition (Baseline and Objects) and Database XML files and Operations Config file. The extracted SQL scripts are stored in the component's folder as SQL files. The ChangeDGDatabaseName folder, Database XML and Config files need to be updated to align to the Target Extract database(s) that are used by the Content Pack.

The ChangeDGDatabaseName folder contains the following subfolders.

> MIGRATE\_CONTENT\_PACK\_TEMPLATE > Databases > Apps > ChangeDGDatabaseName >

| Name        | Date modified        | Type        |
|-------------|----------------------|-------------|
| Components  | 12/16/2019 5:52 PM   | File folder |
| Databases   | 10/17/2019 11:51 ... | File folder |
| Definitions | 10/17/2019 11:51 ... | File folder |
| Operations  | 10/17/2019 11:51 ... | File folder |

The ChangeDGDatabaseName folder may be repeated (with different names) within a Content Pack.

### ChangeDSWDatabaseName

The **ChangeDSWDatabaseName** folder represents the DSW staging databases that are used by the content pack e.g., dswWAVE1\_FIN, dswWAVE1\_FIN\_ReportCache. This is a self-contained folder that is used to define and control the extraction (and installation) of SQL definitions of tables, views, procedures and functions. The instructions for these processes are contained within the Definition (Baseline and Objects, Database XML files and Operations Config files. The extracted SQL scripts are stored in the component's folder as SQL files. The ChangeDSWDatabaseName folder, Database XML and Config files need to be copied and updated to align to the DSW database(s) that are used by the Content Pack.

The ChangeDSWDatabaseName folder contains the following subfolders.

| Name        | Date modified        | Type        |
|-------------|----------------------|-------------|
| Components  | 12/16/2019 5:52 PM   | File folder |
| Databases   | 10/17/2019 11:51 ... | File folder |
| Definitions | 10/17/2019 11:51 ... | File folder |
| Operations  | 10/17/2019 11:51 ... | File folder |

The ChangeDSWDatabaseName folder will be repeated for every DSW and DSW Report Cache database that is used by the Content Pack.

**IMPORTANT!** The data contained within DSW Databases should **NOT** be extracted. This is controlled by the Baseline definition.

### ChangeSDBDatabaseName

The **ChangeSDBDatabaseName** folder represents the Source Database staging databases that are used by the content pack e.g., sdbSAP\_ECC. This is a self-contained folder that is used to define and control the extraction (and installation) of SQL definitions of tables, views, procedures and functions. The instructions for these processes are contained within the Definition (Baseline and Objects), Database XML files and Operations Config file. The extracted SQL scripts are stored in the component's folder as SQL files. The ChangeSDBDatabaseName folder, Database XML and Config files must be copied and updated to align to the DSW database(s) that are used by the Content Pack.

The ChangeSDBDatabaseName folder contains the following subfolders.

| Name        | Date modified      | Type        |
|-------------|--------------------|-------------|
| Components  | 12/16/2019 5:53 PM | File folder |
| Databases   | 12/19/2019 6:48 AM | File folder |
| Definitions | 12/19/2019 6:48 AM | File folder |
| Operations  | 12/19/2019 6:48 AM | File folder |

The ChangeSDBDatabaseName folder will be repeated for every Source database that is used by the Content Pack.

**IMPORTANT!** The data contained within Source Databases should **NOT** be extracted. Data extraction is controlled by the Baseline definition; therefore, ensure that its settings are correct.

The Migrate\_Content\_Pack\_Template/Databases/Apps folder also contains 5 folders that contain definition files related to the content extraction process.

### cMap

The cMap folder contains the following definition:

| Name             | Date modified        | Type         |
|------------------|----------------------|--------------|
| FieldMapping.xml | 10/16/2019 10:03 ... | XML Document |

Refer to [Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier Application Tables Extracted](#) for details of which tables are extracted by this definition.

## Console

The Console folder contains the following definitions:

| Name                | Date modified        | Type         |
|---------------------|----------------------|--------------|
| ConsoleElements.xml | 10/16/2019 10:03 ... | XML Document |
| TargetDesign.xml    | 10/16/2019 10:03 ... | XML Document |
| WaveStructure.xml   | 10/16/2019 10:03 ... | XML Document |

Refer to [Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier Application Tables Extracted](#) for details of which tables are extracted by this definition.

## Cranport

The Cranport folder contains the following definition:

| Name                 | Date modified        | Type         |
|----------------------|----------------------|--------------|
| AssemblePackages.xml | 10/16/2019 10:03 ... | XML Document |

Refer to [Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier Application Tables Extracted](#) for details of which tables are extracted by this definition.

## Cransoft

The Cransoft folder contains the following definitions:

| Name                          | Date modified        | Type         |
|-------------------------------|----------------------|--------------|
| Content.SourceDataSources.xml | 10/16/2019 10:03 ... | XML Document |
| Content.TargetDataSources.xml | 10/16/2019 10:03 ... | XML Document |

Refer to [Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier](#) Application Tables Extracted for details of which tables are extracted by this definition.

## DSW

The DSW folder contains the following definition:

| Name          | Date modified        | Type         |
|---------------|----------------------|--------------|
| Transform.xml | 10/16/2019 10:03 ... | XML Document |

Refer to [Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier](#) Application Tables Extracted for details of which tables are extracted by this definition.

## Step 1: Set Up Content Pack Folder

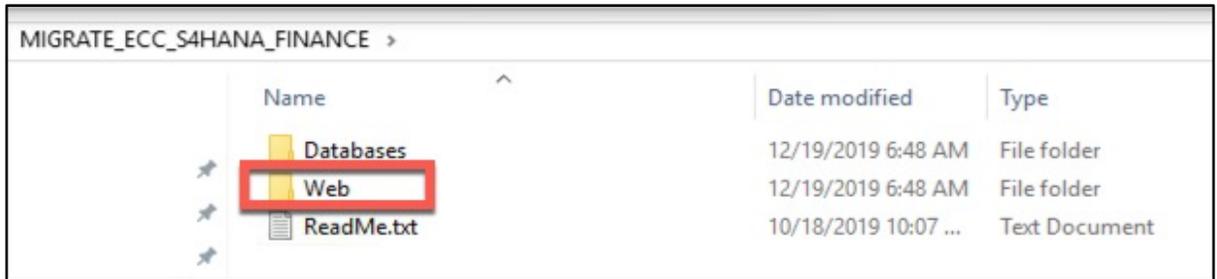
To set up the Content Pack folder:

1. Unzip the **MIGRATE\_CONTENT\_PACK\_TEMPLATE.zip** file.
2. Rename Folder to the name of Content Pack.

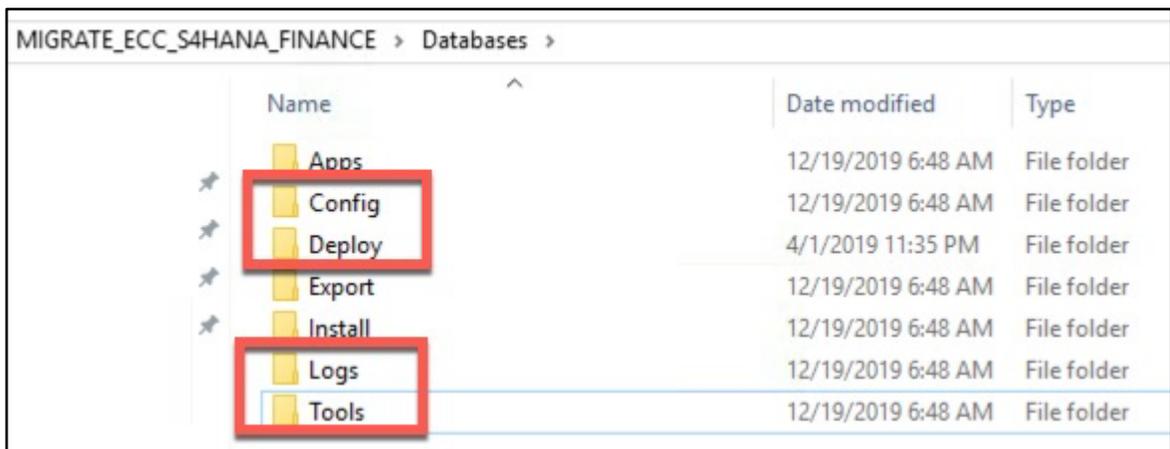
| Name       | Date modified        | Type          |
|------------|----------------------|---------------|
| Databases  | 12/19/2019 6:48 AM   | File folder   |
| ReadMe.txt | 10/18/2019 10:07 ... | Text Document |

**NOTE:** It is recommended that the name of Content Pack created in the Content Pack Extraction Utility is used.

3. Navigate to the Stewardship Tier installation folder (e.g., C:/Program Files (x86)/BOA/DSP).
4. Copy the web folder into the newly renamed folder (e.g., creating C:/MIGRATE\_ECC\_S4HANA\_FINANCE/Web). It should then look like the following screenshot:



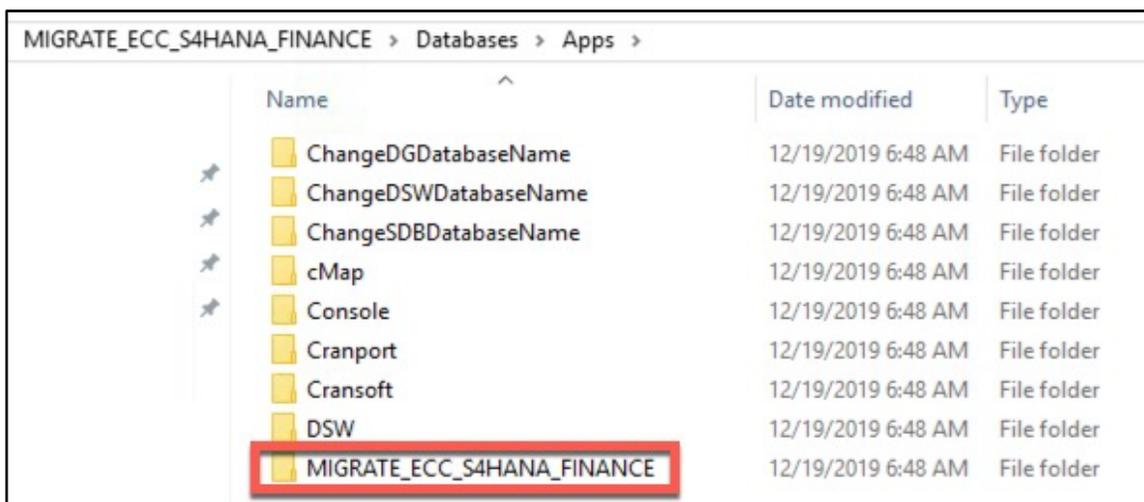
5. Navigate to the Stewardship Tier installation/Databases folder (e.g. C:/Program Files (x86)/BOA/DSP/Databases).
6. Copy the **Config, Deploy, Logs** and **Tools** folders into the **ContentPack/Databases** folder. It should then look like the following screenshot:



## Step 2: Create Migration Content Pack from Template

To create a Migration Content Pack from a Template:

1. Rename the ChangeContentPackName folder to the name of Content Pack.



- Set the value of the variable named "ContentPackName" to the name of the content pack in all XML files in the database folder.

| Name         | Date modified      | Type         | Size |
|--------------|--------------------|--------------|------|
| cMap.xml     | 12/19/2019 7:03 AM | XML Document | 1 KB |
| Console.xml  | 12/19/2019 6:47 AM | XML Document | 1 KB |
| Cranport.xml | 12/19/2019 6:47 AM | XML Document | 1 KB |
| Cransoft.xml | 12/19/2019 6:47 AM | XML Document | 1 KB |
| DSW.xml      | 12/19/2019 6:47 AM | XML Document | 1 KB |

```
<datasource name="cMap">
  <setup databaseVersionLogTableName="ztContentPack_DatabaseVersionLog" />
  <components>
    <component type="FieldMapping">
      <instances>
        <instance name="cMap">
          <variables>
            <variable name="ContentPackName" value="MIGRATE_ECC_S4HANA_FINANCE" />
          </variables>
        </instance>
      </instances>
    </component>
  </components>
</datasource>
```

**IMPORTANT!** This value MUST be the same as the name of the Content Pack created in the Content Pack Extraction Utility.

### Step 3: Customize Target Extract Database Folder

To customize the Target Extract Database folder:

If there are more than one Target Extract database in the content pack, copy theChangeDGDatabaseName folder so that there is a folder for each database.

Repeat the following steps for each Target Extract database:

- Rename the **ChangeDGDatabaseName** folder to the name of Target Extract Database.

MIGRATE\_ECC\_S4HANA\_FINANCE > Databases > Apps >

| Name                       | Date modified      | Type        |
|----------------------------|--------------------|-------------|
| ChangeDSWDatabaseName      | 12/19/2019 6:48 AM | File folder |
| ChangeSDBDatabaseName      | 12/19/2019 6:48 AM | File folder |
| cMap                       | 12/19/2019 6:48 AM | File folder |
| Console                    | 12/19/2019 6:48 AM | File folder |
| Cranport                   | 12/19/2019 6:48 AM | File folder |
| Cransoft                   | 12/19/2019 6:48 AM | File folder |
| dgSAP                      | 12/19/2019 6:48 AM | File folder |
| DSW                        | 12/19/2019 6:48 AM | File folder |
| MIGRATE_ECC_S4HANA_FINANCE | 12/19/2019 6:48 AM | File folder |

2. Rename the XML files in the database folder to the name of Target Extract database.

MIGRATE\_ECC\_S4HANA\_FINANCE > Databases > Apps > dgSAP > Databases

| Name      | Date modified        |
|-----------|----------------------|
| dgSAP.xml | 10/17/2019 11:42 ... |

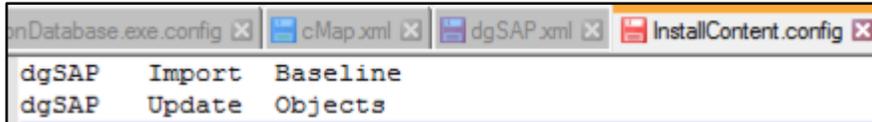
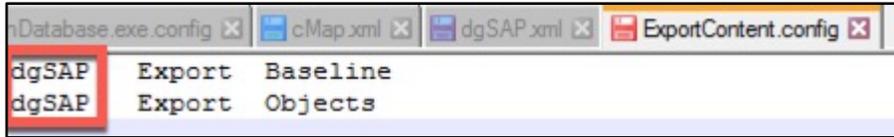
3. Update the Datasource name in the XML file to the name of Target Extract database.

```
<datasource name="dgSAP">
  <setup externalConfigType="CranSoftInstallation"
    databaseVersionLogTableName="Version" />
  <components>
    <component type="Baseline" />
    <component type="Objects" />
  </components>
</datasource>
```

4. Update the Configuration files.

MIGRATE\_ECC\_S4HANA\_FINANCE > Databases > Apps > dgSAP > Operations

| Name                  | Date modified        |
|-----------------------|----------------------|
| ExportContent.config  | 12/19/2019 12:29 ... |
| InstallContent.config | 10/17/2019 11:43 ... |



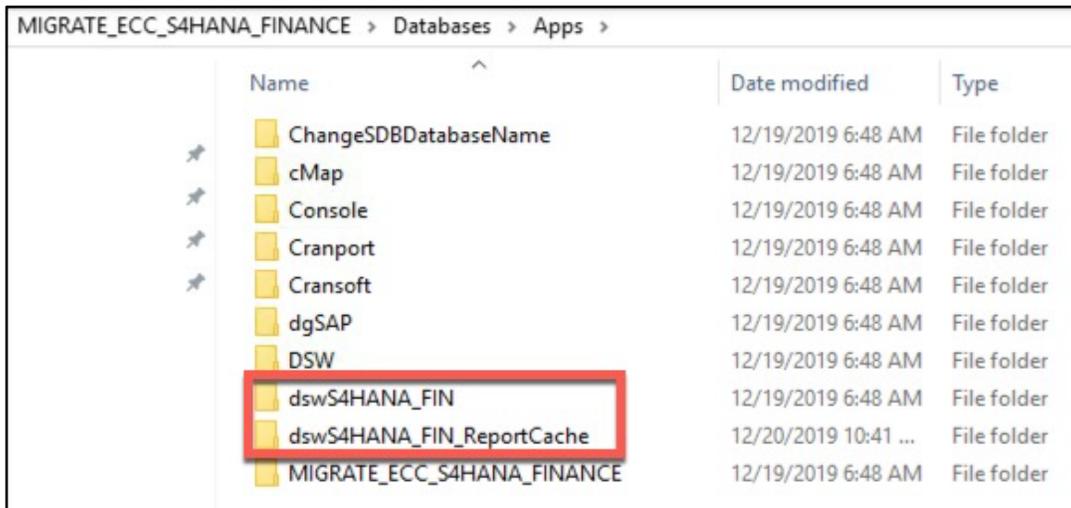
**IMPORTANT!** Ensure there is a TAB between the database name and operation e.g. Export, Import or Update.

5. Rename the XML files in the database folder to the name of Target Extract database.

#### Step 4: Customize DSW Wave/Process Area Staging Database Folder

To customize the DSW Wave/Process Area Staging Database folder:

The steps in the previous section [Step 3: Customize Target Extract Database Folder](#) must be repeated for ALL DSW staging and report cache databases that are used in the Content Pack. The ChangeDSWDatabaseName folder has been provided as the template.



#### Step 5: Customize Source Database Folder

To customize the Source Database folder:

1. The steps in the previous section [Step 3: Customize Target Extract Database Folder](#) must be repeated for ALL Source databases that are used in the Content Pack. The **ChangeSDBDatabaseName** folder has been provided as the template.

MIGRATE\_ECC\_S4HANA\_FINANCE > Databases > Apps >

| Name                       | Date modified        | Type        |
|----------------------------|----------------------|-------------|
| cMap                       | 12/19/2019 6:48 AM   | File folder |
| Console                    | 12/19/2019 6:48 AM   | File folder |
| Cranport                   | 12/19/2019 6:48 AM   | File folder |
| Cransoft                   | 12/19/2019 6:48 AM   | File folder |
| dgSAP                      | 12/19/2019 6:48 AM   | File folder |
| DSW                        | 12/19/2019 6:48 AM   | File folder |
| dswS4HANA_FIN              | 12/19/2019 6:48 AM   | File folder |
| dswS4HANA_FIN_ReportCache  | 12/20/2019 10:41 ... | File folder |
| MIGRATE ECC S4HANA_FINANCE | 12/19/2019 6:48 AM   | File folder |
| sdbSAPECC6                 | 12/19/2019 6:48 AM   | File folder |

### Step 6: Customize Export Batch File

The content pack template includes a Batch file that is run to export the data and SQL that is specified in the various definition files. This file must be tailored to export the Stewardship Tier application data and Target / Source / DSW Staging database table, view and procedures SQL definitions related to the content pack being created.

To customize the Export Batch file:

1. Navigate to the **Content Pack/Databases/Export** folder and rename the .Bat file. It is recommended that the name of the Content Pack is used.

MIGRATE\_ECC\_S4HANA\_FINANCE > Databases > Export

| Name                                      |
|---|
| EXPORT_PKG_MIGRATE_ECC_S4HANA_FINANCE.bat |

2. Right click the .bat file and open it in an appropriate editor, such as Notepad.
3. Modify the references to the Content Pack, the Target databases, the Source databases and the DSW staging databases to be aligned to the Content Pack. Repeat if required.

```

title Export [CONTENT_PACK_NAME] Package

echo Exporting [CONTENT_PACK_NAME] Package

call CD..
call Tools\RunOperation [CONTENT_PACK_NAME] ExportContent

if "%ERRORLEVEL%" NEQ "0" (
    echo ERROR: [PKG CONTENT_PACK_NAME] Export failed.
    goto Failure
)

:: REPEAT THIS SECTION FOR ALL SOURCE DATABASES

echo Exporting [SourceDatabaseName]

call Tools\RunOperation [SourceDatabaseName] ExportContent

if "%ERRORLEVEL%" NEQ "0" (
    echo ERROR: [SourceDatabaseName] Export failed.
    goto Failure
)

:: REPEAT THIS SECTION FOR ALL TARGET DATABASES

echo Exporting [dgDatabaseName]

call Tools\RunOperation [dgDatabaseName] ExportContent

if "%ERRORLEVEL%" NEQ "0" (
    echo ERROR: [dgDatabaseName] Export failed.
    goto Failure
)

```

```

:: REPEAT THIS SECTION FOR ALL DSW STAGING AND REPORT CACHE DATABASES

echo Exporting [dswDatabaseName]

call Tools\RunOperation [dswDatabaseName] ExportContent

if "%ERRORLEVEL%" NEQ "0" (
    echo ERROR: [dswDatabaseName] Export failed.
    goto Failure
)

goto Success

```

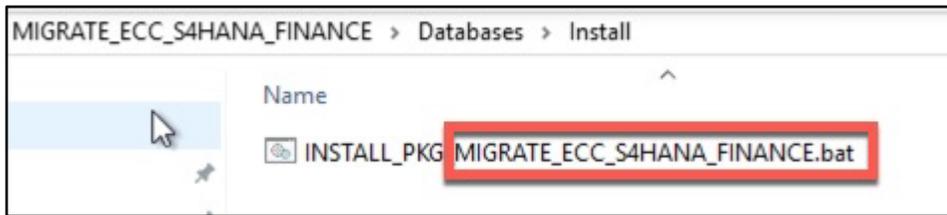
**TIP!** The initial export of SQL objects requires some trial and error. Therefore, when testing the export process, consider creating one .bat file for the application data and individual .bat files for

each of the other databases. Testing the extract process one database at a time allows errors to be identified quickly.

## Step 7: Customize Install Batch File

The content pack template includes a batch file that is run to install the data and SQL scripts that are contained within the Component files of the various folders. This file needs to be tailored to install the application data and Target / Source / DSW Staging database table, view and procedures SQL definitions that were extracted.

1. Navigate to the **Content Pack/Databases/Install** folder and rename the .Bat file. It is recommended that the name of the Content Pack is used.



2. Right click the .bat file and open it in an appropriate editor, such as Notepad.
3. Modify the references to the Content Pack, the Target databases, the Source databases and the DSW staging databases to be aligned to the Content Pack. Repeat if required.

```

title Install CONTENT_PACK_NAME Package

echo Installing CONTENT_PACK_NAME Package

call CD..
call Tools\RunOperation CONTENT_PACK_NAME InstallContent

if "%ERRORLEVEL%" NEO "0" (
    echo ERROR: PKG CONTENT_PACK_NAME Install failed.
    goto Failure
)

:: REPEAT THIS SECTION FOR ALL SOURCE DATABASES

echo Installing SourceDatabaseName

call Tools\RunOperation SourceDatabaseName InstallContent

if "%ERRORLEVEL%" NEO "0" (
    echo ERROR: SourceDatabaseName Install failed.
    goto Failure
)

:: REPEAT THIS SECTION FOR ALL TARGET DATABASES

echo Installing dgDatabaseName

call Tools\RunOperation dgDatabaseName InstallContent

if "%ERRORLEVEL%" NEO "0" (
    echo ERROR: dgDatabaseName Install failed.
    goto Failure
)

```

```

:: REPEAT THIS SECTION FOR ALL DSW STAGING AND REPORT CACHE DATABASES

echo Installing dswDatabaseName

call Tools\RunOperation dswDatabaseName InstallContent

if "%ERRORLEVEL%" NEO "0" (
    echo ERROR: dswDatabaseName Install failed.
    goto Failure
)

goto Success

```

**Step 8: Create Content Pack ReadMe.txt File**

A ReadMe.txt file is stored in the root folder of the content pack template. This file contains some template words to explain what is contained within the Content Pack and what steps need to be done to install it. Update this file to reflect the details of the specific content pack.

Copyright © 2020 BackOffice Associates, LLC dba Syniti and/or its affiliates. All rights reserved. This document contains confidential and proprietary information and reproduction is prohibited unless authorized by BackOffice Associates. Other names appearing on the Site may be trademarks of their respective owners.

## Analyze Migration Content Pack Database SQL

When using this process to extract migration content from a project Stewardship Tier instance, there is likely to be a very large number of SQL tables, views, procedures and functions, some of which will be erroneous and others that will be problematic during extract/installation e.g. due to dependencies across other databases.

**NOTE:** These issues are likely to be limited to the DSW staging databases (not Report Cache). The other databases are likely to mainly contain tables and simple views.

A few SQL views/functions have been provided within the zMig\_Content database (installed with the Content Pack Extraction Utility) to support the analysis of objects within the various SQL databases. It is recommended that these views are created within each DSW staging database (not Report Cache) to support the analysis and cleanup of their SQL objects.

Review [Appendix C: DBVT Extract of SQL Objects with Special Characters Fix](#)

If the Migration Content export fails with an error similar to below, it is because the version of the DBVT installed does not support the extract of SQL objects that contain special characters that are not compatible with file names.

```
Server and database configured via external configuration.
Operations will be performed against the resolved database.

Original.....: dgSAP
Resolved.....: dgSAP

Scripting schema.
Could not find a part of the path 'C:\TABLE\NAME.sql'.
Error encountered during execution without verbose mode.
To enable verbose output, add -Verbose to the execution parameters.

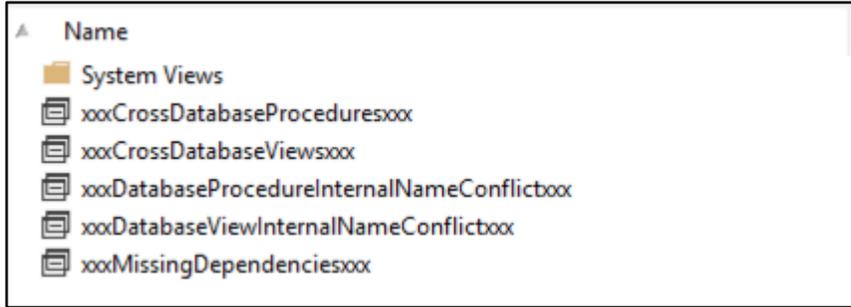
Errored: 1/14/2020 4:12:09 PM
Duration: 00:00:01
```

This issue can be fixed by performing the following steps:

1. In the root of the AdvancedDataMigrationContentPackagingUtility.zip file, copy the **CranSoft.DatabaseVersioning.Library.dll**.
2. Replace file **CranSoft.DatabaseVersioning.Library.dll** in folder ContentPack\Databases\Tools\FOR\_SMO\_10 with this file.
3. Rerun Export of Migration Content.

Appendix D: SQL View and Procedure Analysis SQL\_for the SQL Statements related to these views and procedures.

## Views



## Functions

The following function is used



## Common SQL Problems that Require Analysis/Cleanup

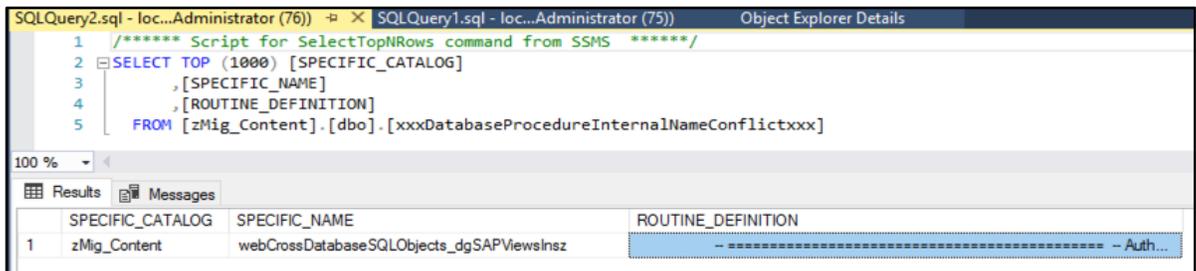
### Misaligned SQL Object Internal versus External Name

**Cause:** Renaming an Object in SSMS

**How to Identify Issues:** Run the following views:

- xxxDatabaseViewInternalNameConflictxxx
- xxxDatabaseProcedureInternalNameConflictxxx

Example of returned result:



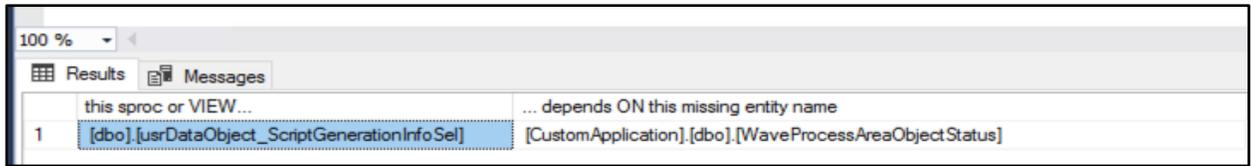
**How to Fix Issues:** Drop and Recreate SQL Object

### Missing SQL Object Dependencies

**Cause:** Dependent View, Procedure or Table Deleted or Changed

**How to Identify Issues:** run the xxxMissingDependenciesxxx view.

Example of returned result:



**How to Fix Issues:** These SQL objects are not valid (will fail to install) and there need to be resolved on a case-by-case basis. If they cannot be resolved, they should be excluded from relevant Baseline or Objects Definition

### Cross Database Objects

**Cause:** Procedures or Views Created that Use SQL Objects from Other Databases

**How to Identify Issues:**

- Run View xxxCrossDatabaseViewsxxx
- Run View xxxCrossDatabaseProceduresxxx

Example of returned result:

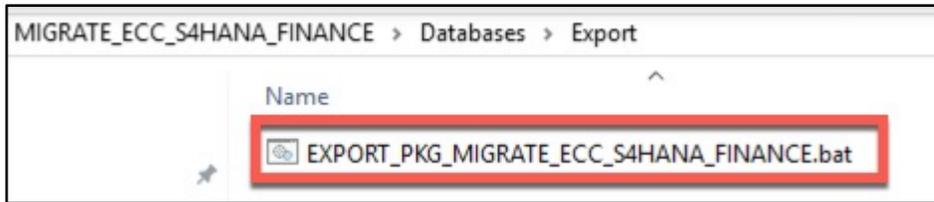
| view   | referenced_database_name | referenced_schema_name | referenced_entity_name  |
|--|--------------------------|------------------------|-------------------------|
| webTarget_ExtendServiceInsSel                        | Console                  | dbo                    | tt Target               |
| webMappingActionList                                 | cMap                     | dbo                    | ztAction                |
| trTargetDataDesignStaging_WaveProcessAreaObjectID... | Console                  | dbo                    | ttObject                |
| trTargetDataDesignStaging_WaveProcessAreaObjectID... | Console                  | dbo                    | ttWaveProcessAreaObject |
| webTargetField_Extend_SystemTypeFieldFoundUpdSel     | Console                  | dbo                    | tt Target               |
| webTargetField_Extend_SystemTypeFieldFoundUpdSel     | Console                  | dbo                    | tt TargetField          |
| webTargetField_Extend_SystemTypeFieldFoundUpdSel     | DSPCommon                | dbo                    | ztSystemType            |
| webTargetField_Extend_SystemTypeFieldFoundUpdSel     | DSPCommon                | dbo                    | ztSystemTypeTable       |
| webTargetField_Extend_SystemTypeFieldFoundUpdSel     | DSPCommon                | dbo                    | ztSystemTypeTableField  |
| trTargetDataDesignStaging_WaveProcessAreaObject T... | Console                  | dbo                    | tt Target               |

**NOTE:** There are processes and automation steps in Stewardship Tier that automatically create views that cross reference objects in other databases. Examples include System Views that start with boa\* or views associated with cMap Value mapping source value inserts that contain \*xt\*InsSel in the name.

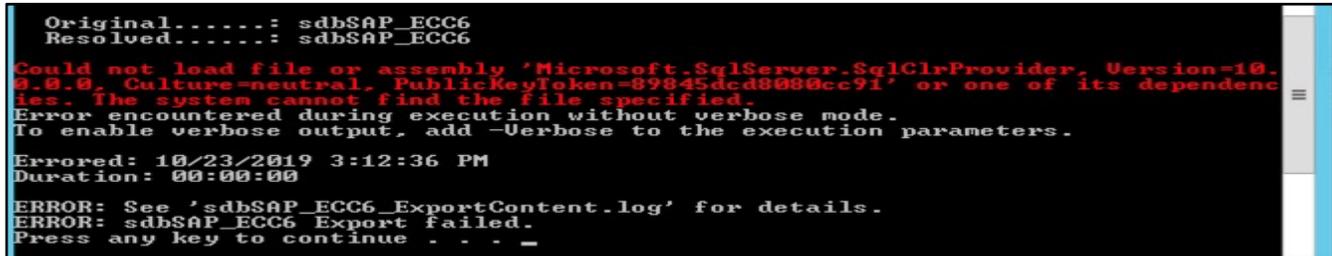
**How to Fix Issues:** The items returned by these views are not necessarily problems that require a fix; however, they may determine the sequence in which the databases are installed. If there are reciprocal dependencies, then modifications to the content pack extraction process through the use of 'StubObjects,' Delta or Post Installation scripts may be required.

### Export Migration Content Pack Data and SQL

1. Navigate to the Content Pack/Databases/Export folder and run the .Bat file by double-clicking it.



**NOTE:** If the export fails with an error similar to below, it is because the version of DBVT installed relies upon MS SQL Server 2008 components. Refer to [Appendix B: DBVT MS SQL 2008 Compatibility Fix](#) for details to fix this error.



**TIP 1!** If the export fails, navigate to the Databases/Logs folder and review the log file for details of the error.

**TIP 2!** If there are issues with the SQL extraction, the exports can be broken down into multiple bat files.

**TIP 3!** If there are considerable problems with the SQL extraction process, consider creating a backup of the troublesome databases and restoring the database to another location where cleanup activities can take place.

**NOTE:** Always consider data security and never transfer actual data (SQL is fine) from a client server without permission.

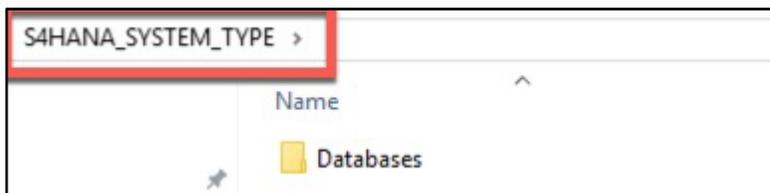
## Create System Type Extract Database from Template

This section of the document explains the process for extracting a System Type from one Stewardship Tier instance and installing it on another. This process does not use the DBVT for extracting and installing System Type data because a full system type created from a System Type model can include several million records, which is too much for the DBVT to handle in a performance way.

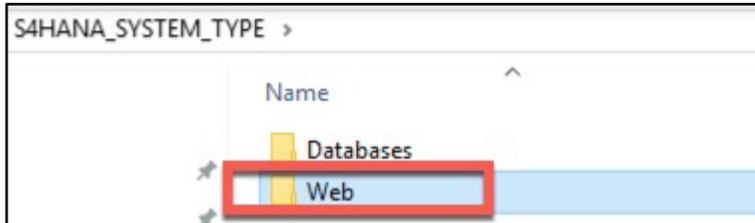
### Step 1: Set up System Type Extract Folder

To set up the System Type Extract folder:

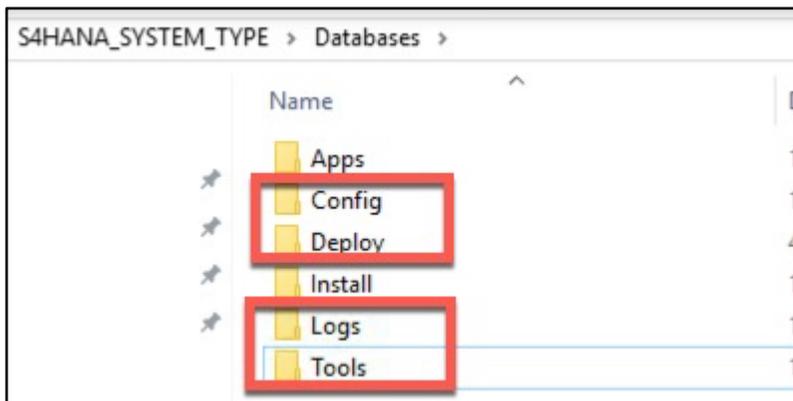
1. Unzip the **SYSTEM\_TYPE\_EXTRACT\_TEMPLATE.zip** file.
2. Rename Folder to the name of the System Type.



3. Navigate to the Stewardship Tier installation folder (e.g., C:/Program Files (x86)/BOA/DSP).
4. Copy the web folder into the newly renamed folder (e.g., creating C:/S4HANA\_SYSTEM\_TYPE/Web). It should then look like the following screenshot:



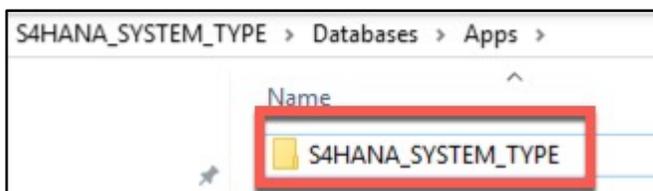
5. Navigate to the Stewardship Tier **installation/Databases** folder (e.g., C:/Program Files (x86)/BOA/DSP/Databases).
6. Copy the **Config, Deploy, Logs** and **Tools** folders into the **SystemType/Databases** folder. It should then look like the following screenshot:



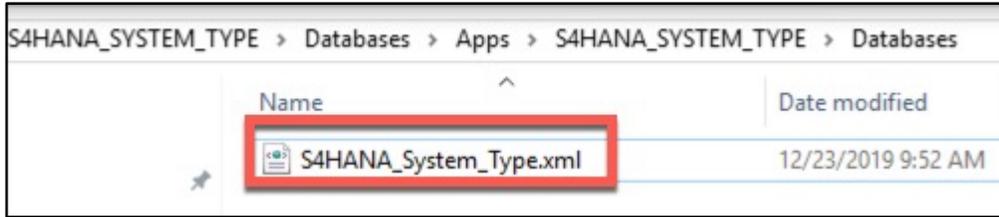
## Step 2: Customize System Type Extract Package from Template

To customize the System Type Extract Package from the template:

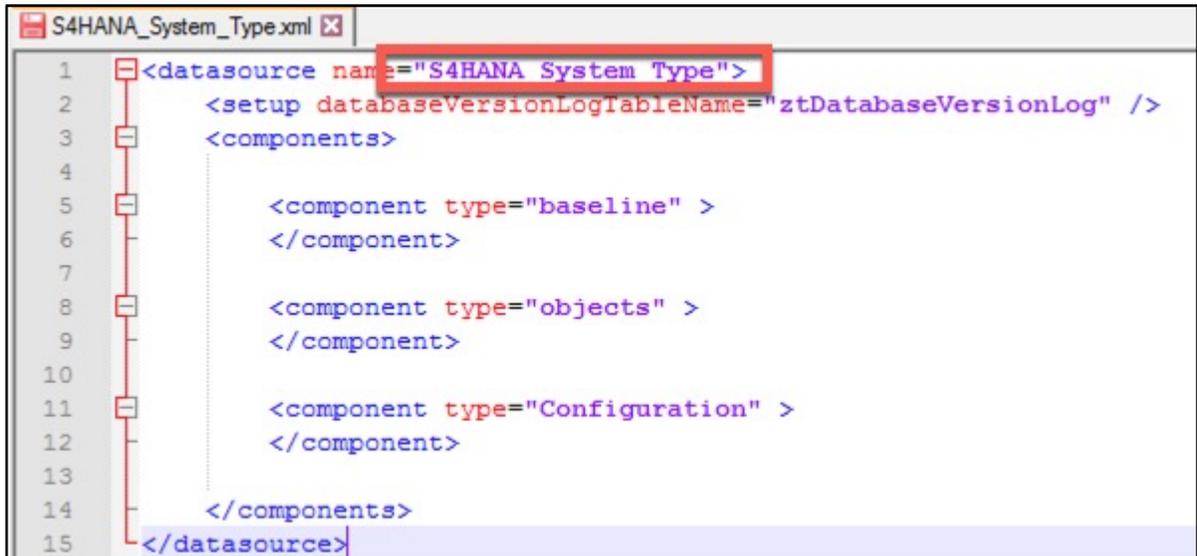
1. Rename the **ChangeMe\_System\_Type** folder to the name of the System Type Extract database that will be created.



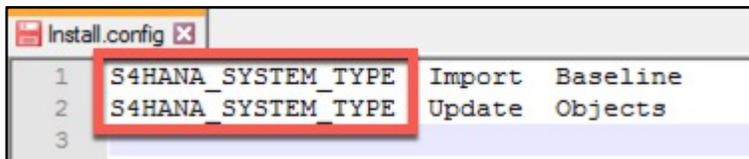
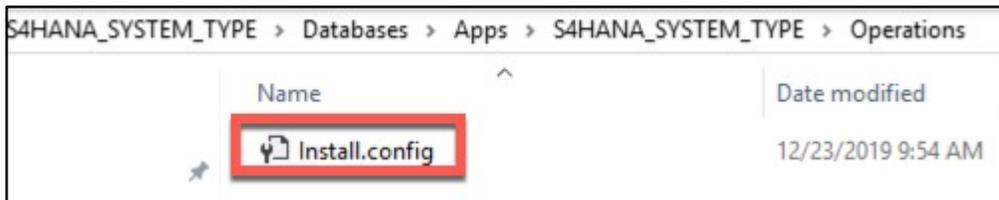
2. Rename the XML files in the database folder to the name of System Type Extract database.



- Update the Datasource name in the XML file to the name of System Type Extract database.



- Update the Configuration files.

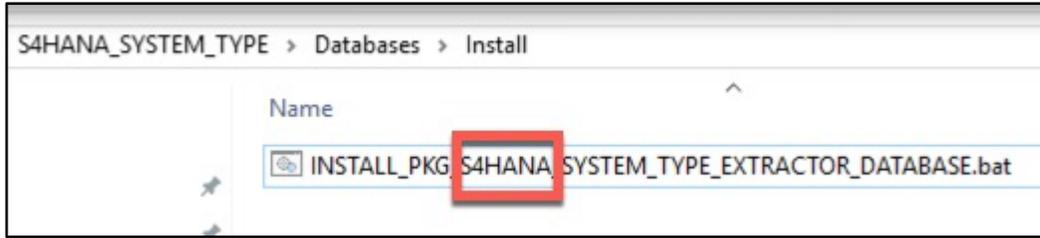


### Step 3: Customize Install Batch File

The System Type extract template includes a batch file that is run to create a database that is used to extract System Type data. This file needs to be tailored to create the required database.

To customize the Install Batch file:

- Navigate to the **System Type/Databases/Install** folder and rename the .Bat file.



2. Right click the .bat file and open it in an appropriate editor, such as Notepad.
3. Replace references of the word “ChangeMe” with the name of the Content Pack or System Type Extract Database.

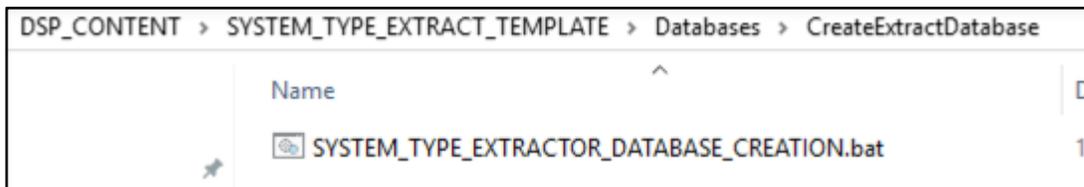
```

1
2 @echo off
3
4 pushd %~dp0
5
6 setlocal
7
8 title Create SystemType Extraction Database Package
9
10 echo Creating S4HANA System_Type Extraction Database
11
12 call CD..
13 call Tools\RunOperation S4HANA System_Type Install
14
15 if "%ERRORLEVEL%" NEO "0" (
16     echo ERROR: S4HANA System_Type Import failed.
17     goto Failure
18 )
19
20
21 goto Success
  
```

#### Step 4: Create System Type Extract Database

To create the System Type Extract database:

1. Navigate to the **System Type/Databases/CreateExtractDatabase** folder.
2. Run the .Bat file by double-clicking it.



Running the .Bat file opens a command line window and runs through the process of creating the System Type Extract database.

```

Administrator: Create SystemType Extraction Database Package

Began: 12/23/2019 10:27:49 AM

Application...: S4HANA_System_Type
Data Source...: S4HANA_SYSTEM_TYPE
Operation....: Update
Component....: objects
Server.....: localhost

Server and database configured via external configuration.
Operations will be performed against the resolved database.

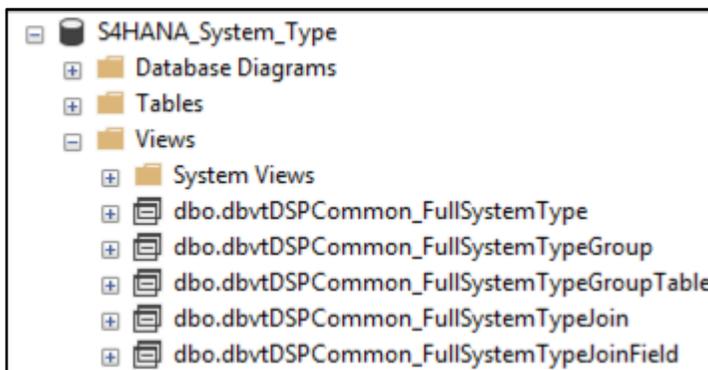
Original.....: S4HANA_System_Type
Resolved.....: S4HANA_System_Type

Verifying database state.
Executing pre-install scripts.
Loading objects.
Loading Procedures.
Loading Views.
Dropping all existing SQL objects.
Creating SQL objects.
  Created 10 Views.
  Created 18 Procedures.
Executing post-install scripts.

Completed: 12/23/2019 10:27:50 AM
Duration: 00:00:00

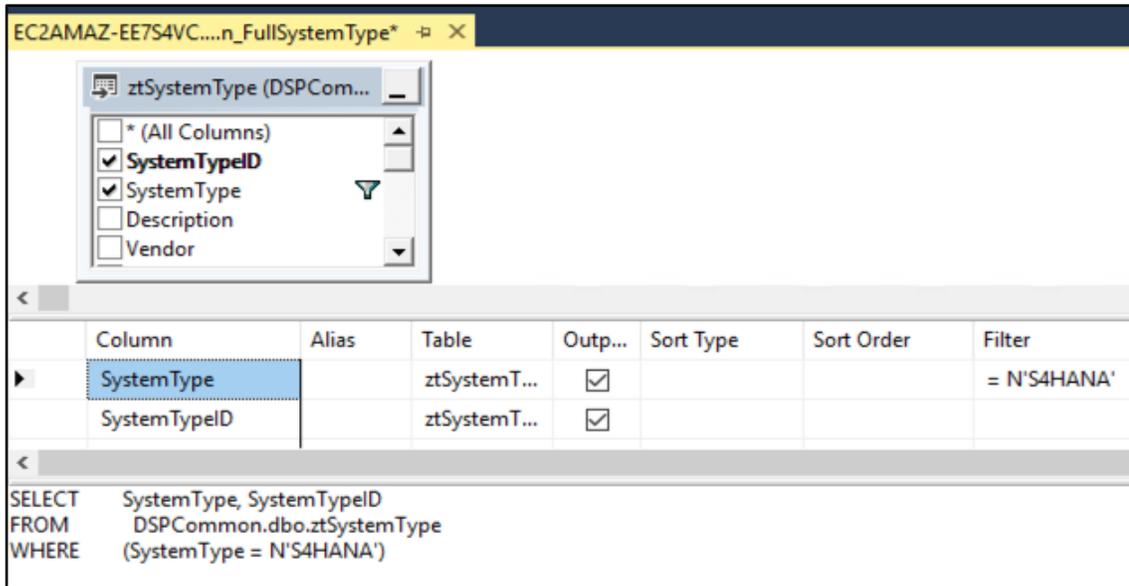
Press any key to continue . . . _
  
```

A database with the same name as the system type /databases folder will be created with the tables, views and procedures required to extract a system type from one Stewardship Tier instance and to install it on another.



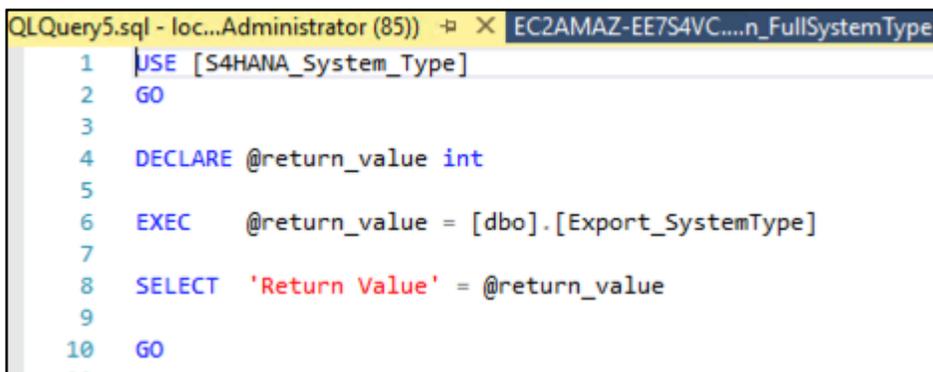
### Step 5: Select Required System Type for Extract

To select the required System Type for extract, edit the **dbvtDSPCommon\_FullSystemType** view in SSMS and update the Where Clause to select the System Type that should be extracted.

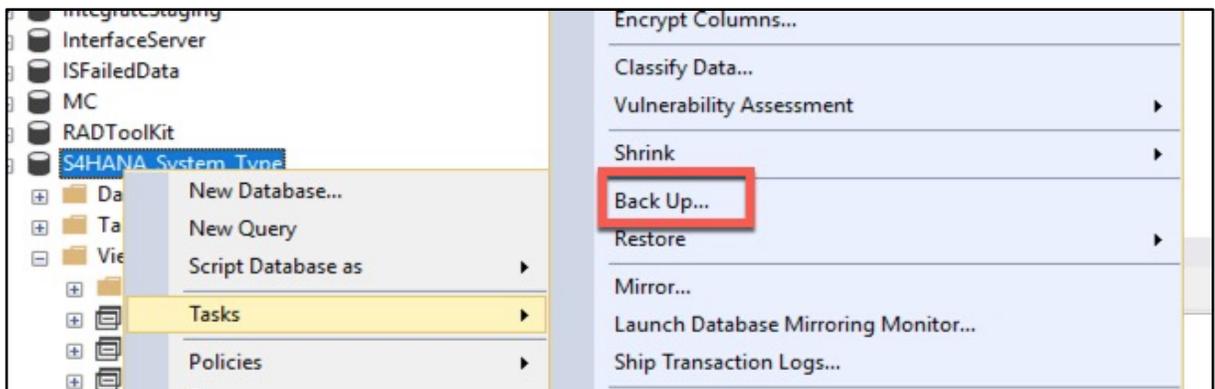


## Export Migration Content Pack System Type

To extract a System Type, run the Export\_SystemType stored procedure.



Once the stored procedure has been run, the various tables in the database should contain ONLY the records for the selected system type. The database can then be backed up and made available for installation on another Stewardship Tier instance.



## Install Migration Content Pack System Types

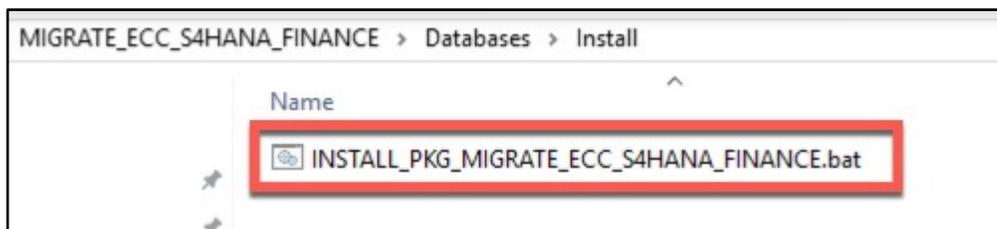
To install Migration Content Pack System Types:

1. Restore System Type Extract Database on target Stewardship Tier database instance using the saved Backup file.
2. Run the **Import\_SystemType** stored procedure to install the System Type.

## Install Migration Content Pack Data

To install the Migration Content Pack Data:

1. Navigate to the **Content Pack/Databases/Install** folder.
2. Run the .Bat file by double-clicking it.



## Appendix A: Advanced Data Migration Content Pack Template Stewardship Tier Application Tables Extracted

The Migration Content Pack DBVT template contains definition files that extract data from several Stewardship Tier applications. This appendix details the definition files used and the tables that are included within each definition.

### Application - Cransoft

**Definition:** Content.SourceDataSources

- Datasources

**Definition:** Content.TargetDataSources

### Application - Console

**Definition:** ConsoleElements extracts the following tables:

- ttObject
- ttProcessArea
- ztLoadCycle
- ztPlanningGroup
- 

**Definition:** WaveStructure extracts the following tables:

- ttWave
- ttWaveProcessArea

- ttWaveProcessAreaObject
- ttWaveMilestone
- ttWaveMilestonePlanningGroup
- ttWavePlanningGroup
- ttWaveLoadCycle
- ttWaveLoadCyclePlanningGroup

**Definition:** TargetDesign extracts the following tables:

- ttFieldGroup
- ttObjectFieldGroup
- ttTarget
- ttTarget\_Console
- ttSource
- ttTargetSource
- ttTargetLookupTable
- ttTargetComplexFieldRule
- ttTargetComplexRuleFieldMap
- ttTargetField
- ttEasyRule
- ttTargetRuleExecutionStage
- ttTargetSimpleFieldRule
- ttTargetSimpleRuleFieldMap

## Application - Map

**Definition:** FieldMapping extracts the following tables:

- ttObject
- ttObjectMappingGroup
- ttObjectMappingGroupFieldGroup
- ttObjectTableColumn
- ttTarget
- ttTargetField
- ttTargetFieldMapping
- ttTargetMappingGroup
- ttTargetMappingGroupFieldGroup
- ttTargetRelationship
- ttTargetRelationshipField
- ttTargetSource
- ttTargetSourceSchema
- ttValueMappingCheckTable
- ttValueMappingCheckTableCollation
- ttValueMappingCheckTableValue
- ttValueMappingSource

- ttValueMappingSourceStagingField
- ttValueMappingSourceStagingFieldCollation
- ttValueMappingSourceValue
- ztWaveServiceCheckTable

### Application - DSW

**Definition:** TargetTransform extracts the following tables:

- ttWaveProcessAreaObject
- ttWaveProcessAreaObjectTarget
- ttWaveProcessAreaObjectTargetExport
- ttWaveProcessAreaObjectTargetReport
- ttWaveProcessAreaObjectTargetReportRemediation
- ttWaveProcessAreaObjectTargetRule
- ttWaveProcessAreaObjectTargetSource
- ttWaveProcessAreaObjectTargetSourceReport
- ttWaveProcessAreaObjectTargetSourceRule

### Application - Cranport

**Definition:** AssemblePackages extracts the following tables:

- CPPackage
- CPPackageColumn
- CPSourceColumn
- CPTargetColumn

## Appendix B: DBVT MS SQL 2008 Compatibility Fix

If the Migration Content export fails with an error similar to below, it is because the version of DBVT installed relies upon MS SQL Server 2008 components.

```
Original.....: sdbSAP_ECC6
Resolved.....: sdbSAP_ECC6

Could not load file or assembly 'Microsoft.SqlServer.SqlClrProvider, Version=10.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91' or one of its dependencies. The system cannot find the file specified.
Error encountered during execution without verbose mode.
To enable verbose output, add -Verbose to the execution parameters.

Errored: 10/23/2019 3:12:36 PM
Duration: 00:00:00

ERROR: See 'sdbSAP_ECC6_ExportContent.log' for details.
ERROR: sdbSAP_ECC6 Export failed.
Press any key to continue . . . _
```

This issue can be fixed by performing the following steps:

1. In the root of the AdvancedDataMigrationContentPackagingUtility.zip file, copy the **VersionDatabase.exe.config**.
2. In folder ContentPack\Databases\Tools\FOR\_SMO\_10

- Remove any dlls that start with "Microsoft.SqlServer."
  - Replace file **VersionDatabase.exe.config** file (XML) with the file stored in the root of the AdvancedDataMigrationContentPackagingUtility.zip.
3. Edit the **VersionDatabase.exe.config** file, changing the newVersion number for the Smo and ConnectionInfo assemblies to the version of sql that is installed.

The following is an example of the items that need to be changed:

```

1 |<?xml version="1.0" encoding="utf-8" ?>
2 |<configuration>
3 |  <configSections>
4 |    <section name="databaseVersioning"
5 |      type="CranSoft.DatabaseVersioning.App.Configuration.DatabaseVersioningConfiguration,
6 |      VersionDatabase" />
7 |  </configSections>
8 |  <databaseVersioning connectors="~\..\..\Config">
9 |    <repositories>
10 |      <add path="~\..\..\Apps" />
11 |    </repositories>
12 |    <externalConfig typeName="CranSoft.Configuration.Integration.ConnectionInfoProvider,
13 |      CranSoft.Configuration" />
14 |  </databaseVersioning>
15 |  <appSettings>
16 |    <add key="CranSoftConfigPath" value="..\..\..\Web\Web.Config" />
17 |    <add key="UseCacheDataSourceSchema" value="false"/>
18 |  </appSettings>
19 |  <runtime>
20 |    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
21 |      <dependentAssembly>
22 |        <assemblyIdentity name="Microsoft.SqlServer.Smo"
23 |          publicKeyToken="89845dcd8080cc91" culture="neutral" />
24 |        <bindingRedirect oldVersion="10.0.0.0-14.0.0.0" newVersion="12.0.0.0" />
25 |        <publisherPolicy apply="no" />
26 |      </dependentAssembly>
27 |      <dependentAssembly>
28 |        <assemblyIdentity name="Microsoft.SqlServer.ConnectionInfo"
29 |          publicKeyToken="89845dcd8080cc91" culture="neutral" />
30 |        <bindingRedirect oldVersion="10.0.0.0-14.0.0.0" newVersion="12.0.0.0" />
31 |        <publisherPolicy apply="no" />
32 |      </dependentAssembly>
33 |    </assemblyBinding>
34 |  </runtime>
35 | </configuration>

```

The following is the MS SQL Server Version Matrix:

|      |      |                    |
|------|------|--------------------|
| 12.0 | 2014 | SQL Server 2014    |
| 13.0 | 2016 | SQL Server 2016    |
| 14.0 | 2017 | SQL Server 2017    |
| 15.0 | 2019 | SQL Server 2019 RC |

4. Rerun Export of Migration Content

### Appendix C: DBVT Extract of SQL Objects with Special Characters Fix

If the Migration Content export fails with an error similar to below, it is because the version of the DBVT installed does not support the extract of SQL objects that contain special characters that are not compatible with file names.

```
Server and database configured via external configuration.
Operations will be performed against the resolved database.

Original.....: dgSAP
Resolved.....: dgSAP

Scripting schema.
Could not find a part of the path 'C:\TABLE\NAME.sql'.
Error encountered during execution without verbose mode.
To enable verbose output, add -Verbose to the execution parameters.

Errored: 1/14/2020 4:12:09 PM
Duration: 00:00:01
```

This issue can be fixed by performing the following steps:

4. In the root of the AdvancedDataMigrationContentPackagingUtility.zip file, copy the **CranSoft.DatabaseVersioning.Library.dll**.
5. Replace file **CranSoft.DatabaseVersioning.Library.dll** in folder ContentPack\Databases\Tools\FOR\_SMO\_10 with this file.
6. Rerun Export of Migration Content.

### Appendix D: SQL View and Procedure Analysis SQL

The following is the SQL for the views that support the SQL analysis.

```
USE [zMig_Content]
```

```
GO
```

```
Create View [dbo].[xxxCrossDatabaseProceduresxxx] AS
```

```
SELECT sys.Procedures.name AS [procedure],
```

```

        sys.sql_expression_dependencies.referenced_database_name,
sys.sql_expression_dependencies.referenced_schema_name,

        sys.sql_expression_dependencies.referenced_entity_name
FROM    sys.Procedures INNER JOIN

        sys.schemas ON sys.Procedures.schema_id = sys.schemas.schema_id INNER JOIN

        sys.sql_expression_dependencies ON sys.Procedures.object_id =
sys.sql_expression_dependencies.referencing_id
WHERE   (sys.sql_expression_dependencies.referenced_database_name IS NOT NULL)
GO

```

```

CREATE View [dbo].[xxxCrossDatabaseViewsxxx] AS
SELECT   sys.views.name AS [view],

        sys.sql_expression_dependencies.referenced_database_name,
sys.sql_expression_dependencies.referenced_schema_name,

        sys.sql_expression_dependencies.referenced_entity_name
FROM    sys.views INNER JOIN

        sys.schemas ON sys.views.schema_id = sys.schemas.schema_id INNER JOIN

        sys.sql_expression_dependencies ON sys.views.object_id =
sys.sql_expression_dependencies.referencing_id
WHERE   (sys.sql_expression_dependencies.referenced_database_name IS NOT NULL)
GO

```

```

CREATE VIEW [dbo].[xxxDatabaseProcedureInternalNameConflictxxx]
AS
SELECT   SPECIFIC_CATALOG, SPECIFIC_NAME, ROUTINE_DEFINITION
FROM    INFORMATION_SCHEMA.ROUTINES

```

```
WHERE (dbo.boaCompareExternalInternalObjectNames(SPECIFIC_NAME,
ROUTINE_DEFINITION, N'P') = 0)
```

```
GO
```

```
CREATE VIEW [dbo].[xxxDatabaseViewInternalNameConflictxxx]
```

```
AS
```

```
SELECT TOP (100) PERCENT TABLE_CATALOG AS [Database], TABLE_NAME AS ViewName,
VIEW_DEFINITION
```

```
FROM INFORMATION_SCHEMA.VIEWS
```

```
WHERE (dbo.boaCompareExternalInternalObjectNames(TABLE_NAME, VIEW_DEFINITION,
N'V') = 0)
```

```
ORDER BY ViewName
```

```
GO
```

```
CREATE VIEW [dbo].[xxxMissingDependenciesxxx]
```

```
AS
```

```
SELECT TOP (100) PERCENT QUOTENAME(OBJECT_SCHEMA_NAME(referencing_id)) + '.' +
QUOTENAME(OBJECT_NAME(referencing_id)) AS [this sproc or VIEW...],
ISNULL(QUOTENAME(referenced_server_name) + '.', '')
```

```
+ ISNULL(QUOTENAME(referenced_database_name) + '.', '') +
ISNULL(QUOTENAME(referenced_schema_name) + '.', '') + QUOTENAME(referenced_entity_name)
AS [... depends ON this missing entity name]
```

```
FROM sys.sql_expression_dependencies
```

```
WHERE (is_ambiguous = 0) AND (OBJECT_ID(ISNULL(QUOTENAME(referenced_server_name) +
'.'. '')) + ISNULL(QUOTENAME(referenced_database_name) + '.', '') +
ISNULL(QUOTENAME(referenced_schema_name) + '.', ''))
```

```
+ QUOTENAME(referenced_entity_name)) IS NULL)
```

ORDER BY [this sproc or VIEW...], [... depends ON this missing entity name]

GO