

CI SaaS Migration Utility Guide

©2016 Morningstar. All Rights Reserved.

Document Version: 3
AccountView Version: 1.20
Custodial Integrator Version: V3.2.003 or greater

Document Issue Date: November 21, 2016

Technical Support: (866) 856-4951
Telephone: (781) 376-0801
Fax: (781) 376-8040
Web: byallaccounts.morningstar.com

Table of Contents

Migrating Translations from CI to AV for CI SaaS	1
Definition of CI SaaS	1
Related documentation	1
Overview of this guide	1
Overview.....	1
Install ExportUtility.....	1
About ExportUtility	1
Where to find it.....	1
Where to put it	1
Execute ExportUtility	1
How the ExportUtility operates.....	1
How to run the utility.....	2
To run a single instance	2
To run multiple instances	2
Status messages.....	2
Output files	3
Output file location	3
Export errors	3
Import the Translations into CI SaaS	4
How the import in AccountView operates.....	4
How to import the files.....	4
To review the status of the import.....	4
Appendix A: Account Translation File	5
Appendix B: Security Translation File.....	7

MIGRATING TRANSLATIONS FROM CI TO AV FOR CI SAAS

Definition of CI SaaS

AccountView (AV) has an enhanced export capability for firms that want to extract financial data and export it in files that are formatted for use by PortfolioCenter (PC) management system. That version of AccountView is referred to as CI SaaS, which stands for Software as a Service.

Related documentation

Refer to <http://www.byallaccounts.net/Manuals/Accountview/index.html> for other CI SaaS documentation.

Overview of this guide

This guide describes the steps required to migrate your security and account translations from CI to AccountView (CI SaaS). The audience for this manual is the Data Administrator who is responsible for running CI and loading the data into PortfolioCenter. As Data Administrator you should be familiar with working directly in the Windows file system.

Overview

The migration process involves doing the following steps:

1. Install a tool, ExportUtility, on your CI system that will be used to extract security and account translations from your CI instance(s),
2. Execute the new tool to write the translations to comma-separated values (.csv) files,
3. Import the translations into AccountView.

The following sections explain those steps in detail.

INSTALL EXPORTUTILITY

About ExportUtility

The ExportUtility tool extracts your account translations and security translations from your CI instance. It writes these translations to .csv files that may then be imported into AccountView for use in CI SaaS.

Where to find it

The ExportUtility is posted in <http://www.byallaccounts.net/CI/CISaasMigrationUtility.zip> and login is required to access it. The .zip file contains:

- This document.
- ExportUtility.jar – the java executable jar file that performs the extraction.
- runExportUtility.bat – the batch file that contains the command to execute ExportUtility.jar.

Where to put it

Place the jar and bat files into the same folder in which CI is installed. This folder is typically C:\Program Files (x86)\Custodial Integrator.

EXECUTE EXPORTUTILITY

How the ExportUtility operates

The ExportUtility runs against the CI database and it references the same files that control CI, namely runCI.bat and CI.ini. Those files provide parameter settings such as the name of the SQL Server and the database to access (typically named BaaWpAci). The parameters obtained from

these files are listed in the following table. If neither file is found or neither provides the required parameters then default values are used.

Parameter	Description	Default value if not found
dbname	Name of SQL Server database used by CI	BaaWpAci
sqlhost	Name of machine hosting SQL Server	localhost
sqllogin	User's SQL Server login name	windows authentication
sqlpw	User's SQL Server password	windows authentication
ciinidir	Folder in which CI.ini file resides	folder in which runExportUtility.bat was run

How to run the utility

The user running the tool must have read access to the runCI.bat file and the CI.ini file that are in the same folder in which the tool is executed.

To run a single instance

If you run a single CI instance on your system then the tool is ready to be executed without any changes. To run the tool, execute the batch file runExportUtility.bat. This script executes a java command and requires no interactive input so can be run directly or from a Command Prompt window (cmd.exe). The script is simply the following command, which can be configured to run against any CI database:

```
java -Djava.library.path=.\extlib -cp ExportUtility.jar;extlib\sqljdbc4.jar;extlib\gnu-regexp.jar com.baa.aci.ExportUtility runcipath:". \runCI.bat"
```

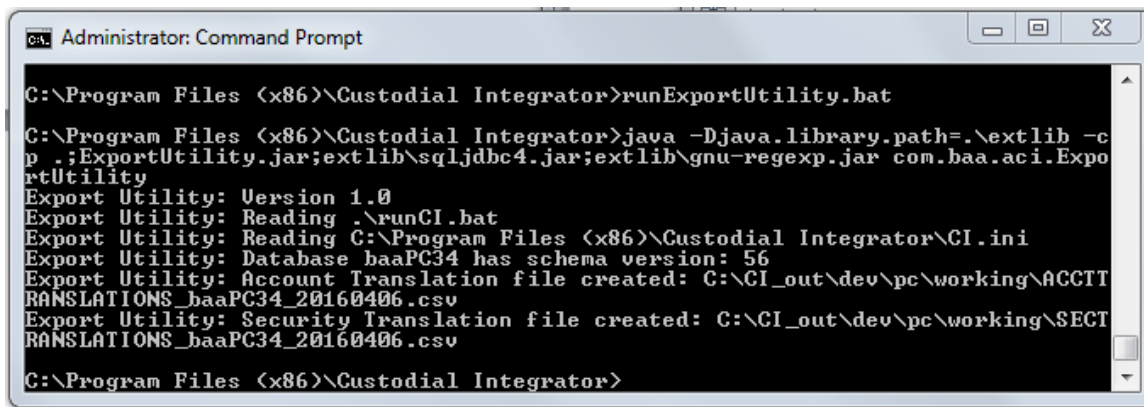
To run multiple instances

If you run multiple instances of CI, edit the command line to show the name of the customized .bat file for each instance as needed. For example, if the .bat is runCIforCustomer2.bat, the command line would be:

```
java -Djava.library.path=.\extlib -cp ExportUtility.jar;extlib\sqljdbc4.jar;extlib\gnu-regexp.jar com.baa.aci.ExportUtility runcipath:"C:\Program Files\Custodial Integrator\runCIforCustomer2.bat"
```

Status messages

When run from a Command Prompt window ExportUtility displays a few progress statements and a final status of the process. For example:



Output files

The tool generates two .csv files; within the files each value is enclosed in double-quotes.

- an account translations file (ACCTTRANSLATIONS_<dbname>_<date>.csv)
- a security translations file (SECTRANSLATIONS_<dbname>_<date>.csv)

In both filenames dbname is the name of the CI database. See the Appendices for more detailed descriptions of the file contents.

Output file location

These files are output to the CI working folder specified in the CI database. Any existing file of the same name in the folder is overwritten.

Note: The tool requires write-access to this folder.

Similar to CI, a log file is created during execution. This file can typically be found in the log folder specified in the CI database. If an error occurs before the pathname of the folder can be obtained from the database then the log file can be found in the user’s temporary working folder which is typically:

\Users\<username>\AppData\Local\Temp\CITemp\log\baaCi####.log

Export errors

The following errors are possible:

Description	Resolution
Unable to process command line parameters	Review the parameters in the runExportUtility.bat file.
Unable to read CI.ini file	Verify that user has access to this file.
<value> is not a valid parameter value.	Review the parameters in runExportUtility.bat file, runCI.bat file and CI.ini file
The configured CI working folder does not exist or could not be accessed: <folder name>. You must have write access to this folder.	Startup CI and verify the working folder is defined in the Configuration tab. Verify that you have write access to the folder.
A database error occurred while creating Security Translation file: <filename>	Verify the database name that is being accessed. Startup CI and verify that it can display translations in the Security Translations tab.
An error occurred while writing to Security Translation file: <filename>	Verify a file of the same name is not opened by another program such as Excel.

A database error occurred while creating Account Translation file:	Verify the database name that is being accessed. Startup CI and verify that it can display translations in the Account Translations tab.
An error occurred while writing to Account Translation file:	Verify a file of the same name is not opened by another program such as Excel.
Please provide your SQL Server password using sqlpw=yourpw in the CI.ini file.	A sqllogin was provided but no sqlpw. Review the parameters in the runCI.bat and CI.ini file.
Invalid database version. Cannot run with this database. Current version for <dbname> is <version>. Required version is 48 or greater.	Verify that the CI version is V3.2.003 or greater. Verify the database name that is being accessed. Contact ByAllAccounts.

IMPORT THE TRANSLATIONS INTO CI SAAS

How the import in AccountView operates

AccountView provides an Import function in each of the translations views: Account Translations and Security Translations. This function attempts to create translations from a .csv file of translations that you specify. For each row in the .csv file the import may create a new translation, override an existing translation, or ignore the translation.

How to import the files

To import the files, run the Import function in AccountView in both the Account Translation and Security Translation views then specify the .csv file to upload. These steps describe how to import the account translations. Use the same procedure to import the security translations in the Security Translations view.

1. In AccountView, go to the Account Translations View.
2. Click **Import**.
3. Browse to the location of the account translations .csv file you created with the ExportUtility and select the file.
4. Click **Upload** to start the import process.
The 'Importing data' link will change to 'Import ready' when the import is completed.
5. In AccountView, view the new or changed translations in the Account Translations table.

To review the status of the import

Once the import is complete the resulting status file can be downloaded to your system for review.

1. Click the 'Import ready' link to view the 'Import Results'.
2. Click the downloads icon next to the latest zip file in the 'Import Results' table to begin the browsers download process.
3. Open the .zip file.

The .zip file that AccountView produces includes a readme file that summarizes the number of translations that were processed, were successful, and that failed. The .zip also contains your original .csv file with two additional columns that provide details about how each row was handled during the import. The following table lists the possible treatments for each row.

Load Status	Load Details
Succeeded	New translation created from import record.
Succeeded	Overwrote an existing translation with import record.

Succeeded	Import record ignored because all values matched an existing translation.
Succeeded	Import record ignored because all values matched the system default translation.
Succeeded	New translation created from import record. Transaction start date ignored: mm/dd/yyyy.
Failed	Failed to import because account not found.
Failed	Failed to import because institution not found.
Failed	Failed to import because a required field has no data: <name of field>.
Failed	Failed to import because of invalid data in field <name of field>.
Failed	Failed to import because of system error.
Failed	Failed to import because no translation data in fields:<name of field1>,<name of field2>...
Failed	Failed to import because translation needs to be account-specific.

APPENDIX A: ACCOUNT TRANSLATION FILE

The account translation file always has a header row. Rows are ordered by the ACCOUNT_ID column, ascending.

Column Header	Description	Requires a Value
ACCOUNT_ID	Account’s unique database ID.	√
OUTPUT_ACCOUNT_IDENTIFIER	User-assigned account identifier.	
ACCOUNT_NAME	Name of account in AccountView.	
ACCOUNT_NUMBER	Account number from the institution.	
INSTITUTION_ID	Unique database ID of the institution.	
INSTITUTION_NAME	Name of the institution.	
TRANSACTION_START_DATE	Earliest allowed execution date for transactions to be included in next CI Import/Export run.	
CURRENCY	Account’s currency type for CI TLM/TOD. Not used for CI PortfolioCenter.	
IMPORT_ENABLED	Specifies whether to include account in next export process: True or False.	
GLOBAL_EXCLUDE_POSITIONS_IF_STALE	Specifies for all accounts whether to include stale positions in next export process: True or False.	
EXPORT_POSITIONS_IF_STALE_OVERRIDE	Specifies whether this account should override the “GLOBAL_EXCLUDE_POSTITIONS_IF_STALE” setting: empty = use global exclude setting, True = override setting and export stale positions.	
MM_SYMBOL	Overriding symbol for MONEY MARKET BALANCE securities.	
MM_SYMBOL_TYPE	CI Universal’s symbol type for MM symbol.	
SWEEP_SYMBOL	CI Universal’s symbol for sweep transactions.	

SWEEP_SYMBOL_TYPE	CI Universal's symbol type for sweep transactions.	
SWEEP_SECURITY_TYPE	CI Universal's security type for sweep transactions.	

APPENDIX B: SECURITY TRANSLATION FILE

The security translations file always has a header row. Rows are ordered by the ID column, ascending.

Column Header	Description	Requires a Value
ID	CI translation identifier from which record was generated, not necessarily unique.	√
INPUT_TICKER	Input security ticker.	*
INPUT_CUSIP	Input security CUSIP.	*
INPUT_NAME	Input security name.	*
INPUT_FI_ID	Unique database ID for the institution, to be used with INPUT_NAME.	**
INPUT_ACCOUNT_ID	Unique database ID for the account to be used with INPUT_NAME for account-specific translation.	
SECURITY_REFERENCE_NAME	Informational name for symbol-based translation.	
OUTPUT_SECURITY_TYPE	Output PortfolioCenter security type.	√
OUTPUT_SYMBOL	Output security symbol.	√
OUTPUT_SYMBOL_TYPE	Output symbol type (CI Universal only).	
CREATION_DATE	Creation date of the translation.	
LAST_MODIFIED_DATE	Last modification date of the translation.	

*One and only one of these fields must contain data.

** Typically valued if INPUT_NAME is valued, but not always.