MORNINGSTAR Wealth

Custodial Integrator Automation Guide

Company Confidential

Custodial Integrator Product Version:	V3.19
Document Version:	17
Document Issue Date:	April 10, 2025

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

Contents

Table of Contents

USING THIS GUIDE

Audience

This guide describes the capabilities of Custodial Integrator Automation (CI Automation), which enables Custodial Integrator (CI) to be run from the command line instead of from the user interface. The manual is written for the IT Administrator who is responsible for configuring and maintaining CI. This document describes how CI Automation works, how it compares to CI, and what options are available to configure automation behavior. It is helpful if the reader is familiar with CI and how it is implemented in the local environment.

Related Documentation

This document describes automation options. Use it in conjunction with your other CI documentation. Refer to:

- *Custodial Integrator User Guide* to fully understand the options available through the user interface.
- *Custodial Integrator Installation Guide* for information about runtime parameters used to configure CI behavior.

Guides for the variant of Custodial Integrator you are using are available at http://www.byallaccounts.net/manuals/Content/CI/CustodialIntegrator.htm.

INTRODUCTION TO CI AND CI AUTOMATION

Custodial Integrator (CI) is an application that transfers data from the ByAllAccounts repository to your local environment and formats that data into files specific to your target system. CI is typically run as a user interface (UI) application that presents to the user the various steps for the data transfer and data checking process, allowing that user to interact with it and resolve exceptions.

As an alternative to being run as a user interface application, CI can be run from the command line using CI Automation. Using CI Automation, CI can be scripted to run automatically at various times of the day to perform its normal processing and produce its output files. Because the user does not interact with the UI, careful setup and monitoring is required to ensure that CI Automation produces the expected output.

CI AUTOMATION – IS IT APPROPRIATE FOR YOU?

Criteria for Running CI Automation

CI Automation is recommended only when **all** of the following items are true for you:

- There is a large volume of accounts to be processed.
- You want to transfer your account data to your local environment as early as possible in the day so that data is ready for your staff when they arrive at work rather than requiring them to perform activities that cause the accounts to be transferred.
- There is a low incidence of processing exceptions or you have the ability to programmatically handle exceptions using information that CI outputs to primary or secondary files. A processing exception is any data situation that requires additional information to be provided in order to properly prepare the data for your target system. An example of a processing exception occurs when a security has no unique identifier such as a Ticker, CUSIP, SEDOL, or ISIN. This guide describes the options CI Automation provides for you to handle exceptions; however, it is likely

that you will still need to handle some types of exceptions in your target system (or using an intermediate processor) in order for CI Automation to work well for you.

- IT staff is available to configure parameter files, batch scripts, and scheduled tasks. This same staff will also need to respond to and troubleshoot any CI Automation errors that might occur when there are changes in your local environment that affect CI, such as Java upgrades or other system changes.

If all of these items do not describe your situation then CI Automation may not be suitable for you.

Special Considerations

There are several factors to consider if you decide to automate CI, some of which may not apply to your environment.

- If you plan to run CI Automation multiple times per day then you will need to determine the best times to run CI Automation in relation to the time that data is available from your financial institutions. ByAllAccounts can work with you to determine the typical time each day when a majority of your accounts will have been updated with prior day close data. This time establishes when you want your first run of CI Automation that day. You will likely need to run CI Automation at least one more time to collect data for accounts at institutions that report data later in the day (such as 8:30 AM Eastern or later). If you have a high volume of accounts you might want to introduce a third run in between the first and second runs of CI Automation. Again, the timing of this additional run should be aligned with the time your financial institutions report data.
- If you use CI with a target system such as PortfolioCenter or Morningstar Office that is expecting to receive only one set of files (position, transaction, etc.) per day and you want to run CI Automation to produce multiple file sets daily (to accelerate delivery) then you must determine how you will handle multiple file sets in your target system.
- If you install and run multiple instances of CI each with a separate CI database, then you must review "Appendix F: Multiple CI Instances and Databases" for how to manage them using CI Automation.

COMPARISON OF CI AND CI AUTOMATION

Similarities and Differences

When running CI manually, a user sees information and makes decisions about whether or not to move forward with a run. The user sees messages and can intervene or proceed. Using CI Automation, the behavior of CI is controlled by parameter settings instead of through direct user interaction. These parameter settings pre-determine the choices for how to handle common exceptions, and enable CI Automation to continue processing through them. For example, parameter settings can control whether CI Automation will update stale accounts and whether it will translate non-validated accounts.

CI can be run more than one time per day (taking delivery of multiple CI file sets) in order to obtain data as it becomes ready in the ByAllAccounts repository rather than waiting for all data to be available before running CI. Likewise, CI Automation can be scripted to run automatically at various times of the day to perform its normal processing and produce output files.

For either method, exceptions detected are written to log files. CI Automation generates an additional log file that provides a history of the actions performed by each run of CI Automation.

Overall Processing Flow for CI and CI Automation

Note: CI and CI Automation have the same general processing flow, but there are some key differences in how they are achieved. You need to understand the differences so you can configure CI Automation to get the behavior you want.

The CI user interface shows four major steps for moving data through CI to its output files. The first step, Setup, has settings that are stored persistently in the CI database including import/export defaults. Before using CI Automation for the first time, the Setup options must be set through the CI user interface. After that, each CI Automation run reads those settings from the database.

After it has been configured, CI Automation can follow the same steps as CI, but without requiring any user intervention to move from one step to the next. The options that a user would set in the user interface during an individual run of CI are managed by parameters for CI Automation. The *autoRun* parameter, which invokes CI Automation, can be set to run up to and including the Import, Export, and Accept steps.

The following items compare how CI and CI Automation behave through all of the steps.

1. **Setup**
   Use the Setup step in the user interface for both CI and CI Automation. Settings such as CI login and password, output folder, and import/export defaults, as well as account translations and security translations are always set through the user interface and stored in the CI database.

   Note: This manual setup is required before running CI Automation for the first time. Any changes made to Setup in the user interface affect both CI and CI Automation.

2. **Import data**
   During the Import step, both CI and CI Automation import financial data for the accounts that have been translated. There are two requirements for Import to take place in both CI and CI Automation:

   - There is at least one type of financial data (securities, positions, transactions, etc.) marked for import in the Setup step.
   - At least one valid account translation has been defined.

   When either of the requirements is not met, the Import button in the CI main dialog is disabled. In CI Automation, the application will stop before the Import step and an error message will be written to the AutoRunHistory.log file.

   Translated and Untranslated accounts: Both CI and CI Automation will import financial data only for accounts that have been translated; accounts that have not been translated are skipped during the Import process. For both CI and CI Automation, the user can manually create account translations through the *Account Translation* dialog during the initial Setup step. CI Automation however offers an additional capability to automatically create account translations. A parameter can be set to automatically translate untranslated accounts before importing data; as described on page 23 when the *translateUntranslatedAccounts* parameter is set, CI Automation will map each untranslated account to the default account identifier for your variant of CI. By using this parameter you can ensure that every day new accounts will be automatically translated and their data imported without the need to first open the CI user interface and manually create translations for the new accounts. Refer to pages 20 and 23 for the parameters that can be set to tune the automatic account translation to your needs.

   Stale accounts: An account is considered stale in the BAA system when it does not reflect the latest data from the custodian or the account data it got from the custodian account was not for the prior business day. In CI, stale accounts are marked in bold in the *Account Translation* dialog; the user can see details of these accounts in the *Account Update Status* dialog. BAA will continue to try to get the data from the custodian until a set time of day. If the user believes the custodian has up-to-date data that BAA did not retrieve, the user can run a manual account update for some or all stale accounts (account update on demand). The goal is to have the stale accounts updated before starting the CI import, so CI will

receive the latest data for these accounts and export it to the output files. When running CI, the user can request to update one or more accounts in the "Account Update Status" dialog. In CI Automation, account update is requested through the *updateStaleAccounts* parameter (page 25). When this parameter is set, CI Automation will update all the stale accounts before running Import. Note that account update may take a considerable length of time, depending on the number and size of the stale accounts. Refer to "Appendix E: Handling Stale Accounts" for strategies than can be used to efficiently manage update of stale accounts when using CI Automation.

Invalid account translations: When running CI manually, if one or more account translations is invalid (for example the account was deleted since the last import), the Import button is disabled and no data can be imported until the user resolves all invalid account translations. CI Automation is different. In CI Automation invalid account translations are skipped during import, but data is still imported for accounts with a valid translation. The AutoRunHistory.log file will contain a message informing that invalid account translations were found and have been disabled for import. Invalid account translations are shown in the CI user interface in bold blue in the *Account Translation* dialog and can only be deleted manually through the user interface.

3. **Export to files**
   After data has been successfully imported to CI, the CI user can click the Export button to generate files to the output folder. In CI Automation, the Export option can be preselected using the *autoRun:EXPORT* parameter. Note that Export works only on translated accounts.

   Managing untranslated (undefined) securities: When importing financial data CI (and CI Automation) verifies that each position, transaction, and lot references a valid security. If the security cannot be identified because the symbol is missing or invalid, the security is marked as "undefined" or "untranslated". The presence of one or more undefined securities in the imported data may affect the Export step depending on the CI variant you are using. Essentially, for some variants, both CI and CI Automation will export data even if there are untranslated securities by inserting a dummy security symbol. For other CI variants, CI and CI Automation cannot export any data if there are untranslated securities. The variant-specific rules are the same in CI and CI Automation. For details about how each variant treats untranslated securities, refer to "How CI Variants Handle Untranslated Securities", page 31.

   If your CI variant does not allow Export when there are undefined securities, the Export button in the main CI screen stays disabled while a warning message is displayed with instructions to manually translate the undefined securities. In CI Automation the application will stop and exit after the Import step. A message in the AutoRunHistory.log file will indicate that undefined securities have been found and need to be manually translated before Export can be run. Unlike account translations, security translations cannot be automatically generated by CI Automation. The user has to manually translate the undefined securities in the CI *Security Translation* dialog and then run CI Automation again for Export to take place.

   Stale accounts and positions export: If you choose not to update stale accounts before import, know that by default CI and CI Automation will not export positions from stale accounts. You can override this default in CI Setup by unchecking the "Exclude positions for stale accounts" advanced Setup option, thereby including them. Making this change in Setup affects both CI and CI Automation. In addition, for both CI and CI Automation the *outputStaleAccounts* parameter (page 20) can be set to output a list of stale accounts that you can then manually update.

4. **Accept exported data**
   Note: Because the Accept step writes state information about the most recent download to

the database, it is always recommended that the user first verify that CI properly produced the output files before accepting the exported data, whether running CI or CI Automation.

When running CI manually, the user clicks Accept so state information about the most recent download is written to the database. In CI Automation, the Accept option can be preselected using the *autoRun:ACCEPT* parameter. When it is, CI Automation performs the whole cycle of Import, Export, and Accept. Note that to allow time to verify CI Automation output before Accept, set parameters *acceptFile* and *acceptFileTimeout* in your runCIAuto.bat file. Refer to "Appendix A: CI Automation Parameters".

Export of Secondary Files with CI and CI Automation

CI and CI Automation export the primary files that are configured for export in Setup in the main CI Configuration screen. These files are output during the Export step.

In addition, CI and CI Automation can optionally export secondary files. Secondary files contain configuration and other auxiliary data that is not fed into the target system but that can be helpful to the CI user.

Secondary files are not exported by default. For both CI and CI Automation, export of each secondary file is controlled by its associated output parameter as described in "Appendix A: CI Automation Parameters".

The following secondary files can be exported:

- Account Translations
- Missing Prices
- Security Translations
- Stale Accounts
- Untranslated Accounts
- Untranslated Securities

When Secondary Files are Available in CI Automation

CI exports all requested files, primary and secondary, during the Export step only and will not automatically export the requested secondary files if you exit CI before running Export. CI Automation, on the other hand, behaves differently.

- **Import-independent secondary files are available without running Import:**
  With CI Automation, you can optionally generate import-independent secondary files without running the sometimes time-consuming Import step. By setting parameters to request output of Untranslated Accounts, Account Translations, Security Translations, then running CI Automaton in a pre-import mode, those files will be exported before CI Automation exits. For an example of when the capability of exporting secondary files without performing Export or Import may come in handy in CI Automaton, refer to Appendix G: "Example of Secondary File Output".
- **Most secondary files can be produced by only running through Import:**
  If you set parameters to request output of secondary files such as Untranslated Accounts, Account Translations, Security Translations, Untranslated Securities, and Stale Accounts, you can run CI Automation through the Import step and those requested secondary files will be exported after the Import step completes and before CI Automation exits.
- **Export of all secondary files:**
  If you set parameters to request output of secondary files (Untranslated Accounts, Account Translations, Security Translations, Untranslated Securities, Stale Accounts, Missing Prices) and Automation is set to run up to the Export or Accept step, all requested secondary files will be

exported during the Export step after the primary files are exported. In this case the behavior is the same in CI and CI Automation.

Switching Between CI and CI Automation

Although you may choose to run CI in an automated way on most days, you will likely need to run CI manually (through the user interface) periodically to change settings or check on some information. Note that:

- You must not run both CI and CI Automation at the same time against the same database.
- There are certain CI parameters that should be configured identically for manual mode and automated mode, such as the account identifier scheme. It is recommended that those parameters be set in the CI.ini file. Refer to " Set Up and Run CI Automation", page 7 for an explanation of where to set parameters.

For additional details and caveats, refer to "If You Run CI Automation and Need to Run CI", page 10.

SET UP AND RUN CI AUTOMATION

The CI interface is invoked using the runCI.bat file. To run CI Automation, it is recommended that you create a customized version of the runCI.bat file and call it runCIAuto.bat. The basis of this recommendation lies in how runtime parameters are specified for CI and CI Automation.

Configuration Options for CI Automation

CI has a number of configuration options and the way you set up CI largely depends on your own implementation and what information you want CI to process. Those options are described in your *Custodial Integrator User Guide.*

This document describes options for CI Automation. Please use this guide in conjunction with your *Custodial Integrator User Guide* to fully understand the options available through the CI user interface. Refer to your *Custodial Integrator Installation Guide* for information about additional CI runtime parameters.

As described in "Appendix A: CI Automation Parameters", parameters specific to automation are available to instruct CI Automation to perform actions that would normally be performed by a user such as establishing a new account translation (mapping). For example, parameters can be set to control whether CI Automation will:

- Translate untranslated accounts
- Update stale accounts
- Translate non-validated accounts along with validated accounts
- Import tax lots
- Export only those accounts that are ready and not yet exported today–if you run CI more than once per day
- Produce secondary output files in addition (or independently of) primary output files.
- Coordinate acceptance of files delivered from CI

Notes:

- If you support multiple CI instances/databases, refer to "Appendix F: Multiple CI Instances and Databases".
- If you use custom translations or filters for positions and transactions refer to "Using Custom Filter and Translation Files" on page 33.

Manual and Automated Configurations for CI Automation

The Setup configurations must be set manually in the CI user interface, and they are stored persistently in the CI database. Other configurations can be controlled using parameters in the CI.ini file or in the runCI.bat (or a variant of it).

- **Run the CI user interface to set items in the CI database.** Every time you run CI Automation, it will import and export data according to the Setup configurations set in the CI user interface. You can run CI manually to change the settings in the Setup step. The options available depend on the variant of CI you are running. They include:
  - import/export defaults
  - account translations (mappings) and security translations
- **Use the CI.ini file and runCI.bat file (or a variant of it) to set parameters.** As described in "Appendix A: CI Automation Parameters", you can control many configuration options when automating CI. For example, parameters can be set to have CI Automation create translations for untranslated accounts or produce additional output files to fully document complex exceptions such as untranslated securities. For information setting the parameters, refer to "Set Up and Run CI Automation", page 7.

Overview of Steps to Set Up CI Automation

1. Run the CI user interface and adjust the Setup options which are then saved in the database. Configurations set in the Setup section of the user interface are persistent in the database and used every time CI or CI Automation is run. The options available depend on the variant of CI you are running. They include:
   - CI access including login, password, and working folder
   - Output configuration including file output folder
   - Data to import and export, such as transaction and positions, to control which primary output files are generated
   - Account translations (mappings) and security translations

   Refer to your *Custodial Integrator User Guide* for information about the other options available for your variant of CI.
2. Set CI parameters.
   a) In the CI.ini file, put the standard parameter settings that you want to use every time CI runs, whether as CI or as CI Automation. Some CI parameters should be configured identically for manual mode and automated mode, such as the account identifier scheme (`defaultAccountIdentifier`) or maximum length for account identifier (`AccountIdentifierMaxLen`). Set those parameters in the CI.ini file. The parameters that work for both modes are described in the *Custodial Integrator Installation Guide* for your variant of CI. The installation guides can be found at http://www.byallaccounts.net/manuals/Content/CI/CustodialIntegrator.htm.

   b) Copy the runCI.bat file and name it runCIAuto.bat. Put parameters specific to automation in the runCIAuto.bat file. Those parameters are described in this guide in "Appendix A: CI Automation Parameters".

   For more information about the parameter files, refer to "Setting CI Parameters", page 9.
3. Run CI manually through the Export step, and verify that CI has write permissions to the output folders and that you get the output you expect. (You do not have to complete the Accept step.)
4. Run CI Automation and verify the output. Refer to "Monitor CI Automation on an Ongoing Basis", page 10.

5. After you get the basic CI Automation working, review "Special Considerations", page 3 to determine if there are additional factors to consider implementing such as timing of automation runs, whether you need to manage single file sets, and if you need to implement multiple instances of CI.

Setting CI Parameters

By default there are two files in which you can adjust parameter settings to customize how Custodial Integrator (CI) functions:

- **runCI.bat** – the file that invokes CI with the user interface.
- **CI.ini** – the initialization file for CI.

Note: Parameter values in the runCI.bat file take precedence over those in the CI.ini file.

Although parameters that customize how CI runs can be placed in either the runCI.bat file (java command line) or in the CI.ini file, this guide recommends that you put the general parameters in the CI.ini file and create a third file called **runCIAuto.bat** and put the parameters specific to automation in that file.

- **runCIAuto.bat** – the file you create to invoke CI Automation, so that CI can run unattended. In this file include parameters related to automation, including the autoRun parameter. For information about automation parameters, refer to "Appendix A: CI Automation Parameters". For more information about the runCIAuto.bat file, refer to the next section, "Using a runCIAuto.bat File".

Using a runCIAuto.bat File

Even after you set up CI to run in a scripted mode, you will still need to access the user interface occasionally to verify or change a setting. Therefore it is recommended that you have separate .bat files for running CI and running CI Automation. Although you want CI to behave the same either way you run it, there are some parameters that apply only to the automated mode and those should be set in the runCIAuto.bat file.

As explained in "Setting CI Parameters" on page 9, it is recommended that you put into the CI.ini file all of the standard parameter settings that you want to use whenever you run CI, then specify all automation parameters in your custom runCIAuto.bat file. Using this method, you can then use the same CI.ini for running CI and CI Automation, and have the parameters that apply to only CI Automation in runCIAuto.bat.

Although the .bat file you start with may be slightly different, this example shows one that is modified to run CI Automation. Autorun specific parameters are shown in bold:

```
java -cp CI.jar;extlib\sqljdbc.jar;extlib\gnu-regexp.jar com.baa.aci.Cache
tool:PC autoRun:ACCEPT translateUntranslatedAccounts:y updateStaleAccounts:y
outputSecurityTranslations:y
```

Note: Be certain to construct the java command starting with the command defined in your runCI.bat (first line in the example). Your java command may be different than the one used in this example.

In this example, CI Automation will translate all untranslated accounts and update all stale accounts before importing. It will also generate a secondary output file that lists the manual security translations.

Note: The parameter outputSecuritTranslations is not specific to automation. However it is likely to be used more in CI Automation because there is no other way to look at the security translations. The same is true for the other output parameters that control export of secondary files.

If You Run CI Automation and Need to Run CI Manually

If you have been running CI Automation and want to run CI manually (via the user interface), follow these steps.

Note: These instructions assume that there is a separate .bat file for automation as recommended, and it is called runCIAuto.bat.

1. Compare your runCIAuto.bat file and your runCI.bat file. Ensure that any of the CI parameters described in your *Custodial Integrator User Guide* are configured identically for CI (manual mode) and CI Automation (automated mode). Although it is recommended that those CI parameters be set in the CI.ini file, they may have been set in the automation runCIAuto.bat file. If they are, make sure they are set identically in the runCI.bat file. Examples of CI parameters are *sqlhost*, *dbname*, and *defaultAccountIdentifier*.

   Note: Do not set CI Automation parameters (which are described in this guide) to run CI. Refer to "Set Up and Run CI Automation", page 7 for an explanation of where to set parameters.

2. Ensure that CI Automation is not running and will not start while you are running CI manually.
   WARNING: You must not run CI and CI Automation at the same time against the same database. When CI runs (in either mode) it accesses the database and puts state information into it. If you want to switch from CI Automation to running CI manually, stop the automated run first, or allow it to finish before running CI manually.

3. Start CI using runCI.bat.

MONITOR CI AUTOMATION ON AN ONGOING BASIS

How to Verify Results

This section describes methods you can use to monitor CI Automation and to determine whether it is producing the output you expect and whether there are exceptions that you should manage. In general:

- Examine the primary output files.
- Examine the secondary output files.
- Examine the CI log file and CI Automation history log file for exception handling.

Details about CI Automation Behavior

The following list describes the behavior of CI Automation.

- Output files are always overwritten. There is no option to prevent output files from being overwritten.
- Invalid account translations are disabled. This behavior allows the automation run to continue as if no invalid translations existed; the output files will contain the data for only the valid account translations.
- Fatal errors stop the automated execution.
- Non-fatal errors do not stop automated execution.

Primary Output Files

As part of its normal processing, CI outputs primary data files that are intended to transfer data to another system. The primary output files that can be produced depend on the variant of CI that is being run, as well as the configurations set in CI. The primary output files typically include transactions and positions. Depending on the variant and settings, they could include:

- Accounts
- Securities
- Prices
- Initial positions
- Position Lots

Examine the Primary Output Files

Look in the output folder identified in Setup in the CI user interface and ensure that CI Automation has produced the files you expect, and that they have the content you expect. In most cases, you would expect to have a position file. If so, check that it has one or more positions and that the data is as expected. Likewise, if you have a transaction file, verify that it has one or more transactions and that the data is as expected.

If Primary Output Files are Missing

If CI Automation does not produce any primary output files, check the Setup settings in CI. Those settings control which primary output files are produced by both CI and CI Automation.

WARNING: You must not run CI and CI Automation at the same time against the same database. When CI runs (in either mode) it accesses the database and puts state information into it. If you want to switch from CI Automation to running CI manually, stop the automated run first, or allow it to finish before running CI manually.

1. Run CI and check the Setup options.
2. In the Configuration tab, verify the path for the output folder.
3. Ensure that Import/Export Defaults are set. Those settings control which primary output files are produced by CI or CI Automation. The options available are different for each variant of CI.
4. Verify that CI has write permissions to the output folder set on the Configuration tab.
5. Ensure that the *autoRun* option in your automation script (runCIAuto.bat) is set to either Export or Accept.
6. Check the CI logs for errors or information messages that may explain why any primary output file did not get generated.
7. If no primary data was exported, check for the presence of the Undefined Securities file. If the file exists and is not empty, that may explain why Export was not performed.
8. Run CI Automation again.

Secondary Output Files

CI Automation can be configured using parameters to generate secondary output files, which can provide additional information about the data produced.

They include:

- Account Translations
- Missing Prices
- Security Translations
- Stale Accounts
- Untranslated Accounts
- Untranslated Securities

For detailed descriptions of Secondary Output files refer to "Appendix B: Examples of Secondary Output Files". For the parameters used to output these files, refer to "Appendix A: CI Automation Parameters".

Examine Secondary Output Files

Examining secondary output files can be an important part of monitoring the behavior of CI Automation. CI Automation generates secondary output files, such as security translations, if parameters are set to generate them.

Look in the output folder identified in Setup in the CI user interface and examine the content of the secondary files. Determine how to address any issues. For example, if there are untranslated securities you can set up the translations in CI manually then re-run CI Automation.

In general, look for unexpected values in output fields and for empty fields that are expected to contain values. Also check the log files to see if there was any error.

Log Files

Look for error information and instructions in the log and history log files.

CI and CI Automation write exceptions to log files. When CI Automation runs, exceptions are written to two different types of log files. By default the log files are found in the log subfolder of your CI working folder.

- **baaCi<XXXXXX>.log** files – These are the normal log files produced when CI runs.
- **AutoRunHistory.log** file – This is the automation log file. CI Automation does not produce status messages that are visible to a user. Instead, it writes messages to this log file. Exception messages are appended to this file for each run of CI Automation. If you delete or rename the file, the next run of CI Automation will create a new one.

Examine the AutoRunHistory.log file

The AutoRunHistory.log contains a running history of each CI startup, Account Update, Import step, Export step, Accept step, and CI exit. (Import, Export, and Accept are run if they are set in the *autoRun* parameter.) Some examples of successful runs that show in the AutoRunHistory log are shown in "Appendix C: AutoRunHistory.log File Examples".

In the AutoRunHistory.log file, look for fatal errors that stop execution, as well as non-fatal errors that do not stop execution.

Fatal Errors that Stop Execution

In the AutoRunHistory.log file, look for fatal errors that stop execution. Fatal errors are logged with the "Requested steps incomplete" message. For example:

```
01-05 17:25:47.194: main: Custodial Integrator V3.1.008 started with database
'BaaWpAci' on SQL host 'localhost'.
01-05 17:25:47.210: main: Error: An error occurred during auto execution. Info:
No import options selected.
01-05 17:25:47.210: main: Requested steps incomplete.
01-05 17:25:47.210: main: CI exiting.
```

Fatal errors that stop execution include:

- CI is unable to read the CI.INI file
- CI is unable to process command line parameters
- CI is unable to access the SQL Server database
- No import option is selected

- The specified working folder cannot be found
- The specified output folder cannot be found
- There are untranslated securities (CI for PC, Axys, or APX)
- Any error when trying to access the internet or the www.byallaccounts.net URL via the internet
- CI fails to import AccountView data
- CI fails to create or write to output files
- The custom translation file, CiTxTransCommonCust.xml, is not correct
- The transaction filter file, CiTxFilter.xml, is not correct
- The position filter file, CiPosFilter.xml, is not correct

Non-Fatal Errors that Do Not Stop Execution

For non-fatal errors, the error is logged, the current step that is being executed is terminated, and the next step is attempted. Non-fatal errors include:

- Any error during the Account Update process.
- Import errors, such as loading any target system data (such as Axys security and price information) or validating security data (untranslated securities and duplicated data).
- Invalid account translations. Invalid account translations are disabled in CI Automation, allowing automation run to continue as if no invalid translations existed; the output files will contain the data for only the valid account translations. The entry in the AutoRunHistory.log file looks like this:
  ```
  Error: Found invalid account translations. Info: Invalid translations have been disabled.
  ```

Appendix A: CI Automation Parameters

Most of the parameters described in this section are specific to CI Automation and should be set in the runCIAuto.bat file. However, the output parameters apply to CI and CI Automation.

You may need additional parameters for your variant of CI beyond those shown here, and typically those as well as the output parameters should be set in the CI.ini file so they are used for both CI and CI Automation. The additional parameters are described in your *Custodial Integrator Installation Guide* along with details about how to set them.

Before setting any parameters, refer to "Set Up and Run CI Automation", page 7 for details.

Note: These CI Automation parameters listed here are in alphabetical order, other than the autorun parameter, and not in order of importance.

Parameter label	Parameter value(s)	Default	Description	Common?
autoRun				
	SETUP IMPORT EXPORT ACCEPT	none	The autoRun parameter tells CI to run automatically and not display any user interface. Include one of the following values to designate the step up through which to run CI Automation. **NOTE:** This value is unrelated to the manual Setup step in CI and does not replace the need to make settings in the user interface as described on page 4. ▪ **SETUP** is a pre-import setting that causes CI Automation to only perform some initial operations and then exit before the Import step. Certain secondary files can be exported by running the SETUP step only and setting the associated output parameters. See Appendix G: "Example of Secondary File Output" for an example of a case where setting autorun=SETUP can be helpful. ▪ **IMPORT** means do Import step only. ▪ **EXPORT** means do Import and Export steps. ▪ **ACCEPT** means do Import, Export, and Accept steps. **Note**: It is recommended that while you are setting up and testing CI Automation that you do not use the ACCEPT value. If you set parameters to request secondary files, refer to "Export of Secondary Files with CI and CI Automation", page 6.	Yes

Parameter label	Parameter value(s)	Default	Description	Common?
acceptFile				
	(file name)		Used with acceptFileTimeout, provides control over when the Accept step is executed during an automated run. If the autoRun parameter is set to ACCEPT and the acceptFile parameter is set to a filename, then the Accept step will not occur until the specified file exists or until the timeout occurs. See the acceptFileTimeout parameter for timeout control.	Rare
acceptFileTimeout				
	(number)	10	Value is the number of minutes to wait for acceptFile to appear. See acceptFile parameter description for more details.	Rare
defaultNumPriceDays				
	(number)	(The number of days since the last CI export, with a maximum of 6 days.)	When this parameter is set in CI or CI Automation it specifies the number of days of historical price data that CI should import and export to the Price and Security files. The valid range of values is 1-6 where 1 indicates only the previous business day. CI will use either the number of days from the parameter (if specified), or the number of days since last CI export, whichever is greater.	Yes
identifierAllowedCharacters				
	(special characters) none all	all	Specifies which special (non-alphanumeric) characters are allowed in the account identifier when requesting automatic account translations. Can be set to allow specified non-alphanumeric characters, none, or all. Applies to both CI interactive (GUI) and CI Autorun. When used in interactive mode, removal of non-allowed characters is enforced only when multiple accounts are selected in bulk for automatic translation. Any characters are allowed when the user manually types the identifier for a single account. Can be set to: ▪ none - only alphanumeric characters will be kept in the account identifier. ▪ all - when set to all or the parameter is not present, then all special characters are allowed in the account identifier.	No Only used in Universal, PC, and UPC

Parameter label	Parameter value(s)	Default	Description	Common?

			▪ One or more of the following characters with no separators: ._\\/+?!;:,%!@*()\"{}^|<>#$=&'`~; - See notes below.
			Notes for special characters:
			Can be specified in .bat scripts (such as runCI.bat or runciauto.bat), but we recommend specifying it in CI.ini for more flexibility setting special characters.
			The character list cannot contain square brackets (neither [or]).
			If specified in CI.ini, the list of characters should be entered with no surrounding double quotes, including the case when the list contains or starts with a space. A double quote can be specified as a special character in the CI.ini file.
			Example for CI.ini:
			`identifierAllowedCharacters=._ \%;"-/+!`
			Conversely, for.bat scripts (for example runciauto.bat), the list of characters must be contained within double quotes, and a double quote (") cannot be specified as an allowed character. To specify a back slash (\) or percent (%) in a .bat file it must be preceded by another incident of itself (\\ or %%) to take effect. Example of runCI.bat:
			`identifierAllowedCharacters=". _ \\%%;-/+!"`
			Applies to both CI GUI and CI Autorun.	

includeAccountsAtFis				
	(comma-separated list of financial institution IDs)		When specified, this parameter causes the automatic account translation capability that runs during "autorun" to only consider for translations accounts that are at any of the listed financial institution IDs and ignore accounts that are NOT at the listed financial institution IDs. Any existing account translations will not be affected. There is no effect on the user interface; this parameter does not prevent a user from adding translations for accounts at an FI that is not in the list.	No
includeAccountsNotAtFis				
	(comma-separated list of financial		When specified, this parameter causes the automatic account translation capability that runs during "autorun" to only	No

Parameter label	Parameter value(s)	Default	Description	Common?
	institution IDs)		consider for translation those accounts that are NOT at any of the listed FIs and to ignore accounts that ARE at the listed FIs. Any existing account translations will not be affected. There is no effect on the user interface; this parameter does not prevent a user from adding translations for accounts at an FI that is not in the list. Note: If the includeAccountsAtFis parameter is set then this parameter is ignored.	
ignorePositionalExtServLevelAccounts				
	y n	n	Specifies whether accounts that have the value "Positional" in the `EXTERNAL_SERVICE_LEVEL` field should be ignored in CI. When the parameter is set to 'y', such accounts will not appear in the list of untranslated accounts and will not be translated in CI Autorun. Therefore, these accounts will never show in CI exported files. Effective both in CI interactive and in CI autorun modes. **Important note for those upgrading from any release prior to CI 3.15**: Refer to your release notes for 3.15 for important upgrade information.	No
includeLots				
	y n	n	Because position lots (tax lots) should be imported infrequently, they are controlled with a parameter instead of with a default setting. When this parameter is set to 'y', CI Automation imports position lots in addition to other data (securities, positions, etc.) configured to be imported. Consider creating an additional .bat file that is for automated tax lot gathering and use it sparingly. **Note:** This parameter is only effective for CI Automation, and only when your firm is licensed for position lot data. Refer to the *Position Lots Guide*.	Rare

Parameter label	Parameter value(s)	Default	Description	Common?
minUpdatedAccountsPerc				
	(0-100)		Optionally used to specify the minimum percentage (0-100) of updated accounts required for CI Automation to proceed with import and subsequent steps. A valid value for the parameter is any number 0 to 100. If the percentage of updated accounts is below the specified minimum, CI Autorun will exit without importing. If the parameter is not specified or the value is 0, CI will run its cycle independently of the updated accounts percentage. For CI Universal this parameter is only processed if the CI Configuration option "Select for import only up-to-date and offline accounts that have not yet been exported today" is not checked. That option does not exist in CI PC, Axys and APX, so for this caveat does not apply to them.	No
outputAccountTranslations				
	y n	n	When this parameter is set to y (yes), CI or CI Automation generates the account translations file ACCTTRANSLATIONS_<date>.csv. When this parameter is set in conjunction with the autoRun=SETUP or autoRun=IMPORT, the file is exported to the output folder (as defined in the Configuration tab) as soon as CI Automation exits. See "Export of Secondary Files with CI and CI Automation" page 6 for additional information about output of secondary files in CI versus CI Automation. For details about the file content and an example of the file refer to "Appendix B: Examples of Secondary Output Files".	Yes

Parameter label	Parameter value(s)	Default	Description	Common?
outputAllSecondaryFiles				
	y n	n	This is a catch-all parameter for secondary output files. When set to y, this parameter enables all of the output parameters (Missing prices file is only exported if Prices are also exported.)	Yes
outputMissingPriceFile				
	y n	n	When this parameter is set to y (yes), CI or CI Automation lists missing prices in the MISSINGPRICES_<date>.csv file. Unlike the other secondary files that can be output in CI Automation without running the Export step, output of the MISSINGPRICES file requires running the Export step and having the 'Include prices' option selected in CI Configuration screen. For details about the file content and an example of the file Examples of Secondary Output Files "Appendix B: Examples of Secondary Output Files".	Yes
outputSecurityTranslations				
	y n	n	When this parameter is set to y (yes), CI or CI Automation generates the file listing manual security translations (SECTRANSLATIONS_<date>.csv). When this parameter is set in conjunction with the autoRun=SETUP or autoRun=IMPORT, the file is exported to the output folder (as defined in the Configuration tab) as soon as CI Automation exits. See "Export of Secondary Files with CI and CI Automation" page 6 for additional information about output of secondary files in CI versus CI Automation. For information about the content and format of the Security Translations .csv file, refer to "Appendix B: Examples of Secondary Output Files" and to your *Custodial Integrator User Guide.*	Yes

Parameter label	Parameter value(s)	Default	Description	Common?
outputStaleAccounts				
	y n	n	When this parameter is set to y (yes), CI or CI Automation generates the stale accounts file (STALEACCT _<date>.csv). When this parameter is set in conjunction with the autoRun=IMPORT the file is exported to the output folder (as defined in the Configuration tab) as soon as CI Automation exits. See "Export of Secondary Files with CI and CI Automation" page 6 for additional information about output of secondary files in CI versus CI Automation. For details about the file content and an example of the file refer to "Appendix B: Examples of Secondary Output Files". **Note:** The file may include an "Export Positions" column containing "yes" or "no" values to indicate whether or not the positions were exported. Settings in the CI user interface control whether this column appears. To include it, either or both Advanced settings for the **Include reconciliation positions for export** option must be selected. That option appears on the Configuration tab.	Yes
outputUntranslatedAccounts				
	y n	n	When this parameter is set to y (yes), CI or CI Automation generates the untranslated accounts file, which is UNTRANSLATEDACCT_<date>.csv. When this parameter is set in conjunction with the autoRun=SETUP or autoRun=IMPORT the file is exported to the output folder (as defined in the Configuration tab) as soon as CI Automation exits. See "Export of Secondary Files with CI and CI Automation" page 6 for additional information about output of secondary files in CI versus CI Automation. For details about the file content and an example of the file refer to "Appendix B: Examples of Secondary Output Files".	Yes
outputUntranslatedSecurities				
	y n	n	When this parameter is set to y (yes), CI or CI Automation will generate two comma-separated values (.CSV) files of	Yes

Parameter label	Parameter value(s)	Default	Description	Common?
			information about Untranslated Securities (exceptions). ▪ The UNDEFSEC_<date>.csv file contains basic information in a row for each untranslated security, including security name and the name of the financial institution in which it was referenced. ▪ The UNDEFSECDET_<date>.csv file provides more detailed information about each holding and/or transaction that references the securities listed in the first file. It lists the holdings followed by the transactions. When this parameter is set in conjunction with the autoRun=IMPORT the file is exported to the output folder (as defined in the Configuration tab) as soon as CI Automation exits. See "Export of Secondary Files with CI and CI Automation" page 6 for additional information about output of secondary files in CI versus CI Automation. For examples of these files refer to "Appendix B: Examples of Secondary Output Files". How your variant of CI behaves when there are untranslated securities is described in "How CI Variants Handle Untranslated Securities", page 31.	

Parameter label	Parameter value(s)	Default	Description	Common?
translateAccountValidationStatus				
	all validatedOnly validatedAndAggregated	validatedOnly	*This parameter is used only when translateUntranslatedAccounts=y* This parameter specifies whether CI Automation will translate both validated and non-validated accounts, only validated accounts, or only accounts that are both validated and aggregating today. ▪ **all** means translate both validated and non-validated accounts ▪ **validatedOnly** means translate only validated accounts ▪ **validatedAndAggregated** means translate only accounts that are both validated and aggregating today Note that for APX and Axys this parameter additionally requires `defaultAccountIdentifier` be set to `WPAccountNumber` or `WPAccountName`	Yes
translateEarliestTxDate				
	Valid date in format: YYYYmmDD	(Today's date)	*This parameter is used only when translateUntranslatedAccounts=y* Specifies the earliest transaction trade date that will be used to filter transactions on the first download. Only transactions with trade date on or after this date will be delivered in the first download.	Yes

Parameter label	Parameter value(s)	Default	Description	Common?
translateFeedEarliestTxDate				
	Valid date in format: YYYYmmDD	(Today's date)	*This parameter is used only when translateUntranslatedAccounts=y* In CI all variants: specifies the transaction date to be used for the initial import data for accounts at FIs designated as "feeds". Accounts at FIs that are not designated as feeds will continue to use the `translateEarliestTxDate` for first import. Requires `translateUntranslatedAccounts` parameter. Note that for APX and Axys this parameter additionally requires `defaultAccountIdentifier` be set to `WPAccountNumber` or `WPAccountName` For CI Universal (only) Setting this parameter will also affect the position export for failed accounts at FIs designated as "feeds". Positions will NOT be exported for feed accounts that failed aggregation. The setting for failed accounts in the "advanced" positions-export dialog will not be used in this case, but will still apply to non-feed accounts and the setting for Stale accounts will still apply to both feed and non-feed accounts. Note that setting this parameter will affect export of positions for feed accounts in both automation and interactive mode. Conversely, the first feed transaction date specified with the parameter will be used for translation of feed accounts only in CI automation.	Yes
translateUntranslatedAccounts				
	y n	n	When this parameter is not set or is set to n (no), CI Automation will skip untranslated accounts and will not import or export them. If set to y (yes), CI Automation will automatically translate (map) each untranslated account to a default account identifier. For this parameter to work, there must be a default account identifier. The field used as the default account identifier depends on the variant of CI. ▪ For CI Universal, CI for PC, and CI for	Yes

Parameter label	Parameter value(s)	Default	Description	Common?
			UPC, WP Account Number is used as the account identifier unless a different field is set using the *defaultAccountIdentifier* parameter. If the *defaultAccountIdentifier* parameter is set to WebPortfolio account number or WebPortfolio account name then when automation is translating untranslated securities it automatically appends the internal account ID to new duplicate account identifiers to make them unique. Note that if the total length exceeds the maximum account identifier length, then the WebPortfolio account number or WebPortfolio account name (depending on the defaultAccountIdentifier setting) will be truncated to make it fit into the account identifier length. ▪ For CI for Axys and CI for APX, there is no default unless it is set using the *defaultAccountIdentifier* parameter. If *defaultAccountIdentifier* is available for the variant of CI you are using, it is listed and described in your *Custodial Integrator Installation Guide*. The installation guides can all be found at http://www.byallaccounts.net/manuals/Content/CI/CustodialIntegrator.htm. When *updateStaleAccounts* is set, stale accounts are updated after untranslated accounts are translated and prior to the Import step.	

Parameter label	Parameter value(s)	Default	Description	Common?
updateStaleAccounts				
	y n	n	If set to y (yes) and the autorun parameter is specified, then CI Automation will perform a Demand Update and force the gathering of data from the custodian on stale accounts prior to the Import step. CI Automation considers an account to be stale if the "last update" date of the account is not today. CI performs the Demand Update only once and accounts may still be stale after this update completes. CI Automation will not request an update on demand for accounts that show a 1007 error (bad user login or password). You should review these accounts, update the login and/or password in AccountView, and then request a manual update. Repeated attempts to Demand Update an account with a bad login or password may result in the account being "locked out" by the Financial Institution and will then require contacting the institution to have access re-enabled.	Yes

Appendix B: Examples of Secondary Output Files

CI Automation can generate secondary output files, depending on parameters described in "Appendix A: CI Automation Parameters".

Note that the names of files output by CI Universal are prefixed with BAA.

- Account Translations file
- Missing Prices file
- Security Translations file
- Stale Accounts file
- Untranslated Accounts file
- Untranslated Securities file and Detailed Undefined Securities file

Account Translations file

The outputAccountTranslations parameter causes CI Automaton to generate the ACCTTRANSLATIONS _<date>.csv file. The following is an example of an Account Translations file:

ACCOUNT ID	WP ACCOUNT	WP ACCOUNT #	INSTITUTION	ACCOUNT UPDATE STATUS INFO	LAST UPDATED	LAST EXPORTED	TX DATE	VALIDATOR	MM SYMBOL	SWEEP SYMBOL	EXPORT POSITIONS IF STALE	IMPORT STATUS	TRANSLATION STATE
16004	Charles Schwab Instl 16004	1111-2222	Charles Schwab, Inc. - Institutional (Investment)	Update successful on Mar 7, 2012 10:13:47 AM	20120307		20120305				Yes	Enabled	Valid
16005	Merrill Lynch Direct 16005	88888888	Merrill Lynch Direct (Investment)	Update successful on Mar 7, 2012 10:17:05 AM	20120307		20120305				Yes	Enabled	Valid

Missing Prices file

The outputMissingPriceFile parameter causes CI Automaton to generate the MISSINGPRICES _<date>.csv file. The following is an example of a Missing Prices file:

SYMBOL	TYPE	NAME	WP ACCOUNT	INSTITUTION
353538200	MUTUALFUND	FRANKLIN AGE HIGH INCOME-C	Merrill Lynch Direct 16005	
G0070K103	STOCK	ACE LTD	Charles Schwab Instl 16004	
53015103	STOCK	AUTOMATIC DATA PROCESSING	Merrill Lynch Direct 16008	
1055102	STOCK	AFLAC INC	Charles Schwab Instl 16004	
31162100	STOCK	AMGEN INC	Charles Schwab Instl 16004	
32511107	STOCK	ANADARKO PETROLEUM CP	Charles Schwab Instl 16004	
25816109	STOCK	American Express	Charles Schwab Instl 16004	

Security Translations File

The *outputSecurityTranslations* parameter causes CI Automation to generate the SECTRANSLATIONS_<date>.csv file.

The following is an example of a Security Translations file. See your *Custodial Integrator User Guide* for more information about this file.

WP Name	WP Ticker	WP Cusip	Institution	WP Account	Symbol	Symbol Type	Security Type	Created	Last Modified
D PEPPER GROUP			Charles Schwab, Inc Institutional (Investment)		DPS	Ticker	STOCK	20121115	20121115
UBS AG REG			Merrill Lynch Direct (Investment)		76CX3	Ticker	MUTUAL FUND	20121115	20121115

Stale Accounts File

The *outputStaleAccounts* parameter causes CI Automaton to generate the STALEACCT _<date>.csv file. The following is an example of a Stale Accounts file:

WP Account	Account #	Institution	Stale Positions
Charles Schwab Inst 16007 Modified	2222-3333	43634	2

Untranslated Accounts File

The *outputUntranslatedAccounts* parameter causes CI Automaton to generate the UNTRANSLATEDACCT_<date>.csv file. The following is an example of an Untranslated Accounts file:

WP Account	Account #	Institution	Last Updated	Validation
Charles Schwab Inst 16004	1111-2222	Charles Schwab, Inc Institutional (Investment)	20120307	
Merrill Lynch Direct 16005	88888888	Merrill Lynch Direct (Investment)	20120307	

Untranslated Securities Files

The *outputUntranslatedSecurities* parameter causes CI Automation to generate two output files:

- the untranslated securities file UNDEFSEC_<date>.csv.
- the detailed untranslated securities file UNDEFSECDET_<date>.csv . This file includes the same detailed information as in the CI user interface, in two separate tables.

The following is an example of an untranslated securities file (UNDEFSEC_<date>.csv):

NAME	FINANCIAL_INSTITUTION
AMERIPRISE FINANCIAL INC	Charles Schwab Inc. - Institutional (Investment)
DUKE REALTY CORP	Merrill Lynch Direct (Investment)
HESS CORPORATION	Charles Schwab Inc. - Institutional (Investment)
SCH ADV CASH RESRV PREM	Charles Schwab Inc. - Institutional (Investment)
TRAVELERS COMPANIES INC	Charles Schwab Inc. - Institutional (Investment)

The following is an example of a detailed untranslated securities file (UNDEFSECDET_<date>.csv):

NAME	FINANCIAL_ INSTITUTION	ACCOUNT_ NAME	ACCOUNT_ NUMBER	TOTAL_ AMOUNT	UNITS	EXECUTION_ DATE	
SCH ADV CASH RESRV PREM	Charles Schwab Inc. - Institutional	Charles Schwab Inst 16007	2222-3333	-23.53	24	20120306	
SCH ADV CASH RESRV PREM	Charles Schwab Inc. - Institutional	Charles Schwab Inst 16007	2222-3333	-166.52	166.52	20120302	
NAME	FINANCIAL_ INSTITUTION	ACCOUNT_ NAME	ACCOUNT_ NUMBER	TOTAL_ AMOUNT	UNITS	PRICE	PURCHASE_ DATE
HESS CORPORATION	Charles Schwab Inc. - Institutional	Charles Schwab Inst 16007	2222-3333	7285.52	86		20110104
TRAVELERS COMPANIES INC	Charles Schwab Inc. - Institutional	Charles Schwab Inst 16007	2222-3333	6509.43	121		20081117
AMERIPRISE FINANCIAL INC	Charles Schwab Inc. - Institutional	Charles Schwab Inst 16007	2222-3333	4962.41	112		20110407
DUKE REALTY CORP	Merrill Lynch Direct	Merrill Lynch Direct 16008	88888889	6212.68	500	12.77	19991213
DUKE REALTY CORP	Merrill Lynch Direct	Merrill Lynch Direct 16008	88888889	2486.65	200	12.77	19991213
DUKE REALTY CORP	Merrill Lynch Direct	Merrill Lynch Direct 16008	88888889	1243.32	100	12.77	19991213

Appendix C: AUTORUNHISTORY.LOG FILE EXAMPLES

When CI Automation is run it creates a separate log file (AutoRunHistory.log) in the working/log folder. The file contains a running history of each Startup, Account Update, Import step, Export step, Accept step, and Exit.

Note that these examples do not show the Accept step because autoRun was not set to Accept. If the Accept step is not performed due to a timeout while waiting for the acceptFile then an "Accept not performed due to timeout" entry is added to the log.

Example 1

Here is an example of the content of the AutoRunHistory.log file:

```
07-07 14:57:23.754: main: Custodial Integrator for Universal V3.2.002 started
with database 'DevCIUniversal' on SQL host 'localhost\SQLEXPRESS'.
07-07 14:57:23.770: main: Import started...
07-07 14:58:09.837: main: Import complete.
07-07 14:58:09.837: main: Export started...
07-07 14:58:11.225: main: Export complete.
07-07 14:58:11.225: main: CI exiting.
```

Example 2

Here is an example of a log file when the **updateStaleAccounts** parameter was used. CI updated stale accounts.

```
07-07 15:04:36.049: main: Custodial Integrator for Universal V3.2.002 started
with database 'DevCIUniversal' on SQL host 'localhost\SQLEXPRESS'.
07-07 15:04:36.080: main: Mapping accounts for 8 unmapped accounts...
07-07 15:04:36.080: main: Mapping accounts completed for 8 accounts. 0 failed to
map.
07-07 15:04:36.111: main: Updating stale accounts...
07-07 15:04:36.938: main: Updating stale accounts complete.
07-07 15:04:36.938: main: Import started...
07-07 15:05:25.517: main: Import complete.
07-07 15:05:25.517: main: Export started...
07-07 15:05:27.030: main: Export complete.
07-07 15:05:27.030: main: CI exiting.
```

Example 3

Here is an example of a log file when automatic account translation is turned on, using the **translateUntranslatedAccounts** parameter. In this example, it translated validated and non-validated accounts.

```
07-07 15:00:02.594: main: Custodial Integrator for Universal V3.2.002 started
with database 'DevCIUniversal' on SQL host 'localhost\SQLEXPRESS'.
07-07 15:00:03.967: main: Mapping accounts for 16 unmapped accounts...
07-07 15:00:03.967: main: Mapping accounts completed for 16 accounts. 0 failed
to map.
07-07 15:00:04.045: main: Import started...
07-07 15:00:52.218: main: Import complete.
07-07 15:00:52.218: main: Export started...
07-07 15:00:53.794: main: Export complete.
07-07 15:00:53.794: main: CI exiting.
```

Example 4

Here is an example of a log file when automatic account translation is turned on, using the **translateUntranslatedAccounts** parameter. In this example, it only translated validated accounts because the **translateAccountValidationStatus** parameter was used and set to **validatedOnly**.

```
07-07 14:52:57.554: main: Custodial Integrator for Universal V3.2.002 started
with database 'DevCIUniversal' on SQL host 'localhost\SQLEXPRESS'.
07-07 14:52:58.958: main: Mapping accounts for 16 unmapped accounts. 12 non
validated accounts will not be mapped...
07-07 14:52:58.973: main: Mapping accounts completed for 4 accounts. 0 failed to
map.
07-07 14:52:58.989: main: Import started...
07-07 14:53:00.861: main: Import complete.
07-07 14:53:00.861: main: Export started...
07-07 14:53:01.844: main: Export complete.
07-07 14:53:01.844: main: CI exiting.
```

Appendix D: Handling Untranslated Securities

How CI Variants Handle Untranslated Securities

The way CI and CI Automation handle untranslated (undefined) securities depends on the variant of CI you are using. An overview of the difference is described in Step 3 in "Overall Processing Flow for CI and CI Automation" on page 3. The following table explains the differences in more detail.

Variant	Behavior for Untranslated Securities
CI Universal	Will export data even if there are untranslated securities, and will use a dummy security symbol. The string used in place of the security symbol is set in Output Configuration of Setup.
CI for PortfolioCenter, CI for Axys, CI for APX	These variants cannot export any data if there are untranslated securities. CI will require the user to translate the securities and CI Automation will stop (exit) and will produce an error in the AutoRunHistory.log file. CI Automation cannot automatically create translations for untranslated securities, therefore the user must manually translate all untranslated securities in both in CI and CI Automation. After all securities are translated, CI Automation can be launched again and will proceed and be ready for the Export step.

Appendix E: Handling Stale Accounts

If some of your accounts tend to be consistently one day stale when you run CI Automation, you can use the following strategy to avoid a long wait in getting the data for the up-to-date accounts.

1. In CI Setup, check the checkbox "Select for import only up-to date accounts that have not yet been exported today". This is a one-time setting; you do not have to set this again.
2. Run CI Automation with: *autorun: ACCEPT updateStaleAccounts:n outputStaleAccounts:y*. CI Automation will import and export data only for up-to-date accounts. It will also generate a file that contains the list of stale accounts (if there are any).
3. Check the list of the stale accounts. If there are stale accounts, run CI Automation a second time (possibly a bit later) with the same parameters, but set *updateStaleAccounts:y.* This time CI Automation will work only on the stale accounts; first it will update the stale accounts and then proceed to import and export the ones that were successfully updated. Accounts exported in the first CI run will not be exported again.

Step 1 is part of the initial CI setup done via UI. Steps 2 and 3 can be set to run automatically every day at pre-set times.

Appendix F: Multiple CI Instances and Databases

If you run multiple instances of CI, there are a few considerations you must understand and manage when running them using CI Automation.

Setting the Working and Output Folders for Each Instance

CI V2.1 and later versions require the designation of a CI Working Folder where CI writes temporary files, including log files and debug files.

If an installation has multiple CI instances/databases (one instance per target output system or per client) configure a separate CI Working Folder for each CI instance. Doing so will enable you to later review log files and Untranslated Security files that are specific to that CI instance.

Within an instance of CI, you can specify the working folder and the output folder for it to use. Those settings are in Setup, in the Configuration tab.

When each CI instance has a separate database, separate target output folders, and separate working folders, you can run CI Automation for each of the multiple instances concurrently.

IMPORTANT NOTE: Refer to your *Installation Guide* for important configuration information regarding password encryption when multiple CI instances or multiple databases are used.

Using Multiple Versions of the .bat File

As described in "Using a runCIAuto.bat File" on page 9, it is recommended that you specify all of the automation parameters in a customized version of the .bat file. If you are running multiple instances of CI, you can create a custom version of the .bat file for each CI instance.

For example, if you have two instances of CI you might have:

- Two databases, one for each instance: MYCUSTDB1 and MYCUSTDB2
- A bat file customized for each: runCIAutoMyCust1.bat and runCIAutoMyCust2.bat

Within each of the .bat files, you can designate which database you want that instance of CI to run. For example, for the runCIAutoMyCust1.bat file use a form like this to specify MYCUSTDB1 as the database:

> java -cp CI.jar;extlib\sqljdbc.jar;extlib\gnu-regexp.jar com.baa.aci.Cache tool:PC **dbname:MYCUSTDB1 autoRun:ACCEPT** updateStaleAccounts:y outputUntranslatedSecurities:y

Note: Be certain to construct the java command starting with the command defined in your runCI.bat. Your java command may be different than the one used in this example.

Using Custom Filter and Translation Files

If you use custom filter and translation files, you can place them in the CI root folder (by default \Program Files\Custodial Integrator) and CI Automation will automatically use them. If you have multiple CI instances and databases, and you want to use different custom filter and translation files, then be sure to put the files for each instance in input folders dedicated to each instance. Then identify the input folder in the .bat file. The files are:

- Custom translation file, CiTxTransCommonCust.xml
- Transaction filter file, CiTxFilter.xml
- Position filter file, CiPosFilter.xml

The following example shows how to specify the location of custom filter and translation files as well as the database to use for a particular instance. It uses the inputFolder parameter, which is described in your *Custodial Integrator Installation Guide*. Note that in this example the inputFolder parameter specifies C:\Program Files\Custodial Integrator\input-files as the location of the custom filter and translation files.

java -cp CI.jar;extlib\sqljdbc.jar;extlib\gnu-regexp.jar com.baa.aci.Cache tool:PC **inputFolder:"C:\Program Files\Custodial Integrator\input-files" dbname:MYCUSTDB1 autoRun:ACCEPT** updateStaleAccounts:y outputSecurityTranslations:y

Appendix G: Example of Secondary File Output

The following example shows a simplified case where the capability of exporting secondary files without performing Export or Import may come in handy in CI Automaton.

A BAA customer running CI Automation for multiple clients wants to automatically get a daily list of untranslated accounts for each of their clients. If such accounts exist, they have to be manually translated so the new accounts will be included in the daily CI import/export cycle for that client. Suppose that this BAA customer cannot take advantage of the automatic account translation feature offered in CI Automation because additional human judgment is required to correctly map each account. However the customer would like to eliminate the need to manually start CI for each of their clients just to determine whether any untranslated accounts exist.

This case can be solved with a script that runs CI Automation in two steps:

Step 1: Execute CI Automation with the following parameter settings:

- *autorun=SETUP* (Run only the 1st Automation step)
- *outputUntranslatedAccounts=y* (Output the Untranslated Accounts file)

Step 2: Check to see if CI output folder contains an Untranslated Accounts file with today's date:

- If the file exists, then exit. (User to manually translate untranslated accounts and execute script again.)
- If the file is not present, then run CI Automation again, this time through the last step. Parameter setting:
  - *autorun=ACCEPT* (Run the whole CI cycle including Accept)