

Custodial Integrator

Installation Guide

(Axys)

---

©2025 Morningstar. All Rights Reserved.

Axys is a registered trademark of Advent Software, Inc.

Custodial Integrator Product Version:	V3.19
Document Version:	36
Document Issue Date:	April 10, 2025

Technical Support:	(866) 856-4951
Web:	<a href="http://byallaccounts.morningstar.com">byallaccounts.morningstar.com</a>

## Table of Contents

<b>ABOUT THE MANUAL.....</b>	<b>4</b>
AUDIENCE AND FORMAT .....	4
RELATED DOCUMENTS .....	4
<b>INSTALL REQUIREMENTS.....</b>	<b>5</b>
SYSTEM OVERVIEW .....	5
SYSTEM REQUIREMENTS .....	6
CI COMPUTER.....	6
DATABASE COMPUTER .....	6
AXYS.....	6
REQUIRED SOFTWARE .....	7
ADDITIONAL CONSIDERATIONS .....	7
<b>INSTALL CUSTODIAL INTEGRATOR .....</b>	<b>9</b>
OVERVIEW .....	9
SOFTWARE DISTRIBUTION.....	9
CUSTODIAL INTEGRATOR.....	9
PREPARE TO INSTALL.....	10
INSTALL .....	11
DATABASE COMPUTER INSTALL – SQL SERVER .....	11
CI COMPUTER INSTALL OR UPGRADE .....	19
CONFIGURE CUSTODIAL INTEGRATOR.....	20
CONSIDER PASSWORD ENCRYPTION MODELS .....	23
INSTALL MULTIPLE INSTANCES THAT ACCESS THE SAME DATABASE .....	23
SETUP FOR MULTIPLE CIs THAT ACCESS THE SAME DATABASE .....	24
DEFAULT ENCRYPTION MODEL .....	24
ENHANCED ENCRYPTION MODEL .....	24
INSTALL MULTIPLE INSTANCES OF CI ON SAME MACHINE WITH DIFFERENT DATABASES .....	26
TROUBLESHOOT INSTALL PROBLEMS.....	27
GENERAL TECHNIQUES .....	27
KNOWN ISSUES .....	28
<b>CUSTODIAL INTEGRATOR CONFIGURATION REFERENCE .....</b>	<b>30</b>
CI PARAMETERS .....	30

CI CUSTOMIZATIONS..... 31

AXYS CUSTOMIZATIONS ..... 34

CORPORATE FIREWALL ..... 39

DATABASE..... 40

DEBUGGING..... 41

## ABOUT THE MANUAL

### **Audience and Format**

This manual describes the system requirements for the Custodial Integrator ("CI") product, the procedure for installing CI, and how to perform initial configuration of CI. The audience for this manual is an Information Technology professional who is responsible for installing and maintaining software systems. You should be familiar with basic system administration procedures, manipulation of Windows services, and working directly in the Windows file system.

### **Related Documents**

The following related documents are available from ByAllAccounts:

- [The Custodial Integrator Solution](#): provides an overview of the Custodial Integrator solution.
- [Custodial Integrator Validation FAQ](#): provides an overview of the account validation process.

## INSTALL REQUIREMENTS

### System Overview

The CI architecture consists of the following three main components:

- CI Database: a Microsoft SQL Server database
- CI Installation Folder: a Windows folder hierarchy where permanent files are stored
- CI Working folder: a Windows folder hierarchy where temporary files are stored
- CI Application: the Java application that brings data from WebPortfolio service and prepares it for PortfolioCenter. CI is a “fat client” that presents a User Interface and interacts with the CI Database, CI Folder, and the WebPortfolio web service. The application requires Oracle Java Runtime Environment (JRE) software.

CI interacts with the following other components:

- Your Advent Axys installation
- The WebPortfolio Service web site at URL <https://www.byallaccounts.net/>

CI requires the following third-party software:

- Oracle Java Runtime Environment (JRE)
- Microsoft SQL Server

See [Required Software](#), page 7 for version requirements

The remainder of this manual refers to the following:

- Axys Computer: the computer where your Axys software is installed.
- CI Computer: the computer on which the CI application is installed and run. You will install the Java Runtime and CI on this computer.
- Database Computer: the computer where Microsoft SQL Server is installed. The CI database will be created on this computer.

Note that you may choose to install CI and the database software on the same computer.

## System Requirements

This section describes the base system requirements for the CI and third-party software installation.

### CI Computer

The following table defines the base system requirements for the CI Computer.

Component	Requirement	Notes
<b>CPU</b>	Intel Pentium or compatible 166-megahertz (MHz) or higher	
<b>Disk space</b>	15 MB	Additional space based on your download volume is needed to hold data downloads.
<b>RAM</b>	As recommended for your operating system	
<b>Operating System</b>	Windows 10 Windows Server 2008 Windows Server 2012 Windows Server 2012 R2 Windows Server 2016	Note: Running CI via Windows Terminal Services is supported.
<b>Internet connectivity</b>	Minimum 56k bps	
<b>Internet Browser</b>	Internet Explorer version 9.0 or higher	Browser is required only for the installation of the CI application.

### Database Computer

The database computer must comply with the minimum system requirements for Microsoft SQL Server edition that you are using. The supported versions of Microsoft SQL Server versions are indicated in [Required Software](#), page 7. Requirements differ for the full version of Microsoft SQL Server versus Express. Please refer to Microsoft's minimum system requirements description for the Microsoft SQL Server system you plan to use with CI to ensure you have the required CPU, disk space, RAM, and operating system.

The disk requirement for the CI Database is 70 MB.

### Axys

This version of CI supports Axys 3.5.1 or later. CI makes use of your Axys installation in the following ways:

1. CI uses the Axys Import/Export utility (imex32.exe) to export Security (sec.inf) and Security Type (type.inf) data from your Axys installation. This information enables CI to produce transactions, positions, and prices that use the security symbols and security types you have defined in Axys.
2. CI uses a listing of your Axys client file folder to build a list of Portfolio codes for use in CI.
3. CI uses the Axys Import/Export utility to import:
  - Transactions into a designated Trade Blotter.

- Prices into the Axys price file (.pri) for a specified date
4. CI uses the Axys Import/Export utility and the Position Post utility (pospos32.exe) to import:
    - Positions into a temporary Trade Blotter, then optionally into the Axys Position Blotter

### Required Software

CI requires the installation of the third-party software shown in the table below.

Software	Target Computer	Vendor	Version	Distribution	Disk Space
Microsoft SQL Server	Database Computer	Microsoft	SQL Server 2016 or later is required, including Express versions.	Available from Microsoft	Depends on the SQL Server version. Refer to Microsoft documentation.
Java Runtime Environment (JRE)	CI Computer	Oracle Corporation	1.8 or later	CI distribution: 1.8 update 261 is included in CI kit	239 MB (for 64-bit) 180 MB (for 32-bit)
Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL)	CI Computer	Microsoft	18.2 or later	CI distribution: Microsoft OLE DB Driver for SQL Server (MSOLEDBSQL) 18.5.0.0 is included in CI kit	11.1 MB

### Additional Considerations

1. CI Computer

The CI Computer should have good network proximity to the Axys Computer. CI will access the Axys executable files, as well as some Axys data files.
2. Database Computer

If you already have a version of Microsoft SQL installation supported by CI, you may use that installation to house the CI database. If you do not have a Microsoft SQL Server installation or choose not to use your existing installation, you may install Microsoft SQL Server Express on the CI Computer or on a separate Database Computer.
3. Internet Connectivity

If you have a corporate firewall, CI must be configured to tunnel through the firewall or to bypass the firewall to access to <https://www.byallaccounts.net>.
4. Multiple Users



This installation is designed to support a single CI user and should only be installed on a single computer. If you wish to install for multiple CI users or provide a redundant installation for failover, please contact ByAllAccounts for assistance.

## INSTALL CUSTODIAL INTEGRATOR

### Overview

The CI installation consists of the following steps, each of which is described in a subsequent section of this chapter:

1. Prepare for the installation by gathering required information and approvals
2. Install Microsoft SQL Server on the Database Computer. This is accomplished using a kit distributed by Microsoft. You may alternatively use a previously installed instance of Microsoft SQL Server.
3. Install the CI Application on the CI Computer. This install includes:
  - a. Java Runtime (pre-requisite)
  - b. CI product installation
  - c. CI Database initialization
4. Configure CI
5. Verify Installation

Depending on your configuration, you may need to refer to [As of version 3.7, CI supports two models](#) for managing encrypted user logins and passwords: default and enhanced. The default model stores the encrypted login/password combinations in the database and requires no special instructions. The enhanced model uses an on-disk keystore model that uses the password-protected keystore file (ciksfile) to encrypt and decrypt user logins and passwords which are stored in the database in an encrypted state. That model is more complicated to set up, especially when multiple instances of CI access the same database, described in [Setup for Multiple CIs that access the same database](#).

**Note:** Do not remove the PBEKeysetPass or KSKeysetPass parameters. Do not change them for any reason without express guidance from ByAllAccounts Technical Support. If the value of either does need to be changed, then the value must be changed to match in all instances.

These parameters are described in [Database](#), page 40.

Install multiple instances that access the same database (page [23](#)) or [Install multiple instances of CI on same machine with different](#) databases (page [26](#)).

### Software Distribution

#### Custodial Integrator

The CI software distribution is available on the Internet at the following URL:

<https://www.byallaccounts.net/CI/>

A login and password is required to access this distribution and is available on request for licensed customers. The distribution contains Custodial Integrator documentation and software kits.

## Prepare to Install

To prepare for the CI installation, please complete the following steps:

1. Verify System Requirements for both the CI Computer and the Database Computer as described in the previous section.
2. Existing Java Runtime Environment (JRE)
  - a. If a version of JRE has been installed on the CI Computer, determine the version. If the version is 1.8 or later, the installation will leave it as is. If it is not at least version 1.8, the installation will upgrade it to 1.8 update 261. Therefore, you must determine if this Java upgrade is acceptable on the CI computer. If another application on the CI computer depends on your current version of JRE and cannot tolerate a later version, you will have to determine whether to move this other application or to install CI on an alternate computer.
3. Windows Administrative Access
  - a. Obtain a Windows Login that is a member of the Administrators Group on the CI Computer for the CI installation
  - b. Obtain a Windows Login that is a member of the Administrators Group on the Database Computer for the SQL Server installation
4. Microsoft SQL Server (Note: for existing SQL Server installations only)
  - a. Request that your SQL Server administrator approve the installation of the CI database
  - b. Obtain the password for the “sa” user or obtain a Windows Login that has “sa” or “dbo” privileges in the existing SQL Server installation that allows Windows authentication.
5. Corporate Firewall

If you have a corporate firewall that restricts access to the Internet, you may need one or more of the following:

- a. Consultation from your ISA Server administrator as to the appropriate methods for tunneling or bypassing your firewall
  - b. Proxy host name
  - c. Proxy port for https (default: 443)
  - d. Proxy username and password (for Basic authentication only; NTLM authentication is not supported and requires firewall bypass)
6. Axys
  - a. Obtain the location of your Axys installation and information about some of the Axys Custom Labels that your Axys administrator may have defined (additional details provided later)

## Install

This section describes the installation procedure for CI and required third-party products. We recommend that you read through the entire text of the installation procedure before starting the installation.

### Database Computer Install – SQL Server

If you plan to use an existing Microsoft SQL Server installation, omit this step.

**Note:** These steps describe installing Microsoft SQL Server 2012 Express SP4, but you can install another version supported by CI.

During the installation, there are a few important options that you must select as described here for SQL Server to be configured properly for use with CI. Refer to [Required Software](#), page 7.

### Downloads

You can download Microsoft® SQL Server® 2012 Service Pack 4 (SP4) Express at this link  
<https://www.microsoft.com/en-us/download/details.aspx?id=56042>.

**Note:** If you are downloading the SQL Server Management Studio stand-alone application to perform a database migration, make sure to use the same version or a newer version than the SQL Server version used on your old server.

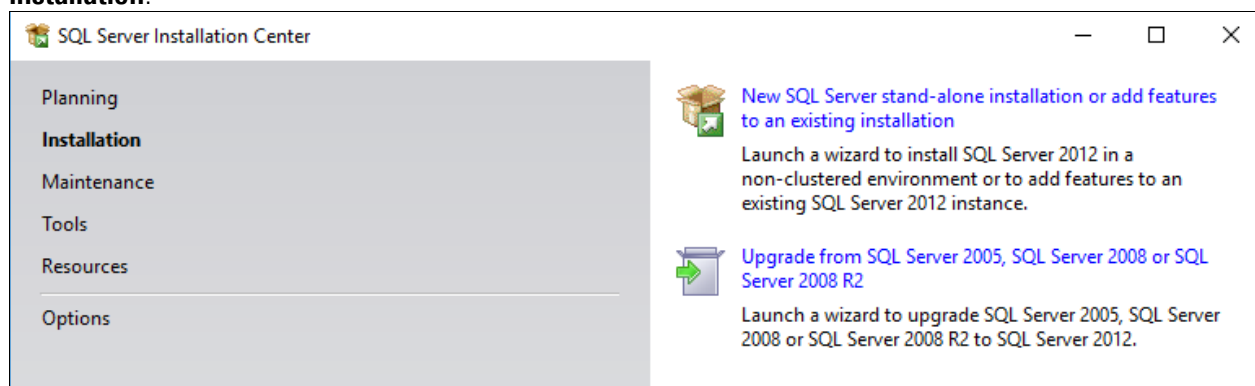
On the download pages for SQL Server Express 2012 SP4 or SQL Server Management Studio there are many options to choose from when downloading.

This list includes most of the files listed on the SQL Server download pages with their descriptions:

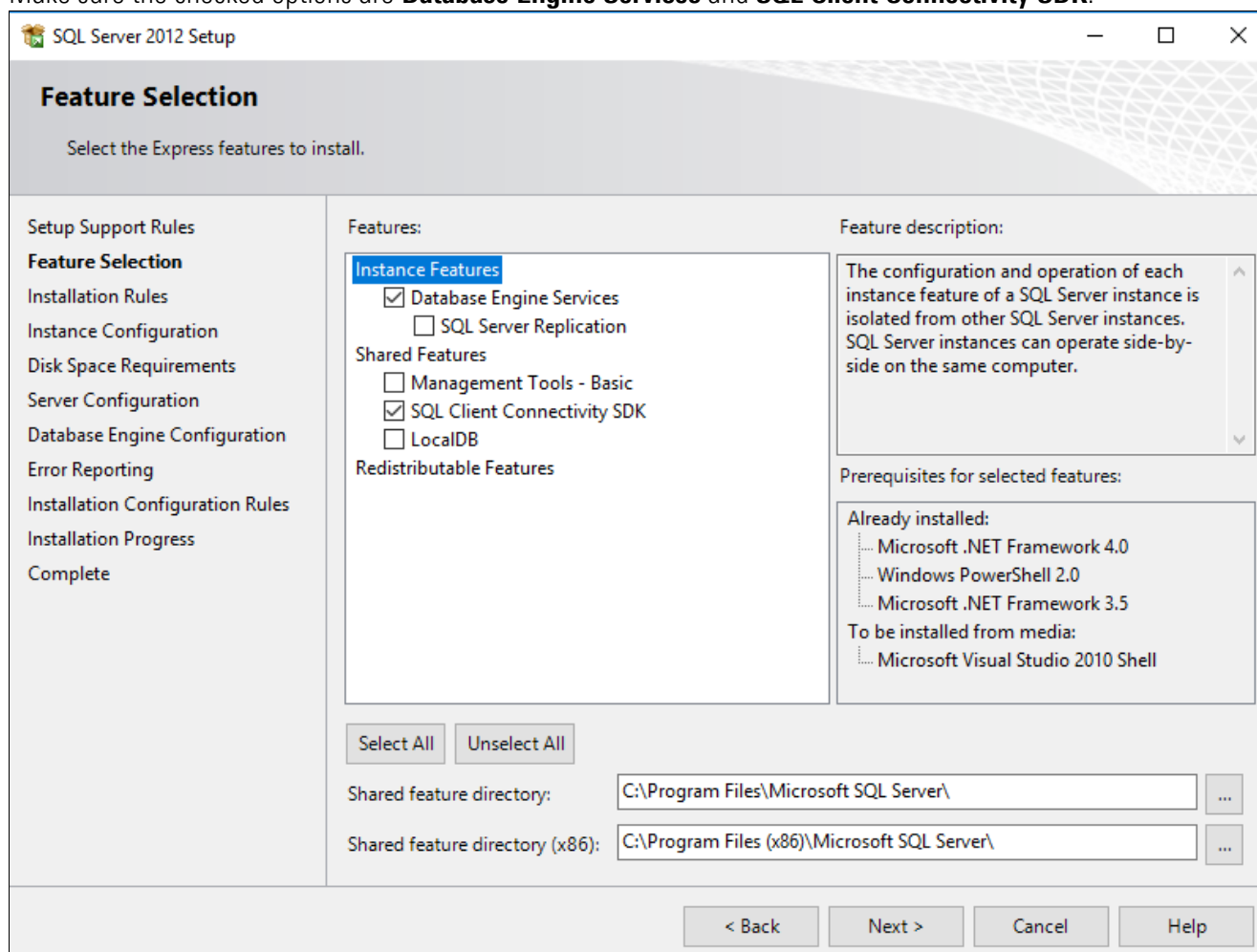
- **SOLEXPRT\_x86\_ENU.exe**  
The installation files for SQL Server Express that gives the option of running as a 32-bit and 64-bit application.
  - **SOLEXPRTWT\_x86\_ENU.exe**  
The installation files for SQL Server Express that gives the option of running as a 32-bit and 64-bit modes. This includes the SQL Server Management Studio Express Application.
  - **SOLEXPRT\_x64\_ENU.exe**  
The installation files for SQL Server Express that are for 64-bit systems only.
  - **SOLEXPRTWT\_x64\_ENU.exe**  
The installation files for SQL Server Express that are for 64-bit systems only and the application SQL Server Management Studio Express.
  - **SQLManagementStudio\_x64\_ENU.exe**  
The standalone application SQL Server Management Studio for 64-bit systems only.
  - **SQLManagementStudio\_x86\_ENU.exe**  
The standalone application SQL Server Management Studio for 64-bit with the option of 32-bit systems.
- Choose the download(s) you need.

## Run SQL Server Install

1. Run the installation file that had been downloaded from the Microsoft website. The installation wizard will launch where you will choose **New SQL Server stand-alone installation or add features to an existing installation**.



2. Accept the user agreement that prompts before the install begins. The application will run a system check.  
**Note:** The system may require a reboot when the scan is complete.
3. Make sure the checked options are **Database Engine Services** and **SQL Client Connectivity SDK**.



4. Check the **Named Instance** radio button and enter the name you wish use to identify your SQL server instance. In this example it is SQLEXPRESS\_DEMO.

**Instance Configuration**

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

Setup Support Rules  
Installation Type  
License Terms  
Feature Selection  
Installation Rules  
**Instance Configuration**  
Disk Space Requirements  
Server Configuration  
Database Engine Configuration  
Error Reporting  
Installation Configuration Rules  
Installation Progress  
Complete

☐ Default instance  
☒ Named instance:

Instance ID:

Instance root directory:  ...

SQL Server directory:

Installed instances:

Instance Name	Instance ID	Features	Edition	Version
SQLEXPRESS_DEMO	MSSQL11.SQLEXPRESS_DEMO	SQLEngine	Express	11.0.2100.60

< Back   Next >   Cancel   Help

5. Confirm that both of the SQL Services are set to **Automatic** (as shown below) to ensure the services are running after the installation.

**Server Configuration**

Specify the service accounts and collation configuration.

Setup Support Rules  
Feature Selection  
Installation Rules  
Instance Configuration  
Disk Space Requirements  
**Server Configuration**  
Database Engine Configuration  
Error Reporting  
Installation Configuration Rules  
Installation Progress  
Complete

Service Accounts   Collation

Microsoft recommends that you use a separate account for each SQL Server service.

Service	Account Name	Password	Startup Type
SQL Server Database Engine	NT Service\MSSQL\$SQL...		Automatic
SQL Server Browser	NT AUTHORITY\LOCAL ...		Automatic

6. Click **Next**.
7. Select the radio button labeled **Mixed Mode**. Enter a password in twice to create the password. This password is for the SQL username "sa". (Keep note of this information.)

**Database Engine Configuration**

Specify Database Engine authentication security mode, administrators and data directories.

Setup Support Rules  
Feature Selection  
Installation Rules  
Instance Configuration  
Disk Space Requirements  
Server Configuration  
**Database Engine Configuration**  
Error Reporting  
Installation Configuration Rules  
Installation Progress  
Complete

Server Configuration | Data Directories | User Instances | FILESTREAM

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode \_\_\_\_\_

☐ Windows authentication mode

☒ Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account. \_\_\_\_\_

Enter password: [Masked Password]

Confirm password: [Masked Password]

Specify SQL Server administrators \_\_\_\_\_

[Add Current User] [Add...] [Remove]

SQL Server administrators have unrestricted access to the Database Engine.

8. Click **Next** until the installation begins. The installation should run and complete automatically.

**Complete**

Your SQL Server 2012 installation completed successfully with product updates.

Setup Support Rules  
Feature Selection  
Installation Rules  
Instance Configuration  
Disk Space Requirements  
Server Configuration  
Database Engine Configuration

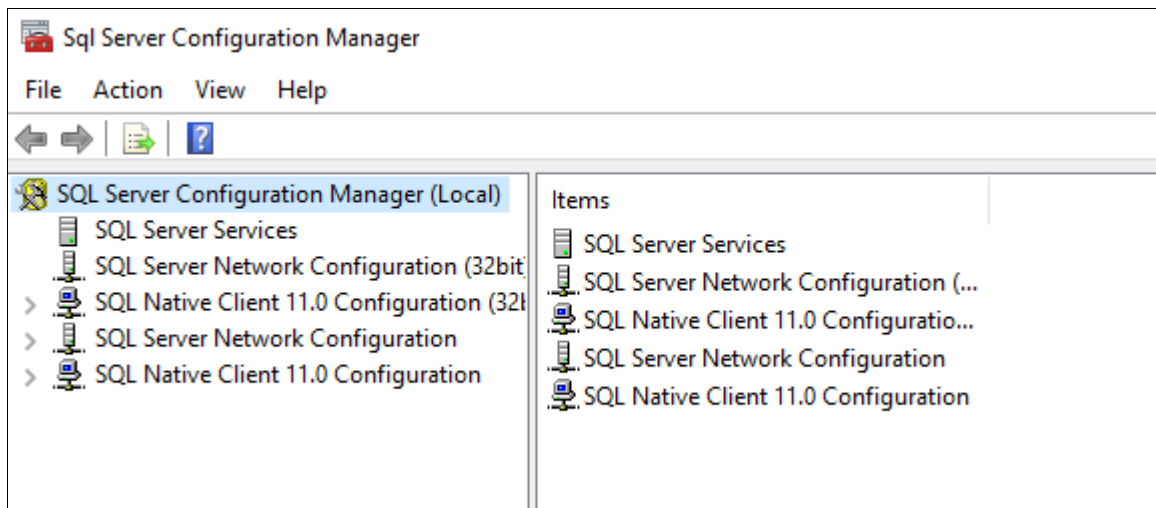
Information about the Setup operation or possible next steps:

Feature	Status
✓ Database Engine Services	Succeeded
✓ SQL Browser	Succeeded
✓ SQL Writer	Succeeded
✓ SQL Client Connectivity	Succeeded
✓ SQL Client Connectivity SDK	Succeeded

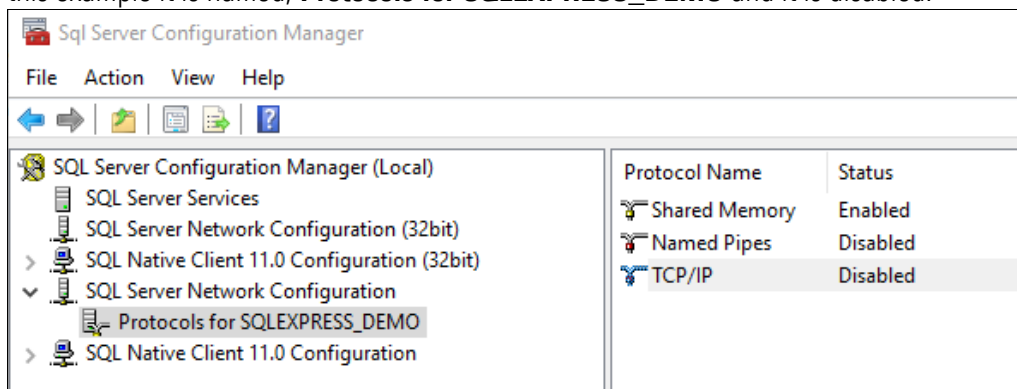
9. Once it is complete, click **Close** then confirm that the TCP/IP ports are enabled by using the following steps.

## Confirm TCP/IP Ports are Enabled

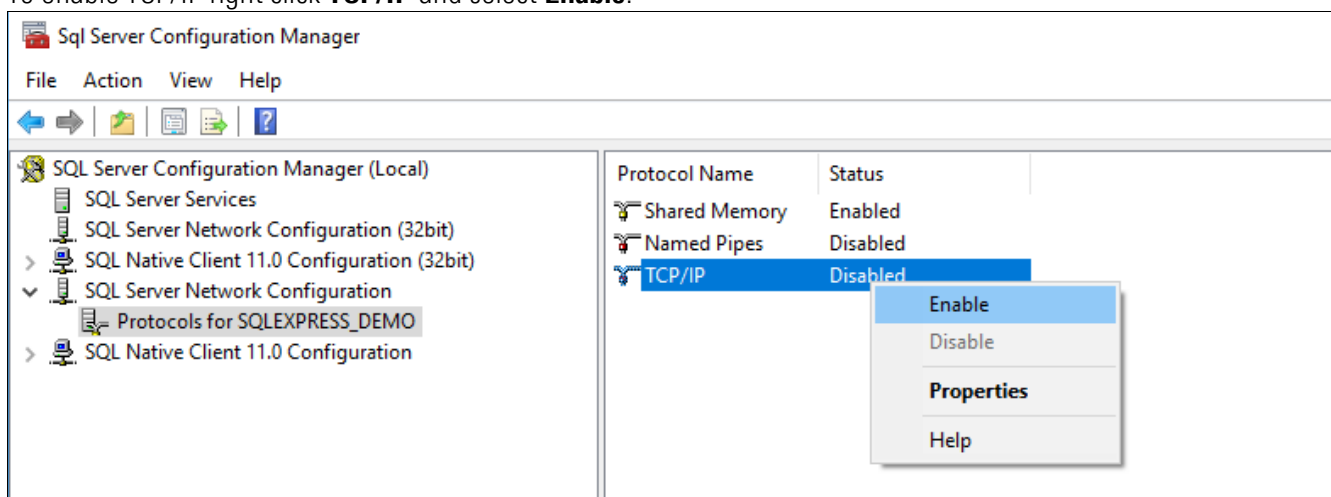
1. Launch the application named **SQL Server Configuration Manager**.



2. Highlight the protocols for the instance name which is located under the **SQL Server Network Configuration**. In this example it is named, **Protocols for SQLEXPRESS\_DEMO** and it is disabled.

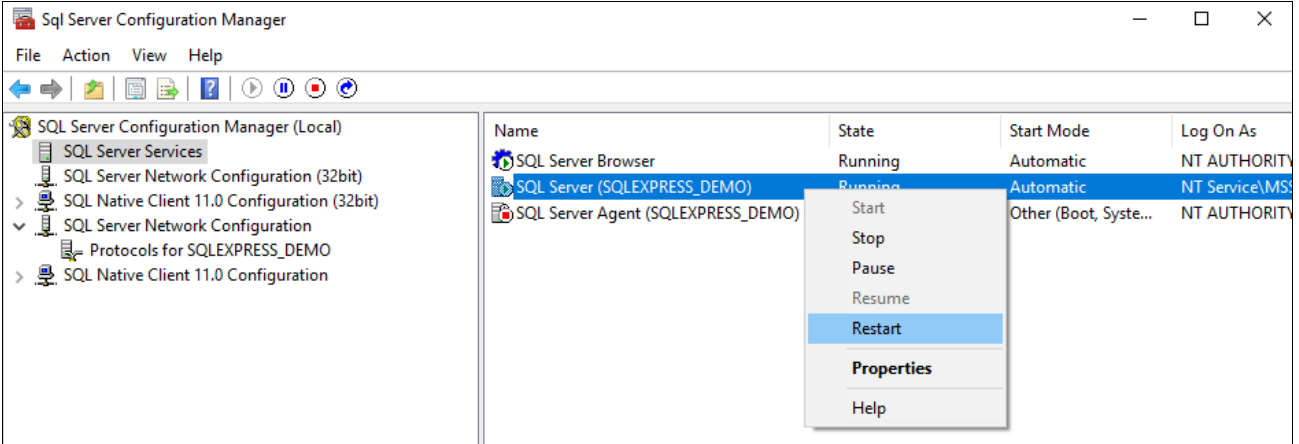


3. To enable TCP/IP right click **TCP/IP** and select **Enable**.





4. If you had to enable the TCP/IP port on your SQL Server service, you must restart it. Highlight the SQL Server Services, right click on the instance, and select **Restart**.

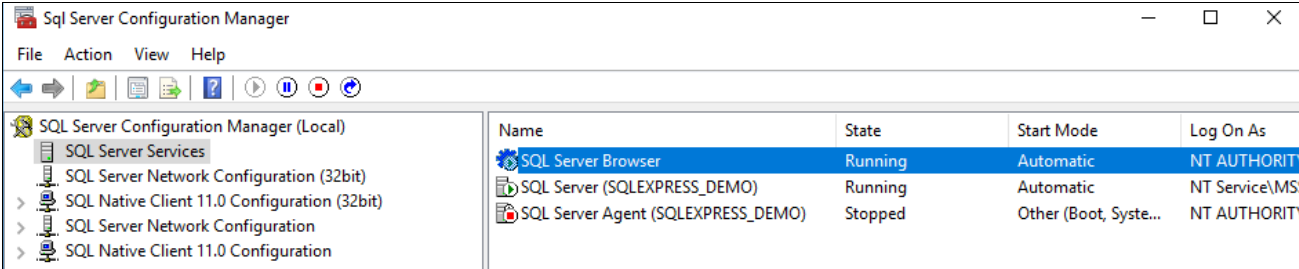


5. After it restarts, configure the SQLBrowser service to run automatically by using the following steps.

**Configure SQLBrowser Service of SQL Server to Run Automatically**

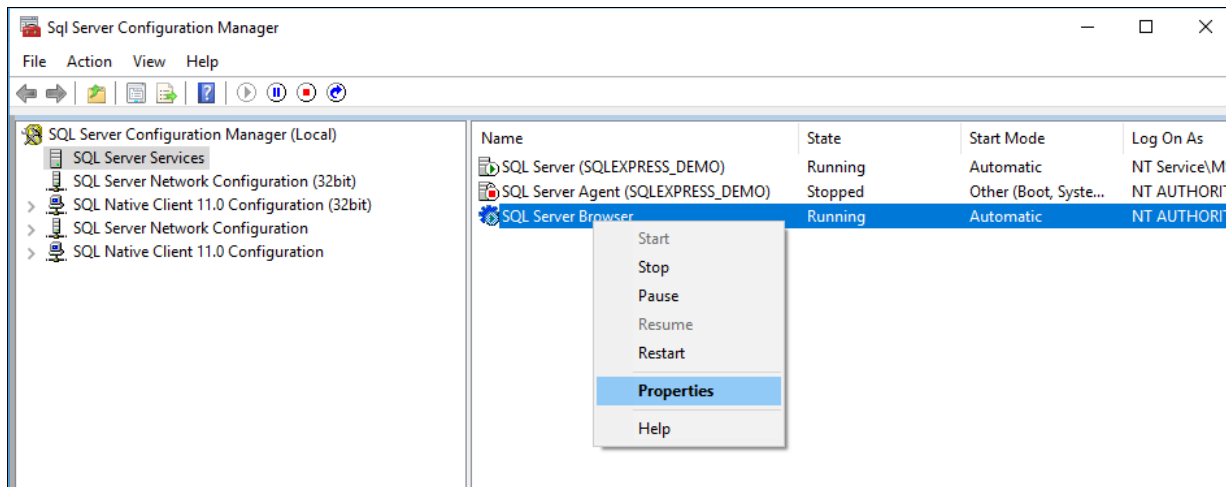
In addition to being configured to accept remote connections, the SQLBrowser service of SQL Server must also be configured to run automatically. As an example, to perform this configuration for SQL Server 2012 SP4, complete the following steps.

1. Open the **SQL Server Configuration Manager**. The path to this program in the Windows menu may differ slightly on different versions of Windows.
2. Click on the **SQL Server Services** node.

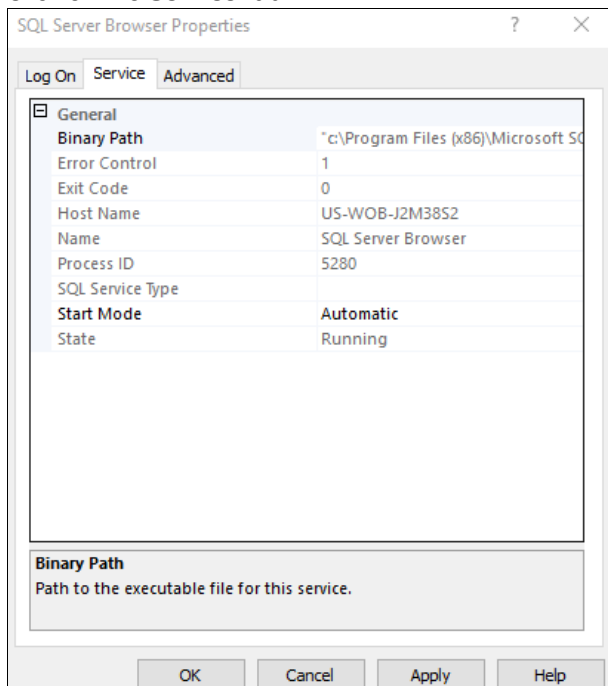


In this example, SQL SERVER Browser is set to start automatically already. If yours is not, use the following steps.

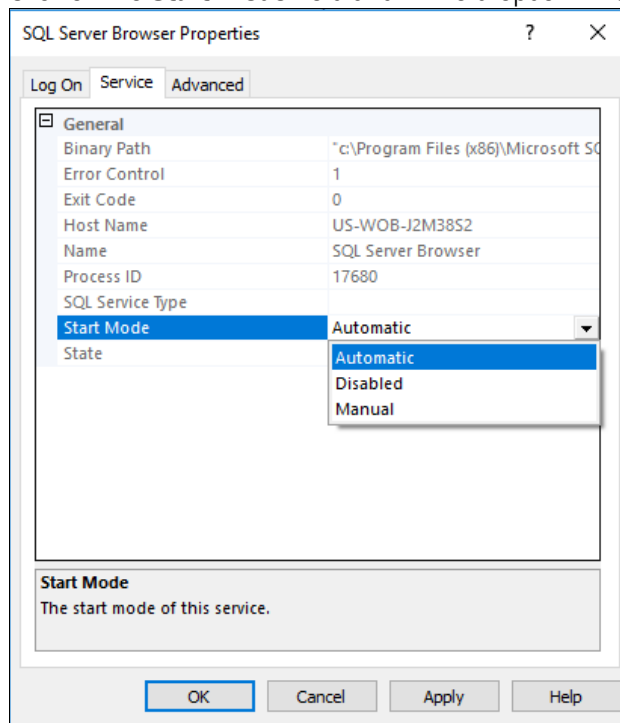
3. Right-click on the **SQL Server Browser** in the right side of the display and select **Properties** from the pop-up menu.



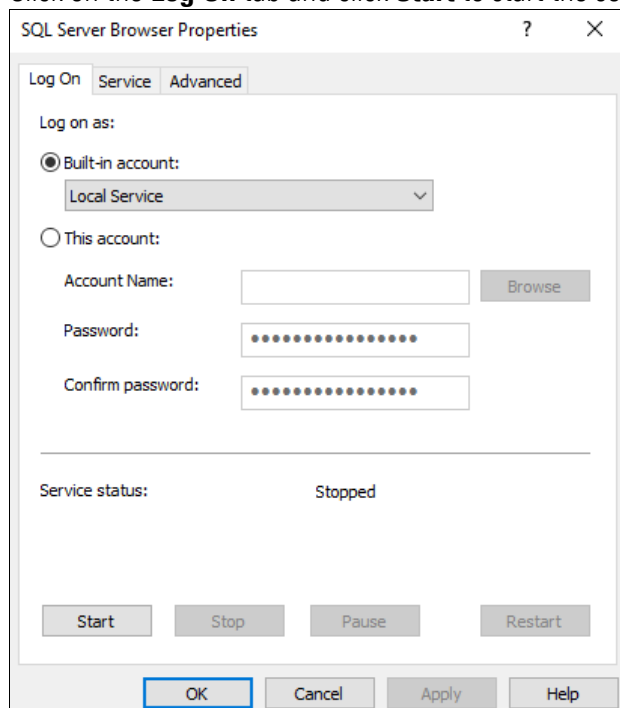
4. Click on the **Service** tab.



5. Click on the **Start Mode** field and in the dropdown field on the right set the value to **Automatic** and click **Apply**.



6. Click on the **Log On** tab and click **Start** to start the service, and click **OK**.



Now install Custodial Integrator.

## CI Computer Install or Upgrade

Perform the following installation by first logging into the CI Computer with a Windows login that is a member of the Administrators group on that computer and then completing all steps presented by the installation wizard.

**Note:** If you are upgrading CI, please review the release notes between the version you currently run and the most recent version to understand changes and actions you may need to take to complete the upgrade. For example, if this is an upgrade from a version prior to 3.5.002, you will be prompted to re-enter the username and password. In version 3.7.001, encryption options changed and instructions are in the Release Notes. Any version upgrading to 3.16.001 may require edits to the runCI.bat. Refer to the *Custodial Integrator Release Notes* at [http://www.byallaccounts.net/Manuals/Custodial\\_Integrator/axys/CI\\_releasenotes.pdf](http://www.byallaccounts.net/Manuals/Custodial_Integrator/axys/CI_releasenotes.pdf).

1. Open an Internet Explorer browser window and enter the CI Application installation URL:  
<https://www.byallaccounts.net/CI/>
2. Enter the distribution page user name and password when prompted. This is **not** the same user name and password you use to access the WebPortfolio application.
3. Click on the **Advent Axys®** link.
4. Click on the **Install Custodial Integrator Now** link.
5. Choose to **Run** the file now (if your browser provides this option) or **Save** the file to disk and then double-click on the saved file to run the installation.
6. If the wizard determines it needs to install Java on the CI Computer then it will present installation screens for this product now.
  - a. Review and accept the License Agreement.
  - b. The Java Runtime installation proceeds and finally the **Java Setup - Complete** screen displays. Click **Finish** to continue in the wizard.
7. The wizard will proceed to install the CI Application on the CI computer.
  - a. In **Welcome to the InstallShield Wizard for Custodial Integrator** click **Next**.
  - b. In **Setup Type**, the Complete option is the default and will install CI in the default location with all documentation. Click **Next**.
  - c. In the **Database** screen you will see that the default name for the CI database is BaaWpAci. If you choose to use a different name then change the text in the **Name** field. Click **Next**.
  - d. In **CI SQL Login Parameters** you will provide the SQL Server login id and password that the CI application will use to access its database. This login must minimally have the following SQL Server privileges for the database named in the previous dialog: public, db\_datareader, db\_datawriter. Enter the SQL login id and password now. Click **Next**.
  - e. In **SQL Server**, use the Browse... button to locate the SQL Server's computer/instance name. SQL Server will by default be named *hostname*/SQLExpress where *hostname* is the name of the computer – be sure to choose the correct instance. Under **Connect using** you may select **Windows authentication** if the Windows Login you are using has "sa" or "dbo" privileges in the target

database. If your Windows Login is not authorized in this way then enter a login id and password for a SQL user that has "sa" or "dbo" privileges. Click **Next**.

- f. In **Start Copying Files** review the settings. Click **Back** to go back and alter settings. Once the settings are correct click **Next**.
- g. The CI Application will be installed and the CI database will be created within the target Microsoft SQL Server. In **InstallShield Wizard Completed** click **Finish**.

### Configure Custodial Integrator

The file \Program Files\Custodial Integrator\CI.ini provides a starting point running CI. Determine which CI parameters you must use by reviewing the **CI Parameters** section on page 30.

1. Edit the \Program Files\Custodial Integrator\CI.ini file to add any required CI Parameters.
2. Run CI by double-clicking on the Desktop shortcut "Custodial Integrator". If the application does not come up or reports an error at startup, please check the **Troubleshoot Install Problems** section on page 27. If the problem continues to occur contact ByAllAccounts Technical Support for assistance.

After you run CI without error, the CI installation is complete and you can move on to configuration and installation verification. Next steps include verification of the basic configuration information you have provided so far (in CI.ini), providing CI additional configuration information required to access your financial information in WebPortfolio as well as to interact with your Axys installation, and troubleshooting any issues you encounter in your environment.

1. Create one test Portfolio (.cli) file in your Axys installation. You will need this portfolio to verify the CI installation. Contact your local Axys administrator if you need assistance with this step.
2. Run CI by double-clicking on the Desktop Shortcut.
3. The main view of CI displays, showing the steps to move data from WebPortfolio to Axys.
4. In step 1 in the user interface, click the **Setup...** button. CI now displays the CI Configuration dialog with the **Configuration** tab shown as follows:

**CI Configuration**

**Configuration** | Portfolio Code Translation | Security Translation

**CI Access**

CI login:

CI password:

CI working folder:

**Ays Configuration**

Ays executable folder (\$pathexe):

Ays user folder (\$pathtrn):

**Import/Export Defaults**

☒ Import Ays security data

☒ Include transactions for export

☒ Include transaction source information as a comment in Trade Blotter

☒ Include positions for export

☐ Post positions to Ays

☒ Include security prices for export

5. In the *CI Access* section, enter the following test credentials (note: these credentials give you access to test data only and should only be used to validate the CI configuration):

- a. CI login: CITEST
- b. CI password: WEEPINGWILLOW1

**Note:** This login and password give you access to test data only and should only be used to validate the CI configuration. Later, access your own data using the CI Login and CI Password assigned to you by ByAllAccounts.

CI working folder:

Create a working folder for CI to use for temporary files. Enter that folder name into this field or use the 'Browse...' button to select the folder within a Windows explorer.

6. Verify that CI can successfully use the CI test credentials to access data from WebPortfolio. Click the **Test Access** button to perform the test. The results of the Test Access are displayed in a popup dialog and at the bottom of the *CI Access* section.
7. In the *Axys Configuration* section, enter your Axys root installation folder in *Axys executable folder (\$pathexe)*. In the *Axys user folder (\$pathtrn)*, enter the Axys user folder of the person who will be using CI. If you have defined any of the following custom labels in your Axys installation, you should click the *Advanced...* button to specify additional Axys folders corresponding to those custom labels for CI:
  - a. Client folder (\$pathcli)
  - b. Information folder (\$pathinf)
  - c. Price folder (\$pathpri)
  - d. Log folder (\$pathlog)

Obtain the settings for these custom labels from your Axys administrator and enter the folders into the corresponding text field in CI in the *Advanced->Axys Folders* dialog.

8. In the *Import/Export Defaults* section, the following options are available to specify which types of files should be generated for Axys in a typical cycle. If an option is checked then that type of file will be generated during the export step unless you override this setting in the main view.
  - Import Axys security data
  - Include transactions for export
  - Include positions for export
  - Include security prices for export

You will want to minimally check the following in the *Import/Export Defaults*:

- a. Check *Import Axys security data* to import Axys Security Info and Security Type information from Axys each time you run CI to perform a data translation.
  - b. Check *Include transactions for export* to transfer Transactions from WebPortfolio to Axys.
  - c. Check *Include positions for export*
  - d. Check *Post positions to Axys* to post reconciliation positions
9. Click **Save** to save your configuration settings.
  10. Click the **Portfolio Code Translation** tab. CI will load the test accounts available and display them in the **Untranslated WebPortfolio accounts** section. Verify that you see accounts listed in this section. Define at least one Portfolio code translation. You can do this by selecting any one of the test accounts from the untranslated section, clicking **Add**, selecting your test portfolio from the **Axys portfolio codes** section, and clicking **Add** again.

11. Click **Close** to close the CI Configuration tab and return to CI's main view.
12. Verify that CI can load data from Axys by clicking **Import**. CI will invoke the Axys Import/Export utility to export the sec.inf and type.inf files for use in CI. CI will also download financial information for the selected WebPortfolio accounts into CI and attempt to match securities to those defined in Axys. The result of your import will most likely be an error message with text 'error validating securities' (meaning that CI could not locate some of the securities referenced by WebPortfolio in the Axys security master) and the appearance of the 'Security Translation...' button. If there are no security translation errors, the **Import** button will be grayed out and the **Export** button will be active. In either case, you have verified the CI installation and configuration.
13. Click **Setup** and go to the Portfolio Code Translation tab. Delete the one entry you created by selecting that row in the table at the top of the dialog and clicking **Delete**. Now navigate to the Configuration tab and remove the CI Access test credentials you entered. CI will later be configured using the actual CI credentials that provide access to your firm's financial information in WebPortfolio. Click **Save** to save the changes.
14. Close the Configuration dialog and exit CI.

### Consider password encryption models

As of version 3.7, CI supports two models for managing encrypted user logins and passwords: default and enhanced. The default model stores the encrypted login/password combinations in the database and requires no special instructions. The enhanced model uses an on-disk keystore model that uses the password-protected keystore file (ciksfile) to encrypt and decrypt user logins and passwords which are stored in the database in an encrypted state. That model is more complicated to set up, especially when multiple instances of CI access the same database, described in [Setup for Multiple CIs that access the same database](#).

**Note:** Do not remove the PBEKeysetPass or KSKeysetPass parameters. Do not change them for any reason without express guidance from ByAllAccounts Technical Support. If the value of either does need to be changed, then the value must be changed to match in all instances.

These parameters are described in [Database](#), page 40.

### Install multiple instances that access the same database

This section explains what you need to know if you want multiple instances of CI to access the same database.

Note: If you are upgrading CI, please review the release notes between the version you currently run and the most recent version to understand changes and actions you may need to take to complete the upgrade. For example, instructions regarding changes to encryption are included in the section for version 3.7.001. Refer the *Custodial Integrator Release Notes* at [http://www.byallaccounts.net/Manuals/Custodial\\_Integrator/axys/CI\\_releasenotes.pdf](http://www.byallaccounts.net/Manuals/Custodial_Integrator/axys/CI_releasenotes.pdf).

**Note:** Each instance must use the same user login and password and only one instance should run at any given time.



## Setup for Multiple CIs that access the same database

### Default encryption model

The default encryption model for user logins and passwords does not necessitate any special considerations when installing multiple instances of CI that access the same database. If you are using that model, you can skip the rest of this section.

### Enhanced encryption model

However, because the enhanced encryption model is more complicated, it does require special instructions as described here.

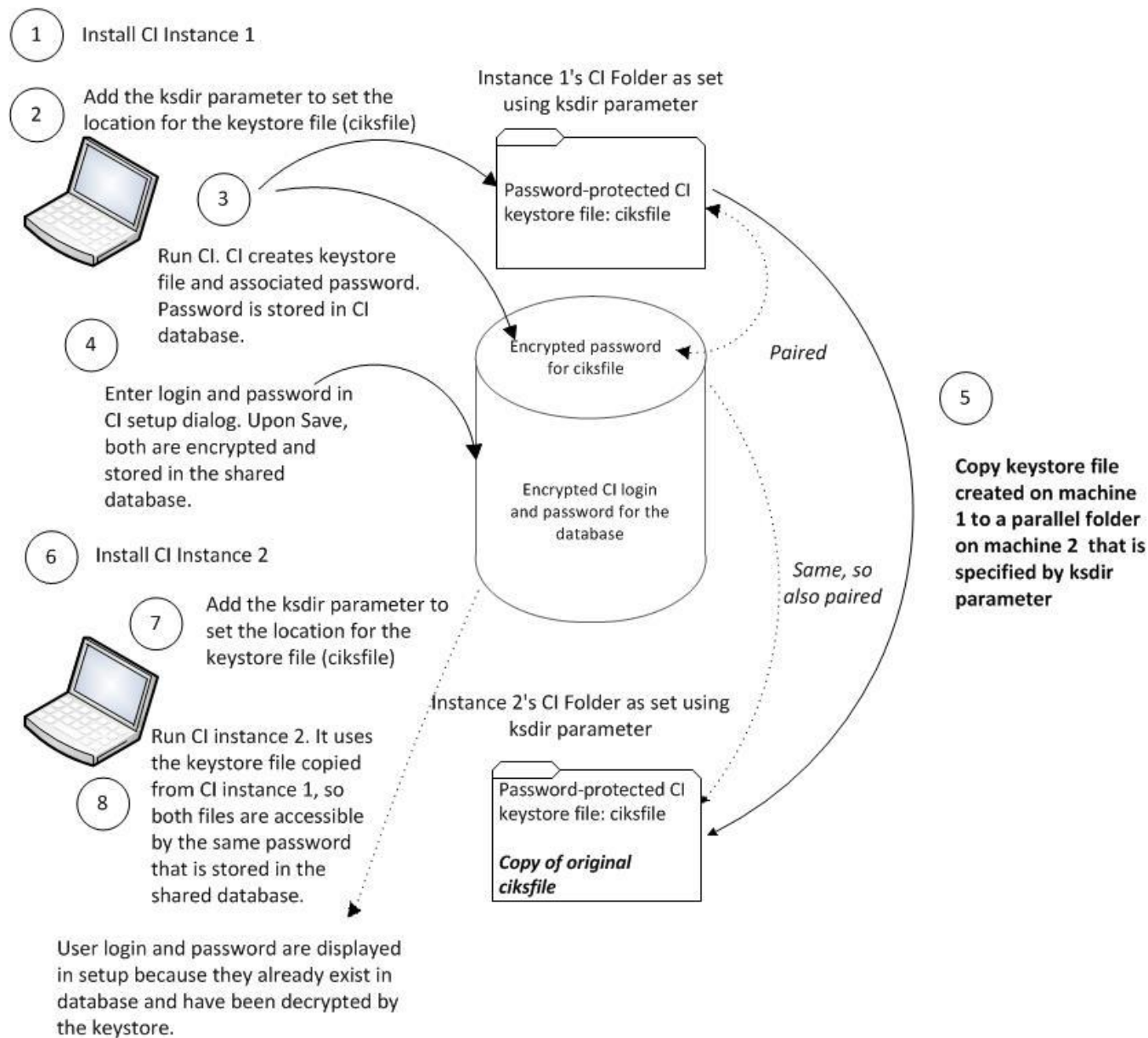
Note: If you are upgrading CI from version 3.15.001 or earlier, you may need to edit each runCI.bat instance as described in the release notes for 3.16.001. Refer to

[http://www.byallaccounts.net/Manuals/Custodial\\_Integrator/axys/CI\\_releasenotes.pdf](http://www.byallaccounts.net/Manuals/Custodial_Integrator/axys/CI_releasenotes.pdf).

The password for the keystore file is stored in the database and is uniquely paired to the keystore file, and each instance of CI that uses the same database must use the same pair. The diagram below illustrates the process to use to ensure that they use the same pair:

1. Install CI instance 1.
2. In the runCI.bat (or CI.ini depending on where the database name is set), set the ksdir parameter to the location for the keystore file (ciksfile).
3. Run CI instance 1 for the first time after install. That creates the keystore file and its associated password which is stored in the CI database. The keystore file is stored in the location that was defined using the ksdir parameter in step 2.
4. Enter the user login and password in the CI setup dialog. Upon saving, the user login and password are encrypted and stored in the shared database.
5. **Before** installing CI instance 2, copy the keystore file created on machine 1 to a parallel folder on machine 2.
6. Install CI instance 2 on machine 2.
7. As you did for CI instance 1, set the location for the keystore file using the ksdir parameter.
8. Run CI instance 2. CI instance uses the keystore file that was copied from CI instance 1, so both keystore files are accessible by the same password that is stored in the shared database.

User login and password are displayed in setup because they already exist in the database and have been decrypted by the keystore.



### Install multiple instances of CI on same machine with different databases

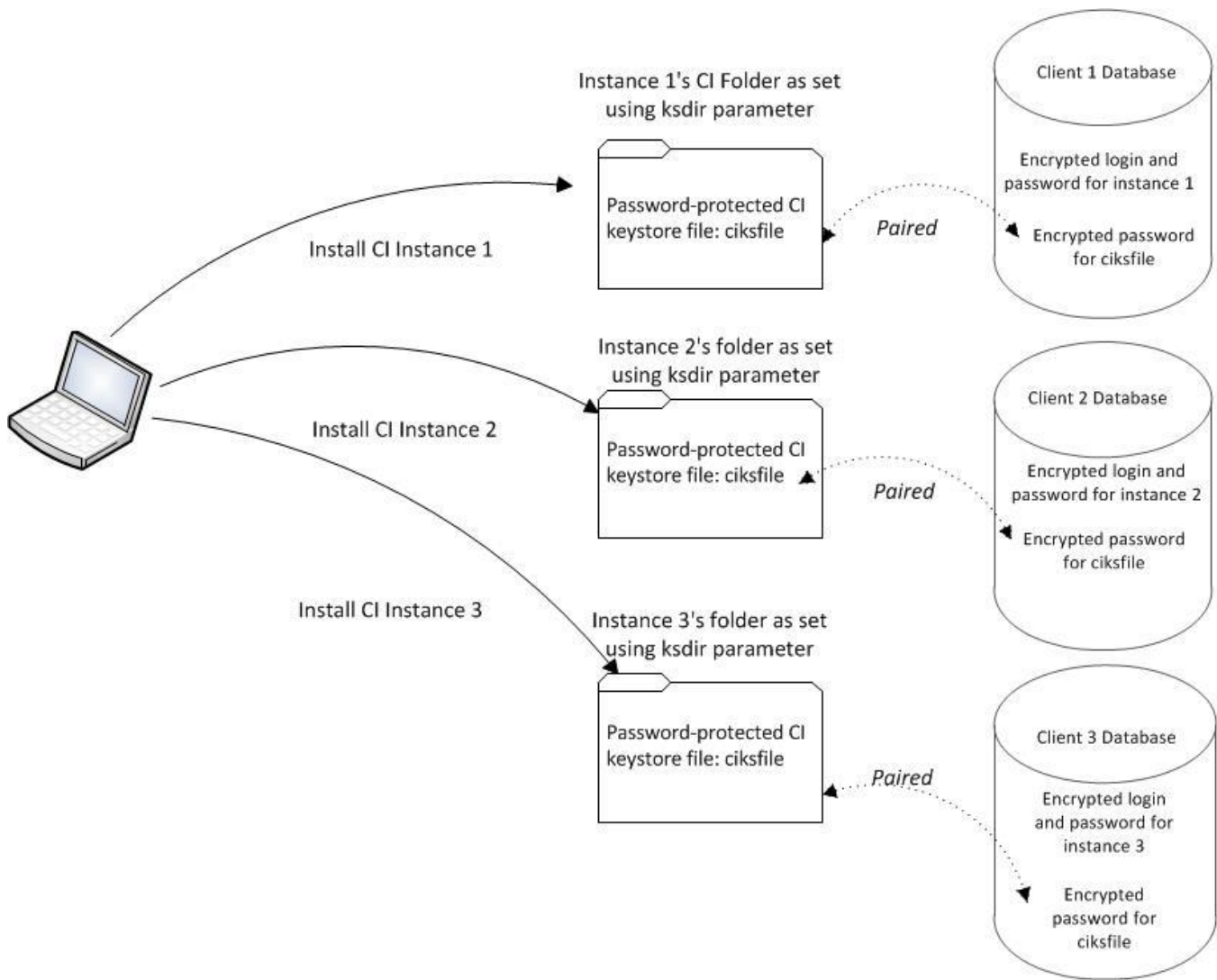
This section describes how to install multiple instances of CI on the same machine that access different databases. The instructions vary a bit depending on whether you use the default or enhanced encryption model for the logins and passwords. For information about the encryption methods, refer to [Consider password encryption models](#), page 23.

Use the setup shown in the following diagram when multiple instances of CI are installed on one machine and each instance accesses its own database.

Use the following procedure to get all CI instances working properly:

1. For each client create a separate *runci<client name>.bat* script that specifies the following parameters settings which must be unique for each client:
  - the client's database name (*dbname:*)
  - if you are using the enhanced encryption model, use the *ksdir* parameter set the location for the uniquely-paired keystore file (*ciksfle*). The password for the keystore file stored in the database is uniquely paired to the keystore file for the instance.
  - If you use bulk insert, specify the bulk insert parameters (*usebulkinsert:y* and *bulkinsertdatafolder:*)
  - If you are upgrading CI from version 3.15.001 or earlier, you must edit *runCI<clientname>.bat* as described in the release notes for 3.16.001. Refer to [http://www.byallaccounts.net/Manuals/Custodial\\_Integrator/axys/CI\\_releasenotes.pdf](http://www.byallaccounts.net/Manuals/Custodial_Integrator/axys/CI_releasenotes.pdf).
2. When setting up CI for each client also configure a separate location for their CI working and output folders.
3. Run CI for one of the customers.
4. Go to the **Setup > Configuration** tab. If this is a CI upgrade from a version prior to 3.5.002, you will be prompted to re-enter the username and password.
5. Make sure the CI working folder is set to the correct folder for the instance.
6. Repeat steps 3- 5 for the additional CI instances.
7. If you are using the enhanced method and used the *ksdir* parameter to set the location for the keystore file, the process should create a separate keystore file (filename: *ciksfle*) in the that folder for each customer, and should allow each instance of CI to run without problems.

Note: If you need to create new CI instances in the future, you'll need to create a separate working folder and INI folder for the new instance and you'll need to add the appropriate parameters to the *CI.ini* and *runCI.bat* files for each new instance.



## Troubleshoot Install Problems

### General Techniques

CI logs information about its activities to a log file. If CI encounters an error, the error should appear with full detail in the log file. CI log files can be found in the \Program Files\Custodial Integrator\log folder. CI log files have a unique, generated name. To obtain the CI log file associated with the most recent CI session, sort the CI log files by date in the Windows Explorer. CI will remove log files older than 14 days, so if you need to retain a log file for support purposes, copy the log file to another folder.

You can direct CI to provide additional information for troubleshooting. Use the **debugall=1** CI.ini parameter to direct CI to log more detailed information about what it is doing. This parameter is described in the **CI Parameters** section on page 30.

## Known Issues

This section describes common installation problems and suggestions for troubleshooting or working around these problems.

### General Errors

Symptom	Possible Cause	Action
Configuration <b>Test Access</b> reports 'Error loading WebPortfolio Data'	Incorrect CI proxy configuration for Internet access	Check the CI log file for more details. Contact ByAllAccounts Technical Support for assistance.
CryptoException error	Password encryption problem. CI database has a stored keystore password in the database and a keystore file exists in the kmdir, but they do not match.	<ol style="list-style-type: none"> <li>1. Close CI.</li> <li>2. Delete the ciksfle in the kmdir folder</li> <li>3. Restart CI</li> <li>4. Go to Setup and reenter your user id and password (you may be prompted to do that as soon as CI starts).</li> <li>5. Click on the "Test Access" button on the right of the screen to verify that the login and password are correct.</li> <li>6. Proceed as normal.</li> </ol> <p>If you have multiple CI instances on a single database and are using the enhanced encryption model, then after these steps you need to copy the ciksfle from the kmdir of the CI instance that was just recovered to each kmdir folder (one per instance) <u>before</u> running these other CI instances. See <a href="#">As of version 3.7, CI supports two models</a> for managing encrypted user logins and passwords: default and enhanced. The default model stores the encrypted login/password combinations in the database and requires no special instructions. The enhanced model uses an on-disk keystore model that uses the password-protected keystore file (ciksfle) to encrypt and decrypt user logins and passwords which are stored in the database in an encrypted state. That model is more complicated to set up, especially when multiple instances of CI access the same database, described in <a href="#">Setup for Multiple CIs that access the same database</a>.</p> <p><b>Note:</b> Do not remove the PBEKeysetPass or KSKeysetPass parameters. Do not change them for any reason without express guidance from ByAllAccounts Technical Support. If the value of either does need to be changed, then the value must be changed to match in all instances. These parameters are described in <a href="#">Database</a>, page 40. Install multiple instances that access the same database (page <a href="#">23</a>) for more information.</p>

**Database Errors**

In the case of a database error, CI will generally report one of the following messages:

- An error occurred when trying to connect to the database.
- An internal error occurred.

Additional details may also be presented in the popup error box. For all errors of this type check the CI log file for more detailed information.

Symptom	Possible Cause	Action
Error connecting to SQL Server with error message "Login failed for sa user - not associated with a trusted SQL Server connection".	SQL Server not configured for mixed-mode authentication.	Check with your SQL Server administrator.
[Microsoft][SQLServer Driver for JDBC] Error establishing socket.	SQL Server not configured for TCP/IP OR insufficient resources available on the SQL Server computer to allocate a new socket.	Check with your SQL Server administrator.

## CUSTODIAL INTEGRATOR CONFIGURATION REFERENCE

### CI Parameters

You can configure the CI Application by adding runtime parameters to the CI.ini file. The basic form of the parameters contained by CI.ini is:

**parameter name**=value

where parameter name is one of the parameters described in the following sections and “=” separates the parameter from its value, and “value” is the value you supply for the parameter. Parameters are separated from each other by a linefeed. Parameters do not need to be listed in any particular order.

The following example defines your Axys system currency as “eu” and your SQL Server password as “mypassword”:

**axyscur**=eu

**sqlpw**=mypassword

**CI Customizations**

These parameters customize the behavior of your CI installation.

Parameter label	Parameter value(s)	Default	Description	Example
<b>defaultAccountIdentifier</b>	WPAccountNumber WPInternalID WPAccountName	none	Used to set the default account identifier for untranslated accounts to one of three values:  WPAccountNumber sets it to the Account Number from WebPortfolio.  WPAccountName sets it to the account name from WebPortfolio.  WPInternalID sets it to the internal ID from WebPortfolio.  You should consistently use the same source for the default account identifier.	



<b>inputFolder</b>	(full path or a path relative to the CI startup folder )	(CI startup folder)	Used to specify the location of custom transaction translations, holding filters, and transaction filters files.  If the input folder is specified and the folder does not exist, or the user does not have access to it, an error message is displayed to the user and the application exits.	For example, if CI startup directory is C:\CI, and the custom translations folder is C:\CI\data\customtranslations, then this parameter would be set to:  inputFolder=C:\CI\data\customtranslations
<b>posstmt</b>	y n	n	If value is 'y' then CI will generate text position statements in the program files\custodial integrator\exported\pos folder whenever positions are included in the export process. One position statement is generated for each Axys portfolio.	posstmt=y
<b>posposdefdir</b>	{folder}		This parameter relates to position posting. Use only when your Axys installation is configured in such a way that the pospos32.exe utility does NOT look in the current working directory for the ptopost.trn.	posposdefdir=c:\axys

<b>txfile</b>	{filename}	topost.trn	Use this parameter if you want to import transactions into a REXX Trade Blotter rather than the Axys Trade Blotter.	txfile=rtopost.trn
<b>ignorePositionalExtServLevelAccounts</b>	y n	n	<p>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.</p> <p><b>Important note for those upgrading from any release prior to CI 3.15:</b> Refer to your release notes for 3.15 for important upgrade information.</p>	ignorePositionalExtServLevelAccounts=y

**Axys Customizations**

These parameters provide information about your Axys configuration or about how you would like data translated for Axys.

Parameter label	Parameter value(s)	Default	Description	Example
<b>axysacbond</b>		f	The letter code in your Axys configuration for the asset class 'Fixed Income'. Older versions of Axys may use a letter that is different from 'f', which can adversely affect CI's data translation for Axys.	axysacbond=i
<b>axysaccash</b>		c	The letter code in your Axys configuration for the asset class 'Cash'. Older versions of Axys may use a letter that is different from 'c' and this can adversely affect CI's data translation for Axys.	axysaccash=s
<b>axysbrok</b>		\$brok	The value to use for the Broker column when a value is required for this column of a transaction (e.g. when Commission has a value, the Broker column must also be valued). The value must not be an empty string.	axysbrok=dummyb

## Axys customizations (continued)

Parameter label	Parameter value(s)	Default	Description	Example
<b>axyscommission</b>	y n	Y	If 'y' then if commission value is available for a transaction, both the Commission and Broker columns are filled in for the transaction. \$brok is used for the Broker value (see <b>axysbrok</b> parameter). If 'n' then the Axys Commission and Broker columns are not filled in, regardless of the information available for a transaction.	Axyscommission=n
<b>axyscur</b>		us	The currency code you have defined as your Axys system currency.	Axyscur=eu
<b>axyslotlocation</b>		253	The value to use for the lot location column.	axyslotlocation=253
<b>defdivquan</b>			For Dividend transactions with no reported Quantity, use this value in the quantity column in the topost.trn transaction file.	Defdivquan=1
<b>defmarkmarket</b>	y n	n	Transactions and positions against a currency other than your Axys system currency require a value for the Mark to Market field. This parameter defines the value to use for the Mark to Market field when this field is required.	Defmarkmarket=y
<b>defperfcw</b>	y n	n	Value to use in the Perf/CW column of the Axys topost.trn file.	Defperfcw=y

## Axys customizations (continued)

Parameter label	Parameter value(s)	Default	Description	Example
<b>fasectype</b>	Axys security type		Gives the Axys security type that you use to identify Fixed Annuities. This causes the following special treatment for these positions in the position file: if no units or price are given by the Financial Institution then Quantity is set to 1 and Price is set to the position's market value. If units are given by the Financial Institution but no price (or a \$0 price) then the price is computed by dividing market value by the units.	Fasectype=vfus
<b>defFeeType</b>	Axys security type	epus	The default Axys Security Type to be used when translating "fee"* transactions.	DefFeeType=epus
<b>defFeeSymbol</b>	Axys security symbol	custfee	The default Axys Security Symbol to be used when translating "fee"* transactions.	DefFeeSymbol=custfee

## Axys customizations (continued)

Parameter label	Parameter value(s)	Default	Description	Example
<b>overwriteposzerounits</b>	useMV use1 none	useMV	Used to handle positions where neither quantity nor price is reported by the financial institution. If not specified or useMV is specified then the quantity is set to the market value of the position and the price is set to \$1 (or 100 for fixed income). If use1 is specified then set the quantity to 1 and the price equal to the market value. If none is specified then do not report any quantity for the position in either the reconciliation file or the position file.	Overwriteposzerounits =useMV
<b>overwriteprice</b>	y n	n	CI only provides prices for securities that do not already have a price in the target price file. Use this option to cause CI to provide a price for all securities that it processes and to overwrite any price already present in the target Axys price file for these securities.	overwriteprice=y

## Axys customizations (continued)

Parameter label	Parameter value(s)	Default	Description	Example
<b>taxSymbol</b>		taxwith	The Axys Security Symbol for use in general tax transactions.	taxSymbol=tax1
<b>foreignTaxSymbol</b>		fortax	The Axys Security Symbol for use in foreign tax transactions.	ForeignTaxSymbol=frtax
<b>xlateFeeDesc&lt;n&gt;**</b>			This parameter specifies the string that should be compared to the "fee"* transaction's description. If the description contains this string then the transaction is considered to "match" this translation and corresponding xlateFeeType and xlateFeeSymbol parameter values are used.	xlateFeeDesc1=Tax
<b>xlateFeeType&lt;n&gt;**</b>		epus	This parameter specifies the Axys Security Type to be used for "fee"* transactions that match the corresponding xlateFeeDesc parameter.	xlateFeeType1=exus
<b>xlateFeeSymbol&lt;n&gt;**</b>		custfee	This parameter specifies the Axys Security Symbol to be used for "fee"* transactions that match the corresponding xlateFeeDesc parameter.	xlateFeeSymbol1=with

\*CI defines a "fee" transaction as any one of the following:

- A transaction with a WebPortfolio transaction type of: Fee, Investment Expense, or Service Charge.
- A non-cash security transaction with a WebPortfolio transaction type of: Debit transaction that has been determined to translate to an Axys "lo" type transaction. This covers various cases of expenses and tax deductions that might need special security type and symbol translations.
- A "dp" or "wd" transaction generated during CI processing of any type of WebPortfolio transaction. Such transactions are typically generated while processing a transaction that implies that a "fee" also occurred,

like a Sell transaction that really represents a sell of a fund to pay a fee and so will be generated for Axys as a "sl" followed by a "dp" (fee).

\*\* CI users can specify that different Axys Security Type and/or Axys Security Symbols be used for "fee" transactions whose description contains specific text. This is accomplished via the three parameters xlateFeeDesc, xlateFeeType, and xlateFeeSymbol. These parameters may appear either as a pair (xlateFeeDesc along with either xlateFeeType or xlateFeeSymbol) or as a trio (all parameters present) in the CI.INI file. The parameters are correlated by the trailing number in their name. An unlimited number of correlated parameters can be specified (e.g. xlateFeeDesc2, xlateFeeType2, etc.). Translations are applied in the order that they appear in the file. Once a match is made only that translation is used. If no translation matches then any default parameters (defFeeType, defFeeSymbol) are used.

Examples:

1. Customize translation for "RECORDKEEPING" fee transactions at all Financial Institutions to "epus" "expense":

```
xlateFeeDesc1=RECORDKEEPING FEE
```

```
xlateFeeType1=epus
```

```
xlateFeeSymbol1=expense
```

2. Customize translation for "RECORDKEEPING" fee transactions at all Financial Institutions to "epus" "expense" and also customize translation for "TAX":

```
xlateFeeDesc1=RECORDKEEPING FEE
```

```
xlateFeeType1=epus
```

```
xlateFeeSymbol1=expense
```

```
xlateFeeDesc2=TAX
```

```
xlateFeeType2=epus
```

```
xlateFeeSymbol2=with
```

## Corporate Firewall

If the CI Computer must access the Internet through a proxy, you will need to configure your proxy settings for CI using one or more of the following parameters:

Parameter label	Default	Description	Example
<b>proxyhost</b>		The name or IP address of the proxy host for Internet access.	proxyhost=comp1
<b>proxyport</b>	443	The proxy port for https Internet access.	proxyport=123
<b>proxyuser</b>		Username for Basic authentication at the <b>proxyhost</b> .	proxyuser=someu
<b>proxypw</b>		Password for <b>proxyuser</b> .	proxypw=mypw



## Database

The following command line parameters can be used to control CI's database access.

Parameter label	Default	Description	Example
<b>sqlhost</b>	localhost	The name or IP address of the SQL Server computer serving the CI database.	sqlhost=smac
<b>sqllogin</b>	Sa	Login to use to access SQL Server using SQL Server authentication.	sqllogin=mylogin
<b>sqlpw</b>	applesandoranges	Password for "sa" login, or for <b>sqllogin</b> (if specified).	sqlpw=myspw
<b>dbname</b>	BaaWpAci	Name of CI database.	dbname=mydb
<b>ksdir</b>	(CI working folder)	<p>Optional CI.ini parameter, only used for enhanced encryption model of user login and password.</p> <p>Specifies the location of the folder where the keystore file (ciksfile) resides. This file contains the encryption/decryption key for the user credentials for the enhanced encryption model. By default, the keystore file is created in CI working folder.</p> <p>For information about how to use this parameter when installing or upgrading multiple CI instances that access the same database see page <a href="#">23</a>. For installing or upgrading multiple instances of CI on same machine with different databases, see page <a href="#">26</a>.</p>	ksdir=<full path of keystore folder>
<b>PBEKeysetPass</b>	(PBE encryption key)	<p>This is a mandatory parameter. It provides a password used in the encryption of select sensitive data stored in the CI database.</p> <p>This parameter is automatically set during CI installation or CI upgrade from versions earlier than 3.12.</p> <p><b>Note:</b> This parameter must NOT be removed, and its value should NOT be changed without guidance from ByAllAccounts Technical Support. For setups with multiple CI instances see page <a href="#">23</a>.</p>	PBEKeysetPass=<fixed string value>

<b>KSKeysetPass</b>	(PBE encryption key)	<p>This is a mandatory parameter. It provides a password used in the encryption of select sensitive data stored in the CI database.</p> <p>This parameter is automatically set during CI installation or CI upgrade from versions earlier than 3.12.</p> <p><b>Note:</b> This parameter must NOT be removed, and its value should NOT be changed without guidance from ByAllAccounts Technical Support. For setups with multiple CI instances see page <a href="#">23</a>.</p>	KSKeysetPass=<fixed string value>
---------------------	----------------------	--	-----------------------------------

### Debugging

The following parameter is used to control the debugging capabilities of CI.

Parameter label	Parameter value(s)	Default	Description	Example
<b>debugall</b>	1 0	0	Use a value of 1 to enable all CI debugging features, including enhanced error and event logging.	debugall=1