Installation Guide

Sybase CEP Option R4

DOCUMENT ID: DC01259-01-0400-04

LAST REVISED: April 2011

Sybase, Inc., One Sybase Drive, Dublin, CA 94568

Contents

Preface

This document contains instructions for installing, configuring and administering Sybase® CEP Engine software on Microsoft Windows and UNIX-like operating systems. In order to access Sybase CEP Engine documentation electronically, you require a PDF reader on your system.

Installing Sybase CEP Engine

Before You Install

Before installing Sybase CEP Engine, determine which operating system each of the components needs to be installed on and the system requirements for the installation.

Supported Operating Systems

A description of operating systems supported by Sybase CEP.

Platform	Supported Operating Systems	Compiler	JDK Version
Linux-64 (AMD/ Intel)	Red Hat 5.0 (AMD) Red Hat 5.0 (Intel) SUSE 10 (AMD)	gcc 3.4.6	5.0 Update 12
Sun-64 (Sparc)	Solaris 10	gcc 3.4.6	5.0 Update 12
Sun-64 (Ultra-Sparc T2)	Solaris 10	gcc 3.4.6	5.0 Update 12
Sun-64 (AMD)	Solaris 10	gcc 3.4.6	5.0 Update 12
Windows (32,64) (Server/Adapters)	Windows 2003 Server (64-bit) XP Professional (32-bit and 64-bit)	MSDEV 2005 SP1	5.0 Update 12
Windows (32-bit) (Studio only)	Windows 2003 Server XP Professional Vista	MSDEV 2005 SP1	5.0 Update 12

See *Sybase CEP Engine Third-Party Software Dependencies* on page 137 for more information about supported operating systems.

Determine Which Components to Install

An introduction to the major components of Sybase CEP–the Sybase CEP Server and Sybase CEP Studio.

- Sybase CEP Server: This required component runs your Sybase CEP Engine applications.
- Sybase CEP Studio: This important component is a visual development tool that allows you to develop and test Sybase CEP Engine projects and the CCL query modules they contain. Sybase CEP Studio is required, unless you use another tool that compiles, registers, loads, starts, and stops the execution of projects.

Either of the components can be installed or upgraded separately. Notethat any Sybase CEP components that communicate with one another need to be of the same version.

Determine Where to Install Each Component

You can choose whether to install all Sybase CEP Engine components on the same computer, or distribute them among several computers.

If you are installing Sybase CEP Engine for development and testing purposes, you may find it easier to install both Sybase CEP Server and Sybase CEP Studio on a single computer. In a production environment, where you need to maximize performance, however, you may prefer to install Sybase CEP Server on a dedicated machine and to install Sybase CEP Studio on another machine.

If you are upgrading or reinstalling Sybase CEP Engine, determine whether you will install the new version in the same directory as the current version, or in a new directory. Note that installing a new version in the same directory as the older version overwrites the older version.

System Requirements

This section outlines how much disk space and memory is required to install Sybase CEP Engine and provides information on the required libraries and third-party components for specific platforms.

Disk Space

To install Sybase CEP Engine, enough disk space is needed for both the downloaded installation files, and the program files created after installation. If you are installing both Sybase CEP Engine components on the same machine, you need enough space for the installation and program files for both components.

The following table lists disk space requirements for Sybase CEP Engine, taking into account both the unzipped installation files and the post-installation program files. Note that the size requirements in this section are approximate: Sybase CEP Engine installation file sizes may change from one version to another, and different UNIX-like operating systems may have slightly different file sizes.

	Sybase CEP Server	Sybase CEP Studio
Microsoft Windows	240 MB	290 MB
UNIX-like Operating Systems	2890 MB	2780 MB

These disk requirements do not take into account the sizes of your Sybase CEP Engine project-related files.

RAM

Memory requirements vary depending on the number and complexity of queries in your application and on whether or not Sybase CEP Server and Sybase CEP Studio are installed on the same machine or on different machines. In general, Sybase recommends the following minimum RAM allocation. Note that the RAM recommendations shown here are for Sybase CEP Engine requirements only, not the entire RAM requirements for your computer.

	Sybase CEP Server	Sybase CEP Studio
Microsoft Windows	50 MB	50 MB
UNIX-like Operating Systems	100 MB	100 MB

Updating Libraries on Linux

Ensure you have the necessary system libraries and third-party components before running Sybase CEP Server and Sybase CEP Studio on Linux.

The required libraries and third-party components reside in the following packages. If any of the listed packages is not on your system, install it before running Sybase CEP Server or Sybase CEP Studio:

Sybase CEP Server

- alsa-lib-1.0.6-5.RHEL4
- audit-libs-1.0.14-1.EL4
- glibc-2.3.4-2.25
- openssl-0.9.7a-43.10
- pam-0.77-66.17
- unixODBC-2.2.11-1.RHEL4.1
- xorg-x11-libs-6.8.2-1.EL.13.36

Sybase CEP Studio

- alsa-lib-1.0.6-5.RHEL4
- atk-1.8.0-2
- expat-1.95.7-4
- fontconfig-2.2.3-7
- freetype-2.1.9-1

- glib2-2.4.7-1
- glibc-2.3.4-2.25
- gtk2-2.4.13-19
- pango-1.6.0-9
- xorg-x11-libs-6.8.2-1.EL.13.36

Updating Libraries on Solaris

Ensure you have the necessary system libraries and third-party components before running Sybase CEP Studio on Solaris.

The required libraries and third-party components reside in the following packages. If any of the listed packages is not on your system, install it before running Sybase CEP Studio:

- SUNWcsl
- SUNWfontconfig
- SUNWgnome-base-libs
- SUNWlexpt
- SUNWmlib
- SUNWxwice
- SUNWxwrtl
- SMCliconv

Sybase CEP Studio and Sybase CEP Server Compatibility

An outline of the compatibility of the Sybase CEP Studio R4 and Sybase CEP Server Sybase CEP Server R4 with each other, and with previous versions of the product (Sybase CEP Studio R3 and Sybase CEP Server R3).

	Sybase CEP Studio R4	Sybase CEP Server R4
Sybase CEP Studio R4	-	Compatible
Sybase CEP Server R4	Compatible	-
Sybase CEP Studio R3	-	Not compatible
Sybase CEP Server R3	Not compatible	-

Installing Sybase CEP Engine on Microsoft Windows

Learn how to install, reinstall, upgrade, or uninstall Sybase CEP Engine on Microsoft Windows.

Backing up Important Information on Windows: Upgrades and Reinstalls Only

If you are upgrading to a newer version, or are reinstalling the current version of Sybase CEP Server or Sybase CEP Studio, first back up the information in your adapters base folder or repository.

By default, both folders are called `SybaseC8Repository`.

If you plan to reinstall or upgrade in the same folder as the existing installation, make sure that the information you back up includes the following:

- Any custom adapter `.adl` files.
- Any user-defined function `.udf` files.
- Any custom library `.dll` files.

Installing Sybase CEP Server on Windows

There are two methods for installing Sybase CEP Server on Windows platforms.

The standard installation procedure, described in *"Standard Sybase CEP Server Installation"* on page 7, includes a panel that prompts you to specify a number of Sybase CEP Server configuration settings. The silent Sybase CEP Server installation method, described in *"Silent Sybase CEP Server Installation"* on page 11, allows you to specify these settings in advance and run the script "silently", that is, without prompts. Sybase recommends this method for installing multiple Sybase CEP Server instances, if they do not require highly customized settings.

If you are upgrading or reinstalling Sybase CEP Engine using the standard installation method, the default installation settings reflect your previous selections. Otherwise, the upgrade process behaves similarly to the initial install process.

Standard Sybase CEP Server Installation on Windows

A description of how to install the Standard Sybase CEP Server on Windows.

1. Get the appropriate Sybase CEP Server installation file.

   Sybase CEP Server installation files for Windows platforms look like this: `c8-server-`*version- operating_system* `_vc8-20100118.1-release.exe` where *version* is a three-part number, separated by dots, which indicates the version of Sybase CEP Engine, and *operating_system* is the name of the Windows platform on which you are installing Sybase CEP Engine.

   If you are installing Sybase CEP Server on Microsoft Windows Vista, make sure that you are logged in as the administrator.

2. Double-click the installation file to start installation.

   The Welcome panel of the Sybase CEP Server Installer opens. The following steps lead you through the Installer workflow. Options for cancelling install and navigating to previous panels in the workflow are available on every panel before actual installation takes place.

3. Click **Next** to continue the installation process.

4. Select the directory where you want to install Sybase CEP Server and click **Next** . The default location is `C:\SybaseC8` on all Windows platforms.

   Note: If you are installing Sybase CEP Server on Microsoft Windows Vista, specify a location for the data files in the **Destination Data Folder** box on the following panel. For Microsoft Windows Vista, the default location is C:\ProgramData\SybaseC8. This location is where the logs directory is installed and is not the same as the program files installation directory. In Windows 2003 or XP, the Program Files installation directory and ProgramData installation directory are the same by default.

   If you are upgrading or reinstalling Sybase CEP Server and specify the installation location to be the same directory as an existing installation, a pop-up message indicates that the Sybase CEP Server is already installed in the folder.

   Click **Yes** to uninstall the previous installation or **No** to upgrade without uninstalling. If you click **Yes**, the next panel confirms that the previous version of Sybase CEP Server will be uninstalled if you continue:

   1. Click **Next** to continue.

      The next panels display the uninstall process.

   2. A final install panel opens when uninstall is complete. Click **Next** to continue with the upgrade installation.

5. Select either to install a licensed copy of Sybase CEP Server or to evaluate a trial version of Sybase CEP Server.

6. The End-User License Agreement panel asks you to select your geographic region before you can read the Sybase License Agreement. Once you have read the license agreement, accept the terms and conditions to continue.

   Click **Next**.

7. If you selected to install a licensed version of Sybase CEP Serverin Step 5, the SySAM License Entry panel opens.

   Either specify a new license key, use a previously deployed license server, or continue installation without a license key.

   - If you choose to specify a new license key, either enter the key manually or browse for and load the license key. Click **Next** to install a SySAM server.

     Note: If you receive an error saying that the installer detects a SySAM server already running on your machine, click **Previous** to return to the SySAM License Entry panel and choose to use the previous deployed license server.

The following panel displays the install process. Once the server install is complete, configure your SySAM email alert mechanism, and click **Next** to return to your Sybase CEP Server installation.

- If you choose to use a previously deployed license server, enter the **Host Name** and **Port Number**. The SySAM server searches your system for the license server file.
  - If it is available, you are prompted to configure your SySAM email alert mechanism. Click **Next** to return to your Sybase CEP Server installation.
  - If the license server file is not found, the following panel informs you that the license(s) could not be installed. You may run Sybase CEP Server for up to 30 days without a license. If you do not obtain and install a license in 30 days, the software stops working.

    Note: You may navigate back to the SySAM License Entry panel to re-enter any incorrect information.

    When you click **Next** to exit the panel and return to the Sybase CEP Server installation, you are informed again about operating the software without a license, and asked to confirm that you understand this policy.

8. Specify clustering options for Sybase CEP Server. These settings may be changed later, if necessary.

- **Standalone Sybase CEP Server** : This option includes all necessary Sybase CEP Server components, including a Manager and Container inside a single process. Select this option if you plan to run Sybase CEP Server as a stand-alone program on a single computer.
- **Cluster Container**: If you are installing a distributed cluster of Sybase CEP Servers and want to designate the current installation as a Container process within that cluster, select this option and specify the Manager URI for the cluster in the Cluster Manager URI box.

  Container Sybase CEP Serverr processes run queries and C/C++ input and output adapters. Every Sybase CEP Server cluster contains one, or (usually) multiple Containers, which are managed by a single active Manager. For more information about the Manager URI in Sybase CEP Server clusters, see *"High Availability"* on page 83.
- **Cluster Manager**: If you are installing a distributed cluster of Sybase CEP Servers and want to designate the current installation as a Manager within that cluster, select this option. Manager Sybase CEP Server processes monitor Containers in a cluster and manage their workload. Every Sybase CEP Server cluster includes one active or primary Manager and may include one or more backup Managers. For instructions on designating a Manager for a cluster of Containers, see *"High Availability"* on page 83.

Click **Next**.

9. Configure communication parameters for your Sybase CEP Server.

a) Specify the host name (the name of the computer on which Sybase CEP Server runs) and the port number for Sybase CEP Server in the **Hostname** and **Port** boxes. The default host name is the local host where the Sybase CEP Server installation is running. The default port number assigned to the Sybase CEP Server is `6789`.

If you are installing multiple instances of Sybase CEP Server which you plan to run at the same time, make sure to assign a different port number to each instance. If you are not sure what host name and port number to specify, consult your system administrator.

Note: Except in cases where you are installing Sybase CEP Engine as a standalone installation, Sybase strongly recommends that you set the host name to a network-addressable name that can be used by all clients needing to communicate with Sybase CEP Server. Avoid setting the host name to `localhost` as this setting prevents external computers from communicating with Sybase CEP Server on your computer and causes connection problems for out-of-process adapters, resolution problems for stream URIs and other Sybase client difficulties.

b) Specify an adapters base folder for Sybase CEP Server in the **Adapters Base Folder** box. This is the folder used to store various Sybase CEP Server data files. By default, this folder is named `SybaseC8Repository`.

If you are installing a Sybase CEP Server cluster, set all Containers in the cluster to the same adapters base folder. See *"High Availability"* on page 83 for more information.

c) Specify a Server Files Storage Folder for Sybase CEP Server in the **Sybase CEP Server Files** storage box. This is the directory where Sybase CEP Server Sybase CEP Serverstores its state files. If you are installing a Sybase CEP Server cluster, set the Server Files Storage Folder for all the Containers and Managers in the cluster to the same storage folder to avoid data loss. See *"High Availability"* on page 83for more information.

d) If you want to install Sybase CEP Server as a Windows service, select the **Install as Windows Service** check box. When installed as a service, Sybase CEP Server is included in the **Services** utility of the **Administrative Tools** folder in the Microsoft Windows **Control Panel**.

Note: When you install Sybase CEP Server as a Windows service, it runs as the Local System account. If your Sybase CEP Serverrequires additional privileges (for example, to access a shared disk or database), use the Log On tab of the Services utility for Sybase CEP Server to specify a different user account and password. For more information about situations requiring additional privileges, see *http://www.microsoft.com/technet/security/guidance/serversecurity/serviceaccount/default.mspx*.

e) To enable Secure Socket Layer (SSL) support for your Sybase Installation, select the **Enable SSL** check box. For more information about configuring SSL, see *"Configuring SSL Support"* on page 39.

Click **Next** to continue to the Pre-Installation Summary panel.

10. Click **Install** to begin installation of Sybase CEP Server. The next panel displays the installation process.

11. If installation is successful, an install summary panel opens. Click **Next** .
12. On the final panel:
    - Leave the **Run** Sybase CEP Server R4 check box selected if you want to start Sybase CEP Server immediately. Otherwise, clear it.
    - Click **Done** to finish installation of Sybase CEP Server.

Silent Sybase CEP Server Installation on Windows

This section describes the procedure for creating and running a script that installs Sybase CEP Server on Windows platforms without the conventional prompts used by the standard installation.

Sybase recommends this installation method for installing multiple identical or almost identical copies of Sybase CEP Server. Sybase does not recommend silent installation for installations that require extensive customizing. Once you modify the installation script, you may run it as many times as necessary to install multiple copies of Sybase CEP Server.

Creating a Silent Install Script

Sybase CEP provides a default installation script named `installer.properties` located in the same directory as the installer. You can use this file or create your own script to capture your installation preferences. To create a silent install script:

1. Locate the appropriate Sybase CEP Server installation file, either from the Sybase Web site, or from the location where it is saved if the file has been previously downloaded. The installer file is an executable (.exe) file beginning with:

   `c8-server-x.x.x` - operating system`...`

   where `x.x.x` is a three-part version number, separated by dots, which indicates the version of Sybase CEP Engine, and `operating_system` is the name of the Windows operating system on which Sybase CEP Engine is being installed.

   If you are installing the Sybase CEP Server on Microsoft Windows Vista, ensure you are logged in as an administrator.
2. From a command line, issue the following command to launch the installer and create a response file from the options you select during installation:

   ```
   <install_launcher_name> -r <response file> [-i console]
   ```

   Where `<response file>` is the response file that will be generated. The specified file name should be an absolute path.
3. If necessary, modify the response file to meet your requirements. Use this sample response file as a guide:

   ```
   #Validate Response File
   #----------------------
   RUN_SILENT=true
   ```

```
#Choose Install Folder
#---------------------
USER_INSTALL_DIR=C:\\sybase

#Choose Product License Type
#---------------------------
SYBASE_PRODUCT_LICENSE_TYPE=license

#Choose Sybase Software Asset Management License
#-----------------------------------------------
SYSAM_LICENSE_SOURCE=license_file
SYSAM_LICENSE_FILE_PATHNAME=C:\\SYCEP\
\CEP_Unserved_CP_unlimited.lic
SYSAM_EXISTING_LICENSE_SERVER_HOSTNAME=my_license_server
SYSAM_EXISTING_LICENSE_SERVER_PORTNUMBER=

#Software Asset Management Notification Setting
#----------------------------------------------
SYSAM_NOTIFICATION_ENABLE=false
SYSAM_NOTIFICATION_SMTP_HOSTNAME=smtp
SYSAM_NOTIFICATION_SMTP_PORTNUMBER=25
SYSAM_NOTIFICATION_SENDER_EMAIL=name@company.com
SYSAM_NOTIFICATION_RECIPIENT_EMAIL=name@company.com
SYSAM_NOTIFICATION_EMAIL_SEVERITY=NONE

#Sybase CEP Server Standalone/Clustering Mode
#--------------------------------------------
SERVER_MODE_SA=1
SERVER_MODE_CC=0
SERVER_MODE_CM=0

#Sybase CEP Server Configuration
#-------------------------------
LOCAL_HOST=myhost
PORT=6789
ADAPTER_FOLDER=C:\\sybase\\SybaseC8\\server\\SybaseC8Repository\
\4.0
SERVER_FILES=C:\\sybase\\SybaseC8\\server\\storage
INSTALL_AS_WINDOWS=1
ENABLE_SSL=0

#Start Server immediately
#------------------------
START_SERVER=\"\"
START_SERVER_1=
START_SERVER_BOOLEAN_1=0
```

Using a Silent Install Script

The command to run the installer in silent mode is:

<install_launcher_name> –f <response file> [-i silent]

Where `<response file>` is the input response file. The specified file name must be an absolute path.

Please note:

- If “INSTALLER_UI=silent” is specified in <response file>, “-i silent” is not required in the command line.
- If –f is not specified, a response file named `installer.properties` that resides in the same directory as the installer is automatically used.

Installing Sybase CEP Studio on Windows

A description of how to install Sybase CEP Studio on Windows platforms.

1. Get the appropriate Sybase CEP Studio installation file.

   Sybase CEP Studio installation files for Windows platforms look like this: `c8-studio-`*version- operating_system* `_vc8-20100118.1-release.exe` where *version* is a three-part number, separated by dots, which indicates the version of Sybase CEP Engine, and *operating_system* is the name of the Windows platform on which you are installing Sybase CEP Engine.

   Note: If you are installing Sybase CEP Studio on Microsoft Windows Vista, make sure that you are logged in as the administrator.

2. Double-click the installation file to start the installation.

   The first Welcome panel of the Sybase CEP Studio Installeropens. The following steps lead you through the Installer workflow. Options for cancelling install and navigating to previous panels in the workflow are available on every panel before actual installation takes place.

3. Click **Next** to continue the installation process.
4. Select the directory where you want to install Sybase CEP Studio and then click **Next**.

   Note: If you are upgrading or reinstalling Sybase CEP Studio and specify that it should be installed in the same directory as an existing installation, a pop-up message indicates that the Sybase CEP Studio is already installed in the folder.

   Click **Yes** to uninstall the previous installation or **No** to upgrade without uninstalling. If you click **Yes**, the next panel confirms that the previous version of Sybase CEP Studio will be uninstalled if you continue:

   1. Click **Next** to continue.

      The next panels display the uninstall process.
   2. A final install panel opens when uninstall is complete. Click **Next** to continue with the upgrade installation.

5. The End-User License Agreement panel asks you to select your geographic region before you can read the Sybase License Agreement. Once you have read the license agreement, accept the terms and conditions to continue.

   Click **Next** to continue to the Pre-Installation Summary panel.

6. Click **Install** to begin installation of Sybase CEP Studio. The next panel displays the installation process.
7. If installation is successful, an install summary panel opens. Click **Next.**
8. On the final panel:

    - Leave the **Run** Sybase CEP Studio R4 check box selected if you want to start Sybase CEP Studio immediately. Otherwise, clear it.
    - Click **Done** to finish installation of Sybase CEP Studio.

Final Steps After Upgrading or Reinstalling on Windows

A description of how to complete your Sybase CEP Server or Sybase CEP Studio upgrades and reinstalls.

1. Copy any desired custom adapter `.adl` files, user-defined function `.udf` files, custom library `.so` or `.dll` files, and CCL projects and supporting files from the old installation, or from your backed up copies, into the equivalent directory in the new installation.
2. Recompile any custom adapters, or C/C++ user-defined function files with the new version of Sybase CEP Engine.

Note that you may need to reconfigure some features of your Sybase CEP application after upgrading or reinstalling Sybase CEP Engine. For example, you need to re-create any workspaces other than the Default workspace, and probably need to reset Persistence features in the new installation.

Starting Sybase CEP Server on Windows

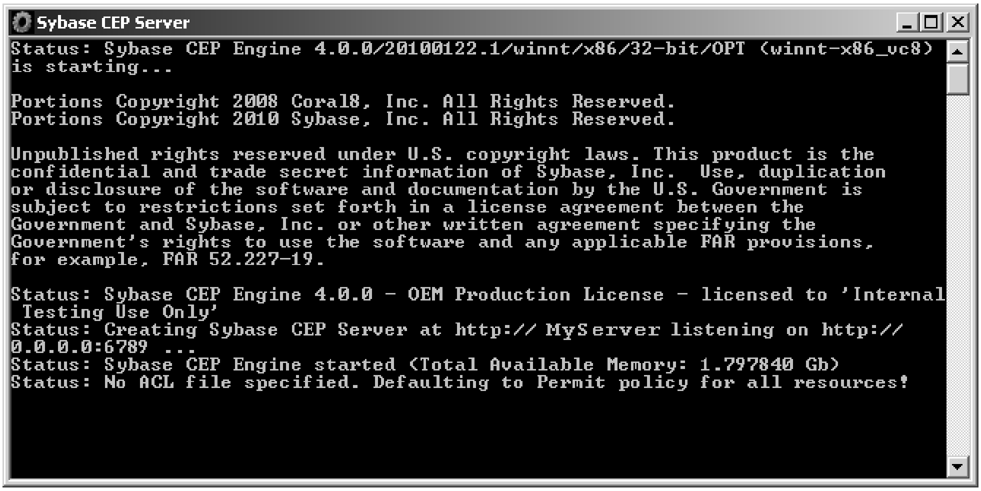
A description of how to start your Sybase CEP components on Windows platforms.

Sybase CEP Server can be started in either of the following ways:

1. Click the **Start button** in the Microsoft Windows taskbar to open the Start menu, and select the Sybase CEP option from the appropriate submenu of the Programs menu.

   By default, Sybase CEP Server appears in the Sybase CEP ServerServer submenu of the Sybase CEP submenu under Programs.
2. Double-click the **Sybase CEP Server** icon on your Desktop.

If Sybase CEP Server starts successfully, a command window appears and displays messages similar to the following (actual messages in this window may be different from the ones displayed here):



```
Status: Sybase CEP Engine 4.0.0/20100122.1/winnt/x86/32-bit/OPT (winnt-x86_vc8)
is starting...

Portions Copyright 2008 Coral8, Inc. All Rights Reserved.
Portions Copyright 2010 Sybase, Inc. All Rights Reserved.

Unpublished rights reserved under U.S. copyright laws. This product is the
confidential and trade secret information of Sybase, Inc.  Use, duplication
or disclosure of the software and documentation by the U.S. Government is
subject to restrictions set forth in a license agreement between the
Government and Sybase, Inc. or other written agreement specifying the
Government's rights to use the software and any applicable FAR provisions,
for example, FAR 52.227-19.

Status: Sybase CEP Engine 4.0.0 - OEM Production License - licensed to 'Internal
 Testing Use Only'
Status: Creating Sybase CEP Server at http:// MyS erver listening on http://
0.0.0.0:6789 ...
Status: Sybase CEP Engine started (Total Available Memory: 1.797840 Gb)
Status: No ACL file specified. Defaulting to Permit policy for all resources!
```

Starting Sybase CEP Studio on Windows

A description of the different methods you can use to start Sybase CEP Studio.

Sybase CEP Studio can be started in any of the following ways:

1. Click the **Start button** in the Microsoft Windows taskbar to open the Start menu, and then select the Sybase CEP Studio option from the appropriate submenu of the Programs menu.

   By default, Sybase CEP Studio appears in the Studio submenu of the Sybase submenu under Programs.
2. Double-click theSybase CEP Studio icon on your Desktop.
3. Enter the following command in a Command Prompt window, where `InstallDir` is where you installed Sybase CEP Studio:

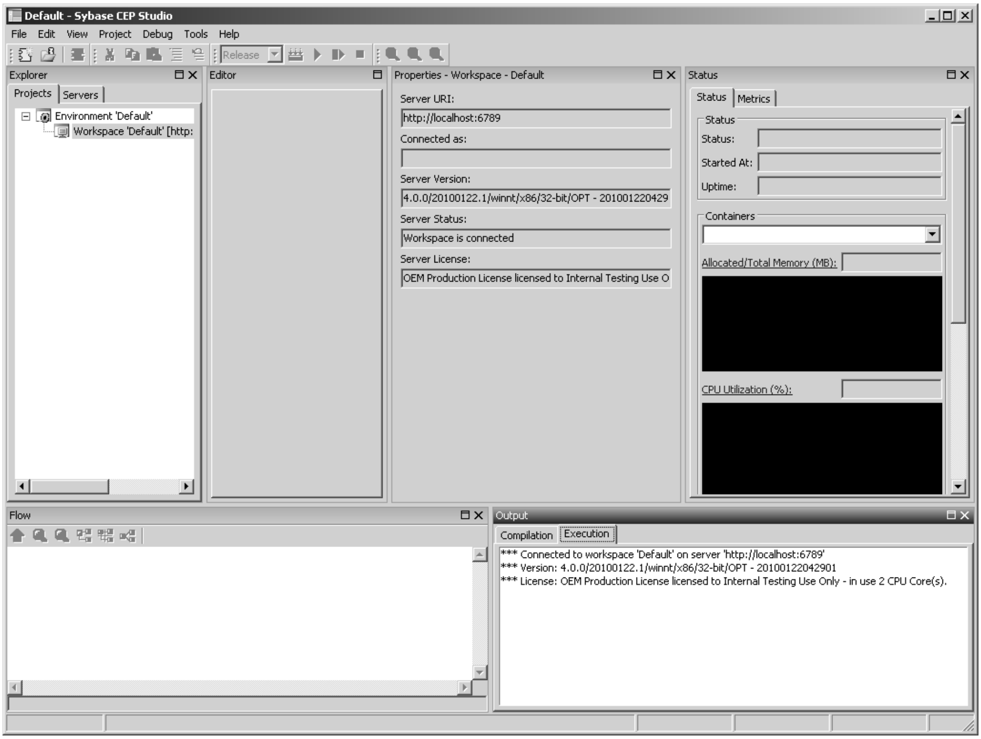
   ```
   InstallDir\Studio\bin\c8_studio.exe
   ```

   If you start Sybase CEP Studio from a Command Prompt window, you can use either or both of the following parameters to modify its behavior:

-r=<folder_path> or --repository=<folder_path>	Use the folder specified with folder_path as the SybaseC8Repository. The SybaseC8Repository is where Sybase CEP Studio stores files containing preference and configuration information, a temporary directory, a directory containing SSL configuration information, and the examples directory. By default, the Sybase Repository is Documents and Settings\user\My Documents\SybaseC8Repository\vers, where user is the name of the user who installed Sybase CEP Studio, and vers is the version number of Sybase CEP Engine.

--dont-copy-examples	Start Sybase CEP Studio without copying the examples directory from the installation directory to the SybaseC8Repository. Without this parameter, Sybase CEP Studio searches for the examples directory in the SybaseC8Repository and copies it from the installation directory if it doesn't already exist there. Use this parameter to save space when you don't need access to the examples.

If Sybase CEP Studio starts successfully, it displays a window similar to the following:



Server URI: http://localhost:6789

Connected as:

Server Version: 4.0.0/20100122.1/winnt/x86/32-bit/OPT - 201001220429

Server Status: Workspace is connected

Server License: OEM Production License licensed to Internal Testing Use O

```
*** Connected to workspace 'Default' on server 'http://localhost:6789'
*** Version: 4.0.0/20100122.1/winnt/x86/32-bit/OPT - 20100122042901
*** License: OEM Production License licensed to Internal Testing Use Only - in use 2 CPU Core(s).
```

Shutting Down Sybase CEP Engine on Windows

A description of how to shut down your Sybase CEP components on Windows platforms.

To shut down Sybase CEP Server Sybase CEP Studio in Microsoft Windows, click **Exit** from the **File** menu.

To shut down Sybase CEP Server, click inside the command window in which Sybase CEP Server is running, and press **Ctrl+C** followed by the **Enter** key.

Note: If, instead of using **Ctrl+C**, you shut down Sybase CEP Server by clicking the ☒ icon in the upper right corner of the Sybase CEP Server window, a message stating that Windows cannot end the program and asking you if you want to force the program to end may appear.

This message warns that all unsaved data will be lost. However, it is safe to shut down Sybase CEP Server in this way, and the data is not lost as a result.

Uninstalling Sybase CEP Enginefrom Microsoft Windows (Non-Vista)

A description of how to uninstall Sybase CEP Engineon Non-Vista Windows Platforms.

To uninstall any component of Sybase CEP Engine on any Microsoft Windows platform except Microsoft Windows Vista, navigate to the submenu of the component you want to uninstall in the Microsoft Windows Start menu and select the Uninstall command for the component you want to remove. For example, to uninstall Sybase CEP Server if it is located in its default location in the Start menu follow these steps:

1. Click the **Start button** in the Microsoft Windows taskbar.
2. Open the Programs submenu of the Start menu.
3. Open the Sybase CEP submenu of the Programs submenu.
4. Open the Server submenu of the Sybase CEP submenu.
5. Select the **Uninstall** command.

   Note: While uninstalling Sybase CEP Server, it may start temporarily and then shut down.

   Note: Uninstalling Sybase CEP Server and Sybase CEP Studio does not automatically remove the adapters base folder or the SybaseC8Repository, which contains the Sybase CEP Engineexample projects. To uninstall the examples,remove the SybaseC8Repository manually. On Microsoft Windows, the default location of the SybaseC8Repository is:

   ```
   C:\Documents and Settings\username\My Documents
   \SybaseC8Repository
   ```

   If Sybase CEP Server is installed as a Windows service, the Sybase CEP Studio SybaseC8Repository is

   ```
   C:\Documents and Settings\username\My Documents
   \SybaseC8Repository
   ```

   while the Sybase CEP Server adapters base folder is:

   ```
   C:\sybase\SybaseC8\Server\SybaseC8Repository
   ```

   For instructions about uninstalling Sybase CEP Engineon Microsoft Windows Server, see *"Uninstalling Sybase CEP Engineon Microsoft Windows Vista* on page 17.

Uninstalling Sybase CEP Engine from Microsoft Windows Vista

A description of how to uninstall Sybase CEP Server and Sybase CEP Studio on Microsoft Windows Vista.

1. Locate the uninstall files in the Server and Studio program folders, which are, typically, `C:\sybase\SybaseC8\uninstall\CEP\server\uninstall.exe` and

```
C:\sybase\SybaseC8\uninstall\CEP\studiouninstall
\uninstall.exe
```

2. Right-click each of these files and select **Run as Administrator**.
3. Manually remove the Sybase CEP Server and Sybase CEP Studio icons from the Desktop.
4. Manually remove the `C:\ProgramData\SybaseC8` folder.

Note: Do not use the unistall procedures listed in "Uninstalling Sybase CEP Engine on Non-Vista Platforms".

Installing Sybase CEP Engine on UNIX-Like Operating Systems

Learn how to install, reinstall, upgrade, or uninstall Sybase CEP Engine on UNIX-Like Operating Systems.

The commands shown in this chapter are presented in Bash shell syntax. Modify the commands as appropriate for other shells.

Backing up Important Information on UNIX: Upgrades and Reinstalls Only

The previous installation of Sybase CEP Engine is automatically removed if you are upgrading to a new version.

Sybase particularly recommends backing up the following information before proceeding with the upgrade or reinstall:

- Any customized files in your `Server\conf` folder.
- Any information in your adapters base folder (by default the adapters base folder is named `SybaseCEPRepository`).
- Any custom adapter `.adl` files.
- Any user-defined function `.udf` files.
- Any custom library `.so` or `.dll` files.

Even if you are upgrading or reinstalling to a different directory, Sybase recommends that you back up your adapter base folder as a precautionary measure before proceeding.

Installing Sybase CEP Server on UNIX

There are two methods for installing Sybase CEP Server on UNIX-like operating systems.

The standard installation procedure, described in *"Standard Sybase CEP Server Installation"* on page 19, includes a panel that prompts you to specify a number of Sybase CEP Server configuration settings. The silent Sybase CEP Server installation method, described in *"Silent Sybase CEP Server Installation"* on page 22, allows you to specify these settings in advance and run the script "silently", that is, without prompts. Sybase recommends this method for installing multiple Sybase CEP Server instances, if they do not require highly customized settings.

If you are upgrading or reinstalling Sybase CEP Engine, using the standard installation method, the default installation settings reflect your previous selections. Otherwise, the upgrade process behaves similarly to the initial install process.

Standard Sybase CEP ServerInstallation on Unix

A description of how to install the Standard Sybase CEP Server on UNIX.

1. Get the appropriate Sybase CEP Server installation file.

   Sybase CEP Server installation files for UNIX-like operating systems look like this:`c8-server-` *version- operating_system* `-20100109.1-release.bin` where *version* is a three-part number, separated by dots, which indicates the version of Sybase CEP Engine, and *operating_system* is the name of the UNIX-like operating system on which you are installing Sybase CEP Engine.
2. Double-click the installation file to start installation.

   The Welcome panel of the Sybase CEP Server Installer opens. The following steps lead you through the Installer workflow. Options for cancelling install and navigating to previous panels in the workflow are available on every panel before actual installation takes place.
3. Click **Next** to continue the installation process.
4. Select the directory where you want to install Sybase CEP Server and click **Next**.

   Note: If you are upgrading or reinstalling Sybase CEP Server and specify the installation location to be the same directory as an existing installation, a pop-up message indicates that the Sybase CEP Server is already installed in the folder.

   Click **Yes** to uninstall the previous installation or **No** to upgrade without uninstalling. If you click **Yes**, the next panel confirms that the previous version of Sybase CEP Server will be uninstalled if you continue:

   1. Click **Next** to continue.
      The next panels display the uninstall process.
   2. A final install panel opens when uninstall is complete. Click **Next** to continue with the upgrade installation.
5. Select either to install a licensed copy of Sybase CEP Server or to evaluate a trial version of Sybase CEP Server.
6. The End-User License Agreement panel asks you to select your geographic region before you can read the Sybase License Agreement. Once you have read the license agreement, accept the terms and conditions to continue.

   Click **Next**.
7. If you selected to install a licensed version of Sybase CEP Server in Step 5, the SySAM License Entry panel opens.

Either specify a new license key, use a previously deployed license server, or continue installation without a license key.

- If you choose to specify a new license key, either enter the key manually or browse for and load the license key. Click **Next** to Install a SySAM server.

  Note: If you receive an error saying that the installer detects a SySAM server already running on your machine, click **Previous** to return to the SySAM License Entry panel and choose to use the previous deployed license server.

  The following panel displays the install process. Once the server install is complete, configure your SySAM email alert mechanism, and click **Next** to return to your Sybase CEP Server installation.
- If you choose to use a previously deployed license server, enter the Host Name and Port Number. The SySAM server searches your system for the license server file.
  - If it is available, you are prompted to configure your SySAM email alert mechanism. Click Next to return to your Sybase CEP Server installation.
  - If the license server file is not found, the following panel informs you that the license(s) could not be installed. You may run Sybase CEP Server for up to 30 days without a license. If you do not obtain and install a license in 30 days, the software stops working.

    Note: You may navigate back to the SySAM License Entry panel to re-enter any incorrect information.

    When you click **Next** to exit the panel and return to the Sybase CEP Server installation, you are informed again about operating the software without a license, and asked to confirm that you understand this policy.

8. Specify clustering options for Sybase CEP Server. These settings may be changed later, if necessary.

- **Standalone Sybase CEP Server** : This option includes all necessary Sybase CEP Server components, including a Manager and Container inside a single process. Select this option if you plan to run Sybase CEP Server as a stand-alone program on a single computer.
- **Cluster Container**: If you are installing a distributed cluster of Sybase CEP Servers and want to designate the current installation as a Container process within that cluster, select this option and specify the Manager URI for the cluster in the Cluster Manager URI box.

  Container Sybase CEP Server processes run queries and C/C++ input and output adapters. Every Sybase CEP Server cluster contains one, or (usually) multiple Containers, which are managed by a single active Manager. For more information about the Manager URI in Sybase CEP Server clusters, see *"High Availability"* on page 83.
- **Cluster Manager**: If you are installing a distributed cluster of Sybase CEP Servers and want to designate the current installation as a Manager within that cluster, select this option. Manager Sybase CEP Server processes monitor Containers in a cluster and

manage their workload. Every Sybase CEP Server cluster includes one active or primary Manager and may include one or more backup Managers. For instructions on designating a Manager for a cluster of Containers, see *"High Availability"* on page 83.

Click **Next**.

9. Configure communication parameters for your Sybase CEP Server.

a) Specify the host name (the name of the computer on which Sybase CEP Server runs) and the port number for Sybase CEP Server in the **Hostname** and **Port** boxes. The default host name is the local host where the Sybase CEP Server installation is running. The default port number assigned to the Sybase CEP Serverr is `6789`.

If you are installing multiple instances of Sybase CEP Serverwhich you plan to run at the same time, make sure to assign a different port number to each instance. If you are not sure what host name and port number to specify, consult your system administrator.

Note: Except in cases where you are installing Sybase CEP Engine as a standalone installation, Sybase strongly recommends that you set the host name to a network-addressable name that can be used by all clients needing to communicate with Sybase CEP Server. Avoid setting the host name to `localhost` as this setting prevents external computers from communicating with Sybase CEP Server on your computer and causes connection problems for out-of-process adapters, resolution problems for stream URIs and other Sybase client difficulties.

b) Specify an adapters base folder for Sybase CEP Server in the **Adapters Base Folder** storage box. This is the folder used to store various Sybase CEP Server data files. By default, this folder is named `SybaseC8Repository` found in your My Documents folder under the appropriate Sybase CEP Engineversion number.

If you are installing a Sybase CEP Servercluster, set all Containers in the cluster to the same adapters base folder. See *"High Availability"* on page 83 for more information.

c) Specify a Server Files Storage Folder for Sybase CEP Server in the Sybase CEP Server Files Storage box. This is the directory where Sybase CEP Sybase CEP Server stores its state files. If you are installing a Sybase CEP Server cluster, set the Server Files Storage Folder for all the Containers and Managers in the cluster to the same storage folder to avoid data loss. See *"High Availability"* on page 83for more information.

d) To enable Secure Socket Layer (SSL) support for your Sybase CEP Server Installation, select the **Enable SSL** check box. For more information about configuring SSL, see *"Configuring SSL Support"* on page 39.

Click **Next** to continue to the Pre-Installation Summary panel.

10. Click **Install** to begin installation of Sybase CEP Server. The next panel displays the installation process.

11. If installation is successful, an install summary panel opens. Click **Done**.

12. On the final panel:

- Leave the **Run** Sybase CEP Server R4 check box selected if you want to start Sybase CEP Server immediately. Otherwise, clear it.
- Click **Done** to finish installation of Sybase CEP Server.

Silent Sybase CEP Server Installation on Unix

This section describes the procedure for creating and running a script that installs Sybase CEP Server on UNIX-like operating systems without the conventional prompts used by the standard installation.

Sybase recommends this installation method for installing multiple identical or almost identical copies of Sybase CEP Server. It does not recommend silent installation for installations that require extensive customizing. Once you modify the installation script, you may run it as many times as necessary to install multiple copies of Sybase CEP Server.

For more installation information, see *"Standard Sybase CEP Server Installation"* on page 19.

The Sybase CEP Server installer on UNIX-like operating systems reads and updates a file called `config.vars`, located in the `SybaseC8/server/conf` subdirectory of the parent directory under which Sybase CEP Server is installed. This file consists of a series of Bash shell settings that describe the configurable Sybase CEP Server preferences. The settings contained in the `config.vars` file always reflect the settings of the most recent previous install (if the Sybase CEP Server has already been installed using the same installation files on the same computer) or the default installer settings.

When performing a standard installation, the installer displays the current settings and prompts you to approve or change them. The installation process can be set to execute silently, however, by adding a `SILENT=yes` variable to the beginning of the `config.vars` file.

Sybase provides a sample silent configuration file called `config.vars.silent` with your Sybase CEP Server installation script. This file can be modified, renamed, and repackaged in the installation `.tar.gz` file to make the install process silent.

Creating a Silent Install Script

To create a silent install script:

1. Get the appropriate Sybase CEP Server installation file, either from the Sybase Web site, or from the location where it is saved if the file has been previously downloaded. The installer file is an executable (.exe) file beginning with:

   `c8-studio-` *x.x.x* - operating system`...`

   where *x.x.x* is a three-part version number, separated by dots, which indicates the version of Sybase CEP Engine, and *operating_system* is the name of the Windows operating system on which Sybase CEP Engine is being installed.

2. Obtain a copy of the `config.vars.silent` file. To do this, either copy the file from the `SybaseC8/server/conf` directory of a previously installed Sybase CEP Server copy, or use these steps to extract the file from the Sybase CEP Server installation file:

   a) Type this command to expand the `.tar.gz` file:

   ```
   tar zxf
   server-installation-fil
   e.tar.gz
   ```

   b) Copy the `config.vars.silent` file to the desired location:

   ```
   cp SybaseC8/server/conf/config.vars.silent
   new-file-location-and-name
   ```

3. Modify the `config.vars.silent` file to your requirements. The example `config.vars.silent` file looks like this:

```
# Sybase CEP Server Configuration
    SILENT=yes
    CONFIG_HOST=`hostname`
    CONFIG_PORT=6789                    #CONFIG_SERVER_CONFIG=c8-
server.rc     #CONFIG_SERVER_SCRIPT=c8-server.rc
#CONFIG_SERVER_LOG=c8-server.log
    #CONFIG_SERVER_PID=c8-server.pid
    #CONFIG_ENABLE_MANAGER=true
    CONFIG_ENABLE_CONTAINER=true
    CONFIG_MANAGER_URI=http://$CONFIG_HOST:$CONFIG_PORT
    CONFIG_ENABLE_SSL=false
    CONFIG_DESKTOP_SHORTCUT=no
    CONFIG_COPY_EXAMPLES=yes
    CONFIG_START_SERVER=yes
    CONFIG_ADAPTERS_BASE=$CONFIG_INSTALL_FOLDER/
SybaseC8Repository
    CONFIG_STORAGE_BASE=$CONFIG_INSTALL_FOLDER/storage
```

Here is an explanation of these settings:

- SILENT=yes: Is a required line at the beginning of the file. It specifies that the installation should be performed silently.
- CONFIG_HOST='hostname': Specifies the host name. Replace hostname either with the local host name or the fully qualified host name of the machine where Sybase CEP Server is installed. Make sure to use the full host name if Sybase CEP Server needs to accept connections from other machines.

  You can use the back quote characters (`) to execute the command with a subshell or use this alternative setting for the host: `CONFIG_HOST=$(` hostname. See "Server Identification Warning" for more information about properly setting the host name.
- CONFIG_PORT=6789: Shows the number of the port used by Sybase CEP Server. The default port is 6789. Change this number, as needed. If you are installing multiple copies of Sybase CEP Server on a single computer, indicate a unique port number for each copy.

- CONFIG_SERVER_SCRIPT=c8-server.rc: Specifies the configuration file for Sybase CEP Server. Do not change this file name.
- CONFIG_SERVER_LOG=c8-server.log: Specifies the log file for Sybase CEP Server. Do not change this file name.
- CONFIG_SERVER_PID=c8-server.pid: Specifies the file that contains the Server Process Identifier. This file only exists when Sybase CEP Server is running. Do not change this file name.
- CONFIG_ENABLE_MANAGER=true and CONFIG_ENABLE_CONTAINER=true: Enable or disable Manager and/or Container processes for Sybase CEP Server. To enable the Manager and/or Container, leave the appropriate settings at true. To disable the Manager or Container, change the appropriate setting to false.

  Containers processes run queries and C/C++ input and output adapters. Managers monitor Container processes and manage their workload. On stand-alone Sybase CEP Server installations, enable both the Manager and the Container process. If the installation is part of a Sybase CEP Servercluster, you may enable it as a Container only, Manager only, or both. Every Sybase CEP Server cluster should include one or more Manager process (an active, or primary manager, and, optionally, one or more backup Managers) and one or more Container processes. For instructions on how to create a Sybase CEP Server cluster, see *High Availability* on page 83.
- CONFIG_MANAGER_URI=http://$CONFIG_HOST:$CONFIG_PORT: Specifies the URI for the primary Manager in a Sybase CEP Server cluster. If you are installing a stand-alone installation of Sybase CEP Server, or are installing this copy of Sybase CEP Server as a Manager, the default setting for the Manager URI does not need to be changed. If you are installing the current Sybase CEP Server copy as a Container that is part of a Sybase CEP Server cluster, however, set the Manager URI to the full host name. For more information about the Manager URI in Sybase CEP Server clusters see *High Availability* on page 83 .
- CONFIG_ENABLE_SSL=false: Enables or disables the Secure Socket Layer (SSL) features. To leave SSL disabled, leave this setting at false. To enable SSL, change this setting to true. For more information about configuring SSL, see *Configuring SSL Support* on page 39.
- CONFIG_DESKTOP_SHORTCUT=no: Indicates whether or not to create a shortcut for Sybase CEP Server on your desktop during installation. Leave this setting at no, if you don't want to create a shortcut. Otherwise change it to yes.
- CONFIG_COPY_EXAMPLES=yes: Indicates whether or not to install a fresh copy of Sybase CEP application examples during installation. To install examples, leave this setting at yes. Otherwise, change the setting to no. Examples are valuable when learning to use Sybase CEP Engine, but are not useful on production installations.
- CONFIG_START_SERVER=yes: Indicates whether or not to start Sybase CEP Server immediately after installation. To start Sybase CEP Server as soon as it is installed, leave this setting at yes. Otherwise, change the setting to no.
- CONFIG_ADAPTERS_BASE=$CONFIG_INSTALL_FOLDER/ SybaseC8Repository: Specifies the Sybase CEP Server directory used to store various

Sybase CEP Serverdata files. By default, this is called `SybaseC8Repository`. If you want to change the path and/or name of this directory, change this setting as needed.

If you are installing a Sybase CEP Server cluster, set all Containers in the cluster to the same adapters base folder. See *High Availability* on page 83 for more information.

- CONFIG_STORAGE_BASE=$CONFIG_INSTALL_FOLDER/storage: Specifies the storage directory for Sybase CEP Server state files. By default, this is called storage. If you want to change the path and/or name of this directory, change this setting as needed.

  If you are installing a Sybase CEP Server cluster, set the storage folder for all the Containers and Managers in the cluster to the same directory to avoid data loss. See *High Availability* on page 83 for more information.

Note: Since the entries in this file are executed as Bash script, any valid Bash script entry may be used in specifying the settings. However, be sure to check the settings of any environment variables you use in the script on any machines where you plan to install Sybase CEP Server to avoid unexpected results during installation. This is particularly true for production installations.

4. Save the edited file under the name `config.vars` (omitting the final `.silent` ending of the original name).

5. Make the new silent install `config.vars` file a part of the installation by following these steps:

a) If you have not already expanded the `.tar.gz` installation file (or have repacked it since expanding) issue this command:

```
tar zxf
server-installation-file
.tar.gz
```

b) Include the new `config.vars` file in the distribution:

```
cp
full-file-path
/config.vars SybaseC8/server/conf
```

c) Repackage the distribution. To avoid confusing the standard installation file with the installation that performs a silent install, repackage the file under a different name:

```
tar zcf
new-server-installation-fil
e.tar.gz
```

Using a Silent Install Script

Once your silent install file is created, you can run it as many times as necessary on one or more computers to install the desired number of Sybase CEP Server copies. For each installation, perform the following steps:

1. Change directories to the parent directory under which you want to install Sybase CEP:

   ```
   cd
   parent-directory
   ```

   where `parent-directory` is the full path of an existing directory under which you want to install Sybase CEP.
2. If you have not already expanded the `.tar.gz` installation file (or have repacked it since expanding), expand the file by using this command:

   ```
   tar xzvf
   new-server-installation-file
   .tar.gz
   ```

   where `new-server-installation-file` is the name of the modified Sybase CEP Server installation file.
3. Run the script `install-server.sh` to install the Sybase CEP Server:

   ```
   ./SybaseC8/install-server.sh
   ```
4. Wait for the installation to complete.

Installing Sybase CEP Studio on UNIX

A description of how to install Sybase CEP Studio on UNIX-like systems.

Prerequisites

If installing Sybase CEP Studio on Solaris, run fc-cache to make sure the font caches are up to date:

```
# svcadm restart svc:/application/font/fc-cache:default
```

fc-cache runs automatically from the fc-cache main page at system boot to ensure font caches are up to date through the fc-cache service.

Task

1. Get the appropriate Sybase CEP Studio installation file.

   Sybase CEP Studio installation files for UNIX-like operating systems look like this: `SybaseC8-studio-` *version-operating_system* `-20100109.1-release.bin` where *version* is a three-part number, separated by dots, and representing the version of Sybase CEP Engine, and *operating_system* is the name of the UNIX-like operating system on which you are installing Sybase CEP Engine.
2. Double-click the installation file to start the installation.

The first panel of the Sybase CEP Studio Installer opens. The following steps lead you through the Installer workflow. Options for cancelling install and navigating to previous panels in the task flow are available on every panel before actual installation takes place.

3. Click **Next** to continue the installation process.
4. Select the directory where you want to install Sybase CEP Studio and then click **Next**.

   If you are upgrading or reinstalling Sybase CEP Studio and specify that it should be installed in the same directory as an existing installation, a pop-up message indicates that the Sybase CEP Studio is already installed in the folder.

   Click **Yes** to uninstall the previous installation or **No** to upgrade without uninstalling. If you click **Yes**, the next panel confirms that the previous version of Sybase CEP Studio will be uninstalled if you continue:

   a) Click **Next** to continue.

      The next panels display the uninstall process.

   b) A final install panel opens when uninstall is complete. Click **Next** to continue with the upgrade installation.
5. The End-User License Agreement panel asks you to select your geographic region before you can read the Sybase License Agreement. Once you have read the license agreement, accept the terms and conditions to continue.

   Click **Next** to continue to the Pre-Installation Summary panel.
6. Click **Install** to begin installation of Sybase CEP Studio. The next panel displays the installation process.
7. If installation is successful, an install summary panel opens. Click **Next**.
8. On the final panel:

   - Leave the **Run** Sybase CEP Studio R4 check box selected if you want to start Sybase CEP Studio immediately. Otherwise, clear it.
   - Click **Done** to finish installation of Sybase CEP Studio.

Final Steps After Upgrading or Reinstalling on UNIX

A description of how to complete your Sybase CEP Server or Sybase CEP Studio upgrades and reinstalls.

If you upgraded or reinstalled Sybase CEP Server or Sybase CEP Studio, perform the following final steps:

1. Compare the `c8-server.conf` and `c8-services.xml` files from the previous installation with the versions of these files in the new installation.

   Copy any desired settings from the old file into the new files.

   If you have installed Sybase CEP Server over an existing installation, use the backed-up copies of these files to compare with the newly-installed files. Do not replace the actual configuration files with your backed-up copies, as Sybase CEP Engine configuration files may change from one release to another. If you have not customized the `c8-services.xml` file or the `c8-server.conf` file in any way you may skip this step.

2. Copy any desired custom adapter `.adl` files, user-defined function `.udf` files, custom library `.so` or `.dll` files, and CCL projects and supporting files from the old installation, or from your backed up copies, into the equivalent directory in the new installation.
3. Recompile any custom adapters, or C/C++ user-defined function files with the new version of Sybase CEP Engine.

Note that you may need to reconfigure some features of your Sybase CEP application after upgrading or reinstalling Sybase CEP Engine. For example, you need to recreate any workspaces other than the Default workspace, and probably need to reset Persistence features in the new installation.

Starting Sybase CEP Server and Sybase CEP Studio on UNIX

A description of how to start Sybase CEP components in different modes.

Starting Sybase CEP Serveron UNIX

A description of how to start Sybase CEP Server on UNIX-like operating systems.

To start Sybase CEP Server on UNIX-like operating systems in background mode, use the `c8-server.rc` script:

```
parent-directory/SybaseC
8/server/c8-server.rc start
```

where *parent-directory* is the parent directory under which Sybase CEP Server was installed.

To start Sybase CEP Server in console mode, double-click on the Sybase CEP Server desktop icon:

When Sybase CEP Server runs in console mode, Sybase CEP Server status and messages appear in a dedicated window.

Starting Sybase CEP Studio

A description of how to start Sybase CEP Studio on UNIX-like operating systems.

Issue the following command:

```
parent-directory
/SybaseC8/studio/c8-studio.sh
```

where *parent-directory* is the parent directory under which Sybase CEP Studio was installed.

Alternately, Sybase CEP Studio can be started by double-clicking the Sybase CEP Studio desktop icon:

If you start Sybase CEP Studio from the command line, you can use either or both of the following parameters to modify its behavior:

-r=<folder_path> or --repository=<folder_path>	Use the directory specified with folder_path as the SybaseC8Repository rather than the default. The SybaseC8Repository is where Sybase CEP Studio stores files containing preference and configuration information, a temporary directory, a directory containing SSL configuration information, and the examples directory. By default, the SybaseC8Repository is /home/user/SybaseC8Repository/vers, where user is the name of the user who installed Sybase CEP Studio, and vers is the version number of Sybase CEP Engine.
--dont-copy-examples	Start Sybase CEP Studio without copying the examples directory from the installation directory to the SybaseC8Repository. Without this parameter, Sybase CEP Studio searches for the examples directory in the SybaseC8Repository and copies it from the installation directory if it doesn't already exist there. Use this parameter to save space when you don't need access to the examples.

Automatically Starting Sybase CEP Server After Rebooting

By default, Sybase CEP Server does not automatically start or restart when a UNIX-like operating system is rebooted, but it can be configured to restart automatically.

On RedHat Enterprise Linux, Fedora, and other RedHat-based distributions of Linux, use the following commands to enable automatic restart. Note that you require root privileges to make the following changes:

```
ln -s
parent-directory
/SybaseC8/server/c8-server.rc \
/etc/init.d/c8-server.rc
/sbin/chkconfig --add c8-server
```

where *parent-directory* is the parent directory under which you installed Sybase CEP Engine.

In addition to restarting Sybase CEP Engine after the system is rebooted, the `init.d` script reboots Sybase CEP Server in the event of a Sybase CEP Server crash.

For more details about restarting, and for the most current information for you operating system, see your operating system documentation.

Shutting Down Sybase CEP Engine on UNIX

A description of how to shut down Sybase CEP Engine on UNIX-like Operating systems. Includes steps for shutting down Sybase CEP Studio, and the Sybase CEP Server in both background and console modes.

To shut down Sybase CEP Studio on UNIX-like operating systems, click **Exit** from the **File** menu.

To shut down Sybase CEP Server:

- If Sybase CEP Server is running in background mode, use the c8-server.rc script to shut down Sybase CEP Server:

  ```
  parent-directory/SybaseC8/server/c8-server.rcstop
  ```

  where *parent-directory* is the parent directory under which you installed Sybase CEP Engine. Make sure that the *same c8-server.rc* script is used to start and stop the Sybase CEP Server process. Using different scripts for starting, stopping, and/or restarting causes the restart process to fail.
- If Sybase CEP Server is running in console mode, click inside the command window in which Sybase CEP Server is running, and press the **CTRL+C** keys or click the ☒ icon in the upper right corner of the window.

Uninstalling Sybase CEP Engine

A description of how to uninstall the Sybase CEP Engine component.

1. Locate the uninstall files in the Server and Studio program folders, which are, typically, `/opt/SybaseC8/uninstall/CEP/serveruninstall/uninstall` and `/opt/SybaseC8/uninstall/CEP/studiouninstall/uninstall`.
2. Run the uninstallation files to remove both Sybase CEP Server and Sybase CEP Studio.

Licensing Sybase CEP

Learn about the Sybase Software Asset Management (SySAM) license, SySAM license models, grace periods, core restrictions and affinities, and SySAM product requirements.

SySAM performs license administration and asset management tasks for Sybase products. A valid SySAM license is required to run on all Sybase CEP Engine supported environments.

Obtaining a Valid SySAM License

Use Sybase Product Download Center (SPDC), an online Web portal, to download and license your software.

When you purchase a Sybase product, you receive an e-mail or web key that tells you how to log in to SPDC.

The e-mail or web key you receive also provides specific product and licensing information. Use this information when you generate your licenses.

Note: The SySAM Quick Start Guide explains how to generate license files. Once generated, place license files (or license server reference license files) in the <CEP installation directory>/server/SYSAM-2_0/licenses folder.

SySAM License Models

Learn about the SySAM license models that Sybase CEP supports.

Sybase CEP supports both served and unserved SySAM license models:

- Served license model: All license files are deployed on a license server. You can obtain the license from the server.
- Unserved license model: The license file is placed on the same machine with the product.

The served license model is the recommended deployment model.

SySAM Grace Periods

Learn about the grace period for obtaining a valid Sybase production or trial licence.

There is a 30 day grace period following installation to obtain a valid production or trial license from Sybase. After the 30 day interval has expired, the product does not operate. To obtain trial licenses, please contact your Sybase sales representative.

Sybase also includes a runtime grace period of 30 days, which is initiated when a previously-acquired license becomes inaccessible for any reason, including if the SySAM license server goes down or you experience network issues which prevent you from connecting to the SySAM license server. All licensed features continue to run during the runtime grace period.

Core Restrictions and Affinities

Sybase CEP supports features that allow administrators to restrict the execution of the Sybase CEP Engine to a specific number of cores and to establish affinities to specific cores.

Note: These features are independent of SySAM licensing requirements.

SySAM Product Requirements

Learn about SySAM product requirements for different platforms and operating systems.

Table 1. SySAM product requirements

Platform	Supported OS	SySAM distribution
Linux-64 (AMD/Intel)	Red Hat 5.0 (AMD) Red Hat 5.0 (Intel) SUSE 10 (AMD)	Sybase-sysam-1.8.2-rhel4-x86_64.tar.gz
Sun-64 (SPARC)	Solaris 10	Sybase-sysam-1.8.2-sunos-sparc.tar.gz
Sun-64 (AMD)	Solaris 10	Sybase-sysam-1.8.2-sunos-x86_64.tar.gz
winx64	Windows Server 2003 x64 SP2	Sybase-sysam-1.8.2- winnt-x86_64_vc8.tar.gz
Windows (32) (Dev only)	Windows (32)	Sybase-sysam-1.8.2- winnt-x86_vc8.tar.gz

Reading the SySAM Documentation

Review the SySAM documentation for specific information about SySAM licensing, license models, and procedures.

- Sybase Software Asset Management 2 Users Guide introduces asset management concepts and provides instructions for establishing and administering SySAM 2 licenses.

- SySAM 2 Quick Start Guide details how to get the SySAM-enabled Sybase product up and running quickly.
- FLEXnet Licensing End User Guide explains FLEXnet Licensing for administrators and end users and describes how to use the tools that are part of the standard FLEXnet Licensing distribution kit from Sybase.

These documents are available on the Getting Started CD and on the Web at Product Manuals at *http://www.sybase.com/support/manuals/*.

Configuring Sybase CEP Engine

After installing and licensing Sybase CEP Engine, you may need to change the default configuration preferences to improve performance or to enable certain features.

Introduction to Configuring Sybase CEP Engine

This section discusses general configuration issues relevant to Sybase CEP Engine.

Sources of information for configuring several Sybase CEP features include:

- This section that discusses general configuration issues relevant to Sybase CEP Engine.
- *"Configuring Sybase CEP Engine"* on page 35 groups several of the shorter configuration requirements and subjects in one chapter.
- For larger issues, such as enabling SSL security, see .
- For Enterprise Sybase CEP Engine feature configuration, see *"Enterprise Sybase CEP"* on page 83.

Editing Sybase CEP Engine Configuration Files

Many sections of this guide refer to configuration files which are used to set various Sybase CEP Engine preferences. This section includes descriptions of where to find configuration files based on the operating system you use, and how to set preferences within the configuration files.

In the case of Sybase CEP Server configuration, the two commonly-used configuration files are called `c8-server.conf` and `c8-services.xml`. These files are located in the `conf` subdirectory of the `Server` directory, in your `c8` installation directory.

For example, if you install Sybase CEP Engine on a UNIX-like operating system, in the `/usr/` *user-name* `/sybasec8` directory, then the configuration files reside in the `/usr/` *user-name* `/sybasec8/server/conf` directory.

If you install Sybase CEP Engine on Microsoft Windows, in the `C:\Program Files\SybaseC8` directory, then the configuration files reside in the `C:\Program Files\SybaseC8\Server\conf` directory.

Sybase CEP Server reads the relevant configuration files only at the time it starts. If something in a configuration file is changed, the Sybase CEP Servershould be stopped and restarted for the change to take effect.

Sybase CEP Engine configuration files use XML format to store lists of preferences and the values to which these preferences are set, as shown in the following example. This example sets the "Hostname" preference to the value of `myhost`like this:

```
<preference name="Hostname">myhost</preference>
```

1. Preferences in configuration files are often grouped together in sections and subsections.

   Each section controls related Sybase CEP Engine settings. When this guide mentions one of these preferences, it often also mentions the section in which the preference is located. For example, a configuration instruction in this guide may tell you to set the 'ConfigurationPath' property, in the 'ManagerCluster' subsection.
2. When preferences are contained in a section that is nested inside other sections, this guide lists the "path" to the preference, like this: set the 'ConfigurationPath' preference in the 'SybaseC8/Server/Manager/HighAvailability/ManagerCluster' section'.

   This means that the "ConfigurationPath" preference is located in the "ManagerCluster" section which is located in the "HighAvailability" section, and so on.
3. When you change the values of preferences, do not change the preference name or any other information inside the angle brackets (<>) except the value.
4. Some configuration file preferences appear enclosed in comment markings, which makes them inactive.

   Here is an example with the comment markings shown in bold face:

   ```
   <!-- <preference name="MinActiveContainersNative" value="0"/> -->
   ```

   If you are setting a value for a preference that is contained inside comments, be sure to remove the comment markings to activate the preference. The previous example would now look like this:

   ```
   <preference name="MinActiveContainersNative" value="1"/>
   ```

General Sybase CEP Engine Configuration Settings

This section contains information about configuring a variety of Sybase CEP Engine features.

About Configuring Sybase CEP Engine

For information about configuring SSL support, see *"Configuring SSL Support"* on page 39. For information about configuring Sybase CEP Engine plugins, see *"Configuring Sybase CEP Engine Plugins"* on page 46. For information about configuring Sybase CEP Engine services, see *"Configuring Sybase CEP Engine Services"* on page 52. For information on configuring Enterprise Sybase CEP Engine features, see *"Enterprise Sybase CEP Engine"* on page 35.

Before changing Sybase CEP Engine configuration settings, review the information in *"Introduction to Configuring Sybase CEP Engine"* on page 35.

Installing a License File

This section describes how to install a Sybase CEP license file.

To install a new license file:

1. Shut down Sybase CEP Server if it is running.
2. Copy the license file you received from Sybase into the **SYSAM-2_0\licenses** folder under the Sybase CEP Server installation folder.
3. Start Sybase CEP Server.

Specifying Sybase CEP Server Log Location

By default, the Sybase CEP Server installation configures Sybase CEP Server to write log messages to a file located in the `SybaseC8/server/logs/` folder.

You may change the default location of this log file by editing the "SybaseC8/Logging/LogWriters" section of the `c8-server.conf` file.

Enabling Sybase CEP Studio Logging

A description of how to enable Sybase CEP Studio Logging.

Follow these steps to turn on logging for Sybase CEP Studio:

1. Shut down Sybase CEP Studio.
2. Open the `studio-preferences.xml`file, located in the SybaseC8Repository, under the appropriate Sybase CEP Engine version number.
3. Scroll down to the bottom of this file.

   If Sybase CEP Studio logging has not yet been enabled, the last line of this file looks like this:

   ```
   </preferences>
   ```
4. Replace this line with the following:

   ```
   <preference name="C8/Logging/LogWriters/FileLogger/Type"
   value="File"/>
   <preference name="C8/Logging/LogWriters/FileLogger/Layout"
   value="Text"/>
   <preference
   name="C8/Logging/LogWriters/FileLogger/LogLevel"
   value="Debug2"/>
   <preference
   name="C8/Logging/LogWriters/FileLogger/DetailsLevel"
   value="Full"/>
   <preference
   ```

```
name="C8/Logging/LogWriters/FileLogger/Filename"
value="c:\log-file-name.log"/> </preferences>
```

where log-file-name is the name is the name of your log (for example, SybaseC8studio.log).

5. Exit the file and restart Sybase CEP Studio.

Increasing the Maximum Input Rate

The `c8-server.conf` file contains a preference that sets an upper limit on the rate at which an input adapter can send information to the Sybase CEP Server.

To override this limit, change the "MaxRate" preference in the "SybaseC8/Adapters" section of this file. The portion to update (the number 100000) appears in italics in this XML code example:

```

<section name="SybaseC8/Adapters">
  <section name="RandomTuplesGenerator">
       <preference name="MaxRate" value="
100000
"/>
```

Memory Usage

A description of modifiable preferences in the `c8-server.conf` file, including memory limit and memory allocation strategy.

The file `c8-server.conf` contains several preferences related to how Sybase CEP Server uses memory. The section "SybaseC8/Memory" includes several preferences, each of which is described by the comments in the file. The most significant preference is "Limit", which sets the maximum amount of memory Sybase CEP Server should use, identified either as a percentage of the total installed physical memory or as a specific amount. The default is 90%. Another modifiable preference is the memory allocation strategy, "MallocStrategy". The default, "Normal", attempts to balance memory usage with performance; "Aggressive" is more generous with memory allocation to increase performance;and "Conservative" is more frugal with memory to the possible detriment of performance. Do not modify the remaining memory allocation preferences unless directed to do so after consulting with Sybase CEP Support.

Internationalization Features

A description of internalization features in Sybase CEP Engine, including the ability to accept input from languages other than English, and the ability to enable runtime loading of localization files using i18n support.

International characters are processed in UTF-8 (8-bit Unicode Transformation Format). If you are processing Sybase CEP Engine data in languages other than English, make sure that your data is in UTF-8 format. No other special formatting or configuration is required to use this feature.

To make user visible messages written either to the screen or logfiles localized, full i18n support has been introduced to the product. i18n support helps promote localization by pushing localized messages into files that are loaded at runtime, and segregating localization from the build process.

To facilitate the runtime loading of localization files, use the following two environment variables:

1. C8_LANG: The locale to use, in other words, "en_US".

   "en_US" will be the default in the event that this envar is not specified.
2. C8_LANG_DIR: The directory where the locale specific directory resides, in other words, "/opt/sybasec8/bin/i18n".

   The final install location is still subject to change. In the event that this envar is not specified, "./i18N" will be used.

Note: Within C8_LANG_DIR, the product will look for a directory by the name of C8_LANG. Within this directory, there should exist property files containing locale specific messages.

SNMP Adapter-Related Configuration

To use an SNMP adapter with Sybase CEP Engine, add a Management Information Bases (MIB) section to the `c8-server.conf` file.

This section should contain the "MIBDIRS" preference, which is used by the SNMP adapter to determine where Sybase CEP Server should load MIBs and from where it should read MIBs. For instructions on settings these preferences, see the Sybase CEP *Integration Guide* .

Automatic Project Restart

Sybase CEP Engine projects can be set to automatically restart in case of a fatal error.

To enable this option, select the Automatically restart module on failure check box within the Sybase CEP Studio project properties. See the Sybase CEP *Studio Guide* for more information regarding this option.

Configuring SSL Support

You can configure Sybase CEP Engine to use Secure Socket Layer (SSL) communications to protect sensitive data. Describes several features that you can include as part of the Sybase CEP Engine SSL configuration.

Features include:

1. **Data encryption:** Data passed between the Sybase client applications (such as Sybase CEP Studio) and Sybase CEP Server are encrypted.

2. **SSL server authentication:** The SSL client requests that the SSL server identify itself.
3. **SSL client authentication:** The SSL server requests that the SSL client identify itself.
4. **Use of non-default communication protocols:** By default, the most secure protocol is automatically negotiated on each connection.

   However, you can restrict the protocols negotiated to only SSL2, SSL3 or TLS, or to any combination of these.
5. **Use of non-default encryption protocols:** You can change the default protocol settings.

Secure and insecure services cannot run together. You can choose to either enable or disable SSL on all active Sybase CEP Servers in your installation.

If you are using SSL with High Availability features, enable SSL and list an https: URI for all Sybase CEP Servers that are running Manager processes (as listed in the `c8-manager-cluster.xml` file). See *"Configuring Managers for Manager or Container High Availability"* on page 85 for more information about setting up High Availability features for Manager processes.

Enabling SSL Support

This section explains how to turn on SSL support.

Select the Enable SSL check box during installation on Microsoft Windows or, while running the installation script on Unix-like operation sustems, answer **`true`** to the question `Should I enable SSL support?`, while running the installation script on UNIX-like operating systems.

If Sybase CEP Server has already been installed without enabling SSL, modify the "EnableSSL" preference in your `c8-server.conf` file to **`true`** and change the Manager URI so that it starts with `https:` instead of `http:`.

For changes in the `c8-server.conf` file to take effect, restart Sybase CEP Server.

Sybase CEP Engine and SSL Client/Server Authentication

Describes SSL support and defines associated terms, "server" and "client".

In the context of SSL support, the word "server" has two meanings.

- Sybase CEP Server refers to the component of Sybase CEP Engine that processes queries.
- SSL server refers to any application that responds to requests from another application.

The word "client" also has two meanings when discussing SSL:

- Sybase CEP client refers to Sybase CEP Studio, or to another program that provides a way for you to interact with Sybase CEP Server. This section uses Sybase CEP Studio as an example of the Sybase CEP client, but SSL may also be enabled for other client applications that communicate with Sybase CEP Server: for example, the c8_client utility, Eclipse Plugin, or an application designed by you to replace Sybase CEP Studio.
- SSL client refers to any application that makes a request of another application.

When setting up client/server SSL features, keep in mind the following guidelines:

- Sybase CEP Server may act as both SSL client and SSL server when communicating with another Sybase CEP Server, and thus requires both server and client SSL features.
- Sybase CEP Studio always acts as an SSL client in communication with Sybase CEP Server, and requires only client SSL features.

Turning on SSL Server Authentication

A description of how to turn on SSL Server authentication after installation.

To turn on server authentication:

1. Enable SSL data encryption for Sybase CEP Engine, as described in *"Enabling SSL Support"* on page 40.
2. Import the proper server certificates into the Sybase SSL server's Certificate Database.

   The Certificate Database is located in the secure subdirectory of the `server` directory of your Sybase CEP Engine installation. Use the following command from your shell or MS-DOS prompt:

   ```
   certutil -A -n
   nickname-of-your-certificate
      -t "u,u,u"
      -d
   certificate-directory
    -i
   certificate-filename
   ```
3. Import the CA certificate into Sybase CEP Studio's certificate database, if you haven't done so already.

   Use the same command as in the previous step, but specify Sybase CEP Studio's secure directory instead of Sybase CEP Server's.

   The Sybase CEP Studio `secure` directory is located in the SybaseC8Repository. For example, on Microsoft Windows, the default location of the secure directory is:

   ```
   C:\Documents And Settings\
   user-name
   \My Documents\
      SybaseC8Repository\
   version
   \secure
   ```

   where *user-name* is the user name under which Sybase CEP Engine is installed and *version* is the version of Sybase CEP Engine (for example, 5.2.0).

   To specify a different secure directory for Sybase CEP Studio, add the following preference to the `studio-preferences.xml` file in the SybaseC8Repository:

   ```
   <preference name="SybaseC8/General/NSSFolder">
   ```

```
directory

</preference>
```

where

```
directory
```

is the secure directory you want to use.

4. Set the following preferences in the `c8-server.conf` file:
    a) Set the value of "ServerAuthenticate" preference, in the "SSL" section, to true.
    b) Add the nickname of your certificate to the configuration file, by setting the value of "ServerCertificate" in the "SSL" section to your Sybase CEP Server certificate's nickname.
    c) Make sure that the hostname of the "ManagerURI" exactly matches the SSL server certificate's Common Name (CN).

       The CN is the name of the machine or server identified by your certificate.
    d) Make sure that the "Hostname" value in the "Server/Common" section exactly matches the Server Certificate's Common Name.
5. In the SSL client's preferences file, set the value for the "ServerAuthenticate" preference to true.

   If you are using Sybase CEP Studio Sybase CEP Studio, this configuration file is called `studio-preferences.xml` and is located in the SybaseC8Repository. Add the following line to this file:

```
<preference name="SybaseC8/Security/SSL/ServerAuthenticate"
    value="true"/>
```

Turning on SSL Client Authentication

A description of how to turn on SSL client authentication after installation.

1. Enable SSL data encryption for Sybase CEP Engine, as described in section *"Enabling SSL Support"* on page 40.
2. Enable SSL server authentication, as described in section *"Turning on SSL Server Authentication"* on page 41.
3. Obtain and import client certificates into the SSL client Certificate Database using the certutil utility included with your Sybase CEP Server. To import the certificates:
    a) Copy the SSL client certificate to the secure directory you want the client to use. When determining which directory to designate as the secure directory, keep in mind that the SSL client searches for the client certificate in the first valid directory it finds. The client performs the search in this order:
        1. The client searches for the folder specified in the NSS folder preference section of the client's `preferences.xml` file.

   2. If it does not find a certificate there, the client searches for it in the secure directory located in the SybaseC8Repository.
   3. If it does not find a certificate there either, the client searches the client's secure directory .

   b) Navigate to the secure directory in which the certificate now resides.

   c) Import the certificate using the following command from your shell or MS-DOS prompt:

```
certutil -A -n
nickname-of-your-certificate
   -t "u,u,u"
   -d . -i
certificate-filename
```

   where *nickname-of-your-certificate* is a nickname you assign to the certificate for easy reference. For an explanation of certutil commands, execute the command

```
certutil -H
```

   from the command line prompt in the `server\bin` directory of your Sybase CEP Engine installation.

4. Use the `certutil` utility to import the client certificates into the server's certificate database. The procedure for this step is the same as for Step 3, but performed on the SSL server machine.
5. Set the following values in the server's `c8-server.conf` file:

   a) Set the value of "ClientAuthenticate" preference in the "SSL" section to **true**.

   b) Add the nickname of your SSL client certificate to the configuration file, by setting the value of "ClientCertificate" in the "SSL" section to your client's certificate nickname.
6. Add the following line to the SSL client preferences file (for Sybase CEP Studio, this file is called `studio-preferences.xml`, located in the SybaseC8Repository, under the appropriate Sybase CEP Engine version number):

```
<preference name="SybaseC8/Security/SSL/ClientCertificate"
   value="
nickname-of-client-certificate"/>
```

Changing the Default for Communication Protocols

A description of how to change the default for communication protocols. Sybase CEP enables you to restrict the SSL protocols negotiated on a secure connection to only the ones you specify.

You can restrict permitted protocols to SSL2, SSL3, or TLS. To change the default communication protocol that the SSL client uses to communicate with the SSL server, add the following line to the client's preferences file. For Sybase CEP Studio, this file is called `studio-preferences.xml`, and is located in the SybaseC8Repository:

```
<preference name="SybaseC8/Security/SSL/UseProtocol"
  value="
comma-separated-list-of-allowed-communication-protocols
"/>
```

To restrict the communication protocols negotiated by the SSL server when communicating with the SSL client, set the "AcceptProtocol" preference in the server's `c8-server.conf` file to a comma-separated list of the communication protocols you want the server to accept. If this preference is omitted, or left blank, then all permitted communication protocols are negotiated.

Changing Cipher Suite Encryption Defaults

A description of how to change default settings for cipher suite encryption data.

To change the default settings for the cipher suite encryption data sent from the SSL client to the SSL server, add the following line to the client's preferences file. For Sybase CEP Studio, this file is called `studio-preferences.xml`, located in the SybaseC8Repository:

```
<preference name="SybaseC8/Security/SSL/UseProtocol"
  value="
comma-separated-list-of-encryption-protocols
"/>
```

To change the cipher suite encryption data sent from the SSL server to the client, set the "AcceptEncryption" preference in Sybase CEP Server's `c8-server.conf` file to a comma-separated list of the encryption protocols for the server to use.

If this line is omitted or not included, then the cipher suite uses only the default encryption protocols.

Disabling SSL Support

A discription of how to disable SSL Support with the Sybase CEP configuration file.

To disable SSL features, set the "EnableSSL" preference in Sybase CEP Server's `c8-server.conf` file to **`false`** and change the "ManagerURI" preference from an **`https:`** address to an **`http:`** address.

Revoking an SSL Certificate

Describes how to revoke a Sybase SSL certificate. In some situations, specific SSL certificates may need to be revoked for security or administrative reasons.

To revoke a Sybase CEP SSL certificate:

1. Obtain a Certificate Revocation List (CRL).

   If the SSL certificate installed was provided by a third-party vendor, contact the vendor for instructions on obtaining the CRL. If the SSL certificate was not provided by a third-party vendor and you installed it yourself, then generate the CRL yourself, using a CRL tool such as crlutil from Mozilla.

2. If your SSL client or server authentication is turned on, create a `preferences.xml` file in one of the following locations:
   - The Sybase CEP Server `bin` directory.
   - The default SybaseC8Repository directory. On Microsoft Windows, this directory is:

     ```
     C:\Documents And Settings\
     user-name
     \
             My Documents\SybaseC8Repository\
     version
     ```

     On UNIX-like operating systems, this directory is:

     ```
     $HOME/SybaseC8Repository/
     version
     ```
   - The current working directory.
3. The `preferences.xml` file requires instructions to enable the c8_client utility to search the appropriate repository for the CA certificate (in the case of SSL server authentication) or the Client Certificate (in the case of SSL client authentication).

   Enter the following lines in this file:

   ```
   <preferences xmlns=
   "http://schema.sybase.com/preference/2004/05">
       <preference name="SybaseC8/General/NSSFolder"
      value="
   certificate-database-directory
   "/>
       <preference name="SybaseC8/Security/SSL/ServerAuthenticate"
      value="
   true-or-false
   "/>
       <!--<preference name="SybaseC8/Security/SSL/
   ServerCertificate"
      value="
   SSL-certificate-name
   "/> -->
       <!--<preference name="SybaseC8/Security/SSL/
   ClientCertificate"
      value="
   SSL-certificate-name
   "/> -->
       </preferences>
   ```

   where:

   - *certificate-database-directory* is the directory where the certificate database resides.
   - *true-or-false* is

```
true
```

if you want the client to perform server authentication against the server. Otherwise, set this value to

```
false
```

- *SSL-certificate-name* is the nickname of the CA Certificate (in the case of SSL server authentication) or the Client Certificate (in the case of SSL client authentication). Enter the certificate name under the "SybaseC8/Security/SSL/ServerCertificate" or "SybaseC8/Security/SSL/ClientCertificate", as appropriate, and remove the markings around the preference where the nickname was entered.

4. Use the c8_client command line utility to import the CRL.

This utility is located in the Sybase CEP Server `bin` directory. To import the CRL, make sure that Sybase CEP Server is running and give the following command:

```
c8_client --cmd=importCRL
   --server-uri=https://
hostname:port
/Server
   --crl-file=
crl-file-path-and-name
```

If the command cannot locate the CRL file in your current working directory, then

```
crl-file-path-and-name
```

should specify the full path and name of the file. Otherwise, you may simply specify the file name.

Configuring Sybase CEP Engine Plug-Ins

You can expand the functionality of Sybase CEP Server with one or more plugins. A *plugin* is an *event* handler.

When a particular event occurs, Sybase CEP Server invokes the plugin, which executes an action or a series of actions depending on the program invoked by the plugin. For a discussion of Sybase CEP Server events, see Sybase CEP Server Events.

The `c8-server.conf` Sybase CEP Server configuration file should include a separate configuration subsection for every instance of every plugin Sybase CEP Server should invoke. The corresponding plugin configuration subsection should be included under the "SybaseC8/Server/Manager/Plugins" section of the `c8-server.conf` file. The "Manager" section already includes a sample "Plugin" subsection. This example section can be customized to create a section for your actual plugin. New sections can also be added if you are configuring additional plugins.

This section describes how to configure:

- The general purpose command line plugin (which you may configure to invoke any process on the computer where Sybase CEP Server is running).
- The specialized plugin, used to generate SNMP traps for various Sybase CEP Server events.

Other sections of this guide contain instructions for configuring additional specialized plugins:

- If you are using Sybase CEP Engine access restriction features, you may configure a Sybase CEP Engine authentication plugin to check user-password matches and user membership in groups. To learn about the available options for this plugin and how to configure them, see "*Configuring Access Restrictions to Sybase CEP"* and the Sybase CEP *Integration Guide* .
- If you plan to send remote procedure calls to external applications, you will need to configure an RPC plugin. For more information about remote procedure calls, see "*Configuring Sybase CEP Engine Services"* of this guide. For detailed information for configuring the RPC plugin, see the Sybase CEP *Integration Guide* .
- If you are using High Availability Manager failover features, you will need to configure a plugin, which is used in the event of failover. This plugin is described in *"High Availability"* on page 83.

Configuring the Sybase CEP Engine General Purpose Plugin

A description of how to customize the appropriate section of the `c8-server.conf` file to configure the general purpose plugin.

Sybase CEP Engine includes a general purpose plugin, which invokes any executable program you specify in response to an event.

If necessary, you may configure multiple instances of the general purpose plugin to execute different programs in response to different events. You may specify as many event/program pairs as you wish for each Sybase CEP Server. To spawn a single event to more than one program, or to spawn one program from more than one event, a separate instance of the general purpose plugin should be configured for each invocation of each program in response to each desired event.

If you are configuring multiple instances of the plugin, follow the steps described here for each instance you are defining:

1. Create a subsection for the plugin configuration in the "SybaseC8/Server/Manager/Plugins" section of the `c8-server.conf` file.

   This section already contains a "SampleCommandLinePlugin" subsection that looks like this:

   ```
   <!--
       <section name="SampleCommandLinePlugin">
   ```

```
   <preference name="LibraryName" value="c8_server_plugins_lib"/>
    <preference name="InitializeFunction"
  value="c8_command_line_plugin_initialize"/>
    <preference name="ExecuteFunction"
  value="c8_command_line_plugin_execute"/>
    <preference name="ShutdownFunction"
  value="c8_command_line_plugin_shutdown"/>
    <preference name="MessageGroup" value="SampleMessageGroup"/>
    <preference name="MessageName" value="SampleMessageName"/>
   <preference name="CommandName" value="/usr/bin/some_command"/>
   <preference name="CommandArgument1" value="-a={MessageGroup}"/
>
   <preference name="CommandArgument2" value="-b={MessageName}"/>
    <preference name="CommandArgument3" value="-c={ObjectID}"/>
    <preference name="CommandArgument4" value="-d={Value}"/>
    <preference name="MaxRunningProcesses" value="10"/>
    <preference name="CommandTimeoutSeconds" value="5"/>
    </section>
    -->
```

You can either remove the markings from this section and customize it, as described in this section, or make a copy of the section below the original and customize that copy.

2. Rename the "SampleCommandLinePlugin" section.

   It is best to give the section a meaningful name. Use any name that is unique within the configuration file. For example:

   ```
   <section name="testplugin1">
   ```

3. Make sure that the "LibraryName" , "InitializeFunction" ,"ExecuteFunction", and "ShutdownFunction" preferences are set as shown here:

   ```
   <preference name="LibraryName"
       value="c8_server_plugins_lib"/>
       <preference name="InitializeFunction"
       value="c8_command_line_plugin_initialize"/>
       <preference name="ExecuteFunction"
       value="c8_command_line_plugin_execute"/>
       <preference name="ShutdownFunction"
       value="c8_command_line_plugin_shutdown"/>
   ```

4. Set the "MessageGroup" preference to the Sybase CEP Server event group that includes the event that you want to invoke the plugin.

   For a list of available Sybase CEP Server event groups and the events included in each group, see "Sybase CEP Server Events".

5. Set the "MessageName" preference to the name of the event that you want to invoke the plugin.

6. Set the "CommandName" preference to the full path and name of the executable program you want the plugin to execute.

Note: On Microsoft Windows, you can specify only the name of the program, without the path, if the path to the program appears in the %PATH% environment variable.

7. If you want the command specified in the "CommandName" preference to execute with command line arguments, specify the arguments with the "CommandArgument#" preferences, where # is a number from 1 to 1024.

a) The "SampleCommandLinePlugin" includes the first four "CommandArgument#" preferences with sample values.

Replace these with the actual arguments you want to pass to your program. You may also add additional "CommandArgument#" preferences as necessary or delete extra preferences, as needed.

Use the "CommandArgument1" preference to specify the first argument you want to pass to the command, then specify any subsequent arguments in numeric order: using preferences "CommandArgument2", "CommandArgument3" and so on. Sybase CEP Server ignores any arguments that are specified out of order.

b) Use a separate "CommandArgument#" preference for every command line argument.

For example, configure a shell command such as this:

```
command
 -a -b -c -d "
-quoted value
"
```

like this:

```
...
        <preference name="CommandArgument1" value="-a"/>
        <preference name="CommandArgument2" value="-b"/>
        <preference name="CommandArgument3" value="-c"/>
        <preference name="CommandArgument3" value="-d"/>
        <preference name="CommandArgument4" value="-quoted
value"/>
        ...
```

The quotation marks in this example are optional. Sybase CEP Engine automatically adds quotation marks around values that require them, so you do not have to explicitly indicate quotation marks in the argument configuration.

c) The value of any command argument preference may contain the following pre-defined special names.

Sybase CEP Engine replaces these names with the corresponding values for every invocation:

```
{SourceTimestamp}
```

is replaced by the time at which the event took place.

```
{MessageGroup}
```

is replaced by the message group of the event.

```
{MessageName}
```

is replaced by the message name of the event, as it occurs within its message group.

```
{ObjectID}
```

is replaced by the URI of the object to which event applies.

```
{Value}
```

is replaced by the value associated with the event.

d) Since the curly bracket characters ({ }) have special meanings, as shown in the previous bullet item, preface any other occurrences of curly brackets with the \ escape character.

For example:

```
...
        <preference name="CommandArgument1" value="\{a\}"/>
        ...
```

8. Set the desired value for the "MaxRunningProcesses" preference, or leave the value at its sample setting of ten (10).

This preference works in conjunction with the "CommandTimeoutSeconds" preference, and allows you to set the maximum number of event-handling programs spawned by the current instance of the plugin that may execute at the same time.

If a new event arrives when the maximum number of programs are running, Sybase CEP Server attempts to terminate any running event-handling processes that have either finished execution, or have reached the maximum time limit specified by the "CommandTimeoutSeconds" preference. If none of the running processes can be terminated, the new process does not launch.

If you delete this preference, the number of maximum running process defaults to 32.

9. Set the "CommandTimeoutSeconds" preference, or leave it at its sample setting of 5.

This works in conjunction with "MaxRunningProcesses" and *takes effect only when the indicated maximum number of processes are running*. Once the number of running processes reaches the maximum indicated by "MaxRunningProcesses", Sybase CEP Engine terminates all processes launched previously by the current instance of the plugin that have been running longer than the number of seconds indicated by "CommandTimeoutSeconds".

If you delete this preference entirely, "CommandTimeoutSeconds" defaults to 60. A value of 0 indicates that processes should never be timed out.

Note: Keep in mind these considerations when you are setting the timeout preferences intervals for events.

- Most event-handling programs take less than one second to process events that regularly occur once per second: for example notifications about CPU consumption.
- Events that occur infrequently, such as Container failure, usually do not occur in groups of more than one, so, in most cases, the maximum running processes and timeout interval is irrelevant. However, Sybase suggests that you consider the maximum number of such events that could occur within a specified time, and set the preferences accordingly.

Sybase CEP Server Monitoring with SNMP Traps

Sybase CEP Server supports a highly customizable SDK solution, including the Sybase CEP Server Plugin SDK and a command line Sybase CEP Server plugin that uses the SDK.

With these features, you can configure the command-line plugin to use the Net-SNMP snmptrap tool to generate SNMP traps on various Sybase CEP Server events. To configure SNMP traps:

1. Install Sybase's custom Management Information Base (MIB) for use with your Net-SNMP tools. You can do this using several methods, one of which is described here:
   a) Copy the `C8-SERVER-STATUS-MIB.txt` file from the Sybase CEP Server `conf` directory to the *NET-SNMP-install-directory* `/share/snmp/mibs/` folder, where *NET-SNMP-install-directory* is the directory where NET-SNMP is installed.
   b) Add the following line to *NET-SNMP-install-directory* `/etc/snmp/snmp.conf`:

   ```
   mibs +C8-SERVER-STATUS-MIB
   ```
2. Open the `c8-server.conf` file and remove the marks from the line that lists the `c8-server-snmp.conf` file.
3. Open the `c8-server-snmp.conf` file and modify the "SNMP_ManagerEvent1" section, as described in the next step. For an additional trap on an event, make a copy of the "SNMP_ManagerEvent1" section inside the file, assign it a unique name, then modify the new section.
4. Set the following preferences in the section you have identified or created in the previous step:
   a) Set the "MessageGroup" and "MessageName" preferences to a valid Sybase CEP Server message group and message name, respectively.

      For a list of valid Sybase CEP Server groups and messages, see the comments in the `c8-server-snmp.conf` file or *"Sybase CEP Server Events"* on page 131.

      You can configure a single preferences section to handle only one type of Sybase CEP Server event. If you need to send traps on multiple Sybase CEP Server events, duplicate the section, give each copy a new unique name, and configure each copy to handle a different event. The number of sections in your configuration file should be the same as the number of events for which you want to generate SNMP traps.

b) Set the "CommandName" preference to the full path of the snmptrap command line tool from Net-SNMP.

For example:

```
        C:\Windows\System32\snmptrap.exe
```

c) Set the "CommandArgument5" preference to the name of the SNMP community to which you want to send the trap.

Most of the time this name will be set to either to **`public`** or to **`sybasec8`** .

d) Set the "CommandArgument6" preference to the host name where the snmptrapd trap daemon receiver is running.

While testing the SNMP trap features, set this preference to the local host name.

e) Set the "CormmandArgument8" preferences to the host name of the Manager or Container from which the trap is sent.

To test whether traps are sent properly, you may configure the Net-SNMP's snmptrapd daemon to receive traps. The simplest way to do this is to use the following command on the machine configured to receive the traps:

```
snmptrapd -Lo -F "\n\n%W(%w) TRAP RECIEVED FROM
   %A(%a)\nEPOCH:%t\nVALUES:\n%V\n%v\n"
```

For more information about SNMP traps, see the `c8-server-snmp.conf` file.

Configuring Sybase CEP Engine Services

The `c8-services.xml` configuration file contains preferences related to interfacing Sybase CEP Engine with other systems, including remote database servers, public windows, remote procedure call servers (RPC servers), and others.

In this chapter you will learn how to configure the various services in this file.

Note: If you change the contents of this file, Sybase CEP Engine automatically reloads the file and incorporates your changes

Enabling External Database Access

Use adapters and CCL queries to set up communication between Sybase CEP Engine and conventional databases.

- Adapters (except in the case of kdb+ databases, for which Sybase does not support adapter use). For extensive information about adapters, see the Sybase CEP *Integration Guide* .

- CCL queries, using the CCL database subqueries and Database Statements.
  - Database subqueries are subqueries within CCL Query Statements that retrieve data from external relational databases or CCL public windows by means of an SQL **SELECT** statement. Database subqueries do not modify the external databases.
  - Database Statements are CCL statements that send SQL **UPDATE**, **INSERT** and/or **DELETE** statements to an external relational database. Database Statements modify the data in external databases, but do not receive any data from the databases.

  For information about database subqueries and Database Statements, see the Sybase CEP *CCL Reference Guide* .

In order for Sybase CEP Engine to communicate with databases using either of these features, you:

- Should have a working ODBC or native Oracle OCI connection.
- Modify the `c8-services.xml` file with the appropriate configuration information, as described in the database-related sections of this guide. This file contains groups of preferences, some of which relate to both adapter and CCL statement connections to databases, and other of which are applicable only if you are connecting to the database directly from CCL statements.
- If you are using Oracle OCI with Sybase CEP Engine, you should also install and configure Oracle Client.

Before Setting up Sybase CEP Engine for Database Connections

This section describes the preliminary steps you must complete before you try to establish communication between Sybase CEP Server and an external database, except if you are using kdb+.

The Sybase CEP kdb+ driver is automatically installed when you install Sybase CEP Server. If you are using kdb+, skip this section and proceed to *"Configuring Connections to Databases with User Name, Password, and DSN"* on page 55.

1. Install the appropriate ODBC or native Oracle driver for the external database you want to query. Contact the appropriate vendor to obtain the driver.

   Note: If you are planning to connect to an Oracle database, keep in mind that Sybase CEP Engine does not support stored procedure calls that return values when using the native Oracle driver.

   If you have installed an Oracle native driver, skip the rest of the steps in this section.

   If you have installed an ODBC driver on a UNIX-like operating system, proceed to the next step.

2. If you are using an ODBC driver on a UNIX-like operating system, install the ODBC Driver Manager, if it is not already installed. The Driver Manager controls the access to the ODBC drivers.

Sybase CEP Engine has been tested and is supported with the unixODBC Driver Manager. It should also work with the iODBC Driver Manager, but this configuration has not been tested.

For more information on UNIX ODBC, see: *http://www.easysoft.com/developer/interfaces/odbc/linux.html*.

3. If you are using an ODBC driver on a UNIX-like operating system, register the installed driver with the Driver Manager. You can either use the ODBCConfig graphical program, or the **`odbcinst`** command, both of which come with unixODBC, to register the program. The instructions in this section assume that you are using the `odbcinst` command:

a) Create a file called `tds.driver.template` in the `/usr/local/etc` directory.

b) Add a few lines describing the driver to this file. For example:

```
[FreeTDS]
        Description = v0.64 with protocol v8.0
        Driver = /usr/local/freetds/lib/libtdsodbc.so
```

This example configures an SQL Server connection on a UNIX-like operating system, using the FreeTDS driver. If you are using a different database, your entry should correspond to your driver.

c) Execute **odbcinst** , using one of the commands listed in this step, to install a driver entry using the `tds.driver.template` file. (Use the **odbcinst -help** command for instructions on which of the two commands to use). The commands are:

```
odbcinst -i -d -t tds.driver.template
```

with a space between the **`-t`** flag and the file name, or:

```
odbcinst -i -d -f tds.driver.template
```

4. (Optional). If you are using ODBC, you may create a Database Source Name (DSN) for each database to which you want to connect, however, Sybase CEP Engine allows connections to the database with or without a DSN. If you do not want to create a DSN, skip this step and proceed to *"Enabling External Database Access"* on page 52.

- Creating the DSN on Microsoft Windows:

  1. Open Windows Control Panel.

  2. Double-click Administrative Tools.

  3. Double-click Data Sources (ODBC).

  4. Click the Add button to add a system, or click user DSN and select the appropriate ODBC driver from the displayed list.

  5. Click the Finish button once you have selected the driver.

  6. Provide the appropriate database connection information on the page that appears. The specific page depends on the driver you select.

  When the installing program prompts you to provide the Data Source Name, enter the name you want to use to access the database. This is also the name you will provide when setting the DBDriverConnectString value in the `c8-services.xml` file.

- Creating the DSN on a UNIX-like operating system:

  Add a DSN entry to the `odbc.ini` file, for every system DSN that you or any other user plans to access from your system, or add a user DSN entry to the `.odbc.ini` file, for every user DSN that a specific user will access.

  If you are using unixODBC 2.2.12, the default location of the `odbc.ini` is in the `/usr/local/etc` directory. The default location of the `.odbc.ini` is in the user's home directory. The contents of every DSN entry depend on the ODBC driver that is required by the specific data source.

  Note that the Driver line of the DSN entry in the `odbc.ini` or `.odbc.ini` file must match the appropriate section entry in the `odbcinst.ini` file, which describes all the available installed ODBC drivers. The `odbcinst.ini` file is also located in the `/usr/local/etc` directory.

Configuring Connections to Databases with User Name, Password, and DSN

The most common way of configuring CCL queries to interact with an external relational database includes a database user name, password, and Database Source Name (DSN).

To set up this type of connection from within Sybase CEP Engine:

1. In the `c8-services.xml` file, create a separate service entry section for every external database to which you want to connect. This entry provides information to enable Sybase CEP Server to communicate with the external database server. If you want to connect to multiple database servers, or to multiple databases on a single server, create a separate section for each database.

   The `c8-services.xml` file contains several sample database service entry sections, each for a different type of database service. To find these entries, search for the string

   ```
   Type="DATABASE"
   ```

   inside the file. You can either customize one of these sections, or make a copy of it elsewhere in the file and customize the copy.

2. Set the "Service Name" preference to a unique service name for the remote service. This name is case-sensitive, must begin with a letter, and may contain a character string consisting of letters, numbers, underscores, dots, and/or colons. The service name should be the same name as the name of the database you use in your CCL queries. For example, if your database subquery is:

   ```
   ...
       FROM (DATABASE "Database1" SCHEMA "valuation.ccs"
       SELECT * FROM Stocks WHERE T.symbol=Stocks.symbol)
   ```

The database name (`Database1`) in the database subquery should match the service name in the `c8-services.xml` file.

3. Set the "Type" preference to **DATABASE**.
4. Optionally, add a description of your service entry in the "Description" element.
5. Set the "DBDriverType" preference to the type of driver you use to communicate with your external database. Sybase supports four the following driver types:
   - DBDriverOracle, which you should use if you are connecting to the database using the native (non-ODBC) Oracle drivers. The connect string for "DBDriverOracle" can be an Oracle TNS Name or an EZ Connect string.
   - DBDriverODBCSybaseIQUnix, which should be used if you are connecting to the database using a Sybase IQ driver on UNIX-like operating systems.
   - DBDriverODBC, which should be used if you are connecting to the database using an ODBC driver, including a Sybase IQ driver on Microsoft Windows. If you are setting up an ODBC connection that uses the DSN name as the connect string name, set up a DSN. Refer to your ODBC documentation for instructions on installing and setting up the necessary ODBC driver(s) for your database.
   - DBDriverKDB, which you should use if you are connecting to a kdb+ database.
6. If necessary, include a "DatabaseType" preference in your service entry. Set this preference if you are connecting to the Oracle TimesTen database. This is the only way you can connect to the TimesTen database from Sybase CEP Engine. This preference can only be used in conjunction with a "DBDriverType" setting of DBDriverODBC. It may improve performance and compatibility for database drivers that do not fully implement the ODBC standard, or that provide features or optimizations that are not part of the standard. This preference can be set to the following values:
   - MyODBC35.
   - PostgreSQL.
   - SQLServer2005.
   - TimesTen.

   Here is an example of a "DatabaseType" setting:

   ```
   <Param Name="DatabaseType">TimesTen</Param>
   ```
7. Set the "DBDriverConnectString" preference to the information needed to actually connect to the external database.

   For DBDriverOracle, the "DBDriverConnectString" may be the Oracle TNS Name, or may use the following format:

   ```
   dbserver:port/OracleSID
   ```

   where:

   - *`dbserver`* is the name of the computer on which the other server resides.
   - *`port`* is the port number used to communicate with the other server.

- *`OracleSID`* is the service name of the Oracle database to which you want to connect.

For DBDriverODBC and DBDriverODBCSybaseIQUnix, the "DBDriverConnectString" is the name of the DSN entry you have set up for your connection in the ODBC control panel.

For DBDriverKDB, the "DBDriverConnectString" is the name of the host name of the kdb+ server.

Note: Most database servers allow a single server to contain multiple databases. If you want to perform queries on more than one database on a given server, you may need to configure a separate connection for each database.

Note that, if your "DBDriverConnectString" contains one of the following characters, the character must be preceded by a back slash escape character (\). These characters are: semicolon (;), forward slash (/), question mark (?), colon (:), at sign (@), ampersand (&), equal sign (=), plus sign (+), dollar sign ($), and comma (,). For kdb+ services, set the "Port" preference to the port number of the kdb+ server.

8. Optionally, set the "CacheMaximumAge" preference to specify how long Sybase CEP Server should keep a cached copy of the information that it read from the external database.

For more information on caching, see *"Caching Data from an External Database, RPC Server, or Public Window"* on page 73.

9. Set the "Username" preference to the user name that should be used when communicating with the external database.

Note that this value is unencrypted, so anyone with access to the `c8-services.xml` file may read the user name.

10. Set the "Password" preference to the password for the user name you specified in the "Username" preference.

Like the user name, the password is unencrypted.

11. Set any desired optional preferences for your database service entry in the `c8-services.xml` file.

These are discussed further in *"Setting Optional Preferences for Services"* on page 71.

Here are three examples of how you might set the configuration preferences.

This is an example of an entry that uses the Oracle OCI driver:

```
<Service Name="OracleDb" Type="DATABASE" >
 <Description>local oracle dod db</Description>
 <Param Name="DBDriverType">DBDriverOracle</Param>
 <Param Name="DBDriverConnectString">myhost:6789/mydb</Param>
 <Param Name="CacheMaximumAge">0</Param>
 <Param Name="Username">myuser</Param>
 <Param Name="Password">mypass</Param>
</Service>
```

This example uses an ODBC driver. Note the different value for DBDriverType:

```

<Service Name="MyDB" Type="DATABASE" >
 <Description>sample odbc db</Description>
 <Param Name="DBDriverType">DBDriverODBC</Param>
 <Param Name="DBDriverConnectString">MyDSNName</Param>
 <Param Name="CacheMaximumAge">0</Param>
 <Param Name="Username">myuser</Param>
 <Param Name="Password">mypass</Param>
</Service>
```

This example uses a Sybase IQ driver. Note the different value for DBDriverType:

```

<Service Name="MyIQDB" Type="DATABASE" >
 <Description>sample sybase iq db</Description>
 <Param Name="DBDriverType">DBDriverODBCSybaseIQUnix</Param>
 <Param Name="DBDriverConnectString">MyDSNName</Param>
 <Param Name="CacheMaximumAge">0</Param>
 <Param Name="Username">myuser</Param>
 <Param Name="Password">mypass</Param>
</Service>
```

Finally, here is an example using the kdb+ driver:

```

<Service Name="MyKdbService" Type="DATABASE" >
 <Description>sample kdb</Description>
 <Param Name="DBDriverType">DBDriverKDB</Param>
 <Param Name="DBDriverConnectString">myhostname.mydomain.com</Param>
 <Param Name="Port">12345</Param>
 <Param Name="CacheMaximumAge">0</Param>
 <Param Name="Username">myuser</Param>
 <Param Name="Password">mypass</Param>
</Service>
```

If you are using a driver other than the ones discussed here, consult the driver documentation for the correct database driver connect string.

Make sure that the external database to which you are connecting is running properly before attempting to connect to it from Sybase CEP Engine.

Configuring ODBC Databases Service Entries without User Name or Password

Describes how to configure ODBC database service entries in the `c8-services.xml` file without specifying the user name and password. This type of configuration is useful when you have already specified the user name and password elsewhere, such as in the DSN.

To set up this type of service entry, follow all the steps in *"Configuring Connections to Databases with User Name, Password, and DSN"* on page 55 with the following exceptions:

- Specify the "DBDriverConnectString" preference using the following format:

```
<Param Name="DBDriverConnectString">DSN=
connect-string
;</Param>
```

The prefix DSN and semicolon (;) at the end of the connect string are required when creating this type of service entry. Specifying only the connect string will generate an error message.

Omit the "Username" and "Password" preferences from the entry.

Configuring ODBC Database Service Entries without a DSN

A description of how to configure ODBC service entries without a DSN.

When you configure this type of connection, you do not define the DSN or specify the user name and password in separate preferences. Instead, all connection-related information is collected in a single connect string.

For example, a `c8-services.xml` entry for an SQL Server database may look like:

```
<Service Name="MyDSN" Type="DATABASE" >
<Param Name="DBDriverType">DBDriverODBC</Param>
<Param Name="DBDriverConnectString">Driver=
   {SQLServer};Server=myServerAddress;Database=myDataBase;
   Uid=myUser name;Pwd=myPassword;</Param>
<Param Name="CacheMaximumAge">0</Param>
...
</Service>
```

For connect string format requirements for different databases, see *http://www.connectionstrings.com*. All other service entry preferences are described in *"Configuring Connections to Databases with User Name, Password, and DSN"* on page 55.

Before connecting to ODBC databases by this method, Sybase recommends that you first make sure that your computer's client library can find the ODBC driver (which is not part of the CEP Engine software suite). On Microsoft Windows the driver is located in the registry, while on UNIX-like operating systems, the driver is located in the `odbcinst.ini` file. Since your database vendor may provide utilities for connecting to your ODBC database that don't themselves use ODBC, Sybase recommends that you test the ODBC connection through a universal data access (UDA) tool.

Enabling Access to an ASE Database

Configure the Sybase CEP Server to communicate with an ASE database.

Enable access to an ASE database by configuring the ASE Database driver. Use only supported ASE data types to communicate with the ASE Database.

Configuring the ASE Database Driver

Configure the Sybase CEP Server to communicate with an ASE database.

Set up the Sybase CEP Server to communicate with an ASE database by configuring the `c8-server.conf` and `c8-services.xml` files. Before starting the CEP server, source the Sybase ASE/OCS environment (SYBASE.csh/sh) in the console and ensure Sybase Open Client (OC) version 15.5 is installed.

Note: ASE v15.5 is installed with the necissary OC elements, so any workstation with both ASE and Sybase CEP installed will be fully functional. If you do not have ASE or Sybase Open Client installed on your Sybase CEP workstation, you have a number of options depending on your operating system:

- Windows users can install the ASE v15.5 Network Client, which includesOC v15.5 . This client is only available for Windows platforms.
- Unix users can install the OC components from ASE v15.5 or download and install the Sybase Developers Kit (SDK) v15.5, which contains the same components.

To configure the ASE Database Driver:

1. Open the `c8-server.conf` configuration file.
2. Locate the <section name> element.

   ```
   <section name="C8/General"><section name="Plugins">
   ```

   Within this section, add the libary:

   ```
   <preference
   name="AdditionalLibraries">c8_db_driver_sybase_ocs_lib</
   preference>
   ```
3. Save and close `c8-server.conf`.
4. Open `c8-services.xml`.
5. Add:

   ```
   <!-- Sybase CEP OCS Driver - testing -->

      <Service Name="SybaseC8_OCS_Driver" Type="DATABASE" >

        <Description>Sybase CEP OCS Database</Description>

        <Param Name="DBDriverType">DBDriverSybaseOCS</Param>

        <Param Name="DBDriverConnectString">pdrepsybase 5000</Param>

        <Param Name="Username">user</Param>

        <Param Name="Password">password</Param>

        <Param Name="Timezone">timezone that exists in server/
   plugins/c8_timezones.csv</Param>
   ```

```
     <Param Name="OCSDatabaseName">dbname</Param>

     <Param Name="OCSAppName">appname</Param>

     <Param Name="OCSConfigFile">filename</Param>

     <Param Name="OCSServerAddress">hostname port</Param>

     <Param Name="OCSServerName">servername</Param>

     <Param Name="OCSNoTransactions">true/false</Param>

     <Param Name="OCSFetchMultipleResultSets">true/false</Param>

     <Param Name="OCSInlineTimestampParamFormat">timestamp
format</Param>

  </Service>
```

6. Define the following parameters:
   - Timezone – specifies the timezone the ASE database is running in. If set to a timezone other than GMT, the timestamp value will be corrected by the standard time GMT offset.
   - OCSDatabaseName – (optional) sets the database name.
   - OCSAppName – (optional) specifies the stanza of an external OCS configuration file to read for properties and options.
   - OCSConfig File – (optional) allows an external OCS configuration file to be used. If this value is set and the file does not exist, an error occurs. If this value is not set, a platform-specific default file is used.

     On UNIX platforms, the default configuration file is `$SYBASE/$SYBASE_OCS/config/ocs.cfg`.

     $SYBASE is the path to the Sybase installation directory, specified in the SYBASE environment variable. *$SYBASE_OCS* is the Open Client and Open Server subdirectory, specified in the SYBASE_OCS environment variable.

     On Windows platforms, the default configuration file is `%SYBASE%\%SYBASE_OCS%\ini\ocs.cfg`.

     %SYBASE% is the path to the Sybase installation directory, specified in the SYBASE environment variable. *%SYBASE_OCS%* is the Open Client and Open Server subdirectory, specified in the SYBASE_OCS environment variable.

     CT_ OPTIONS that govern transaction isolation and transaction chaining can be set in one of the platform-dependant external configuration files listed above.

     Do not set the option CS_NETIO=CS_ASYNC_IO in the external configuration file. It is possible to set CS_NETIO=CS_SYNC_10, but this causes the “abort" feature for long-lived calls not to function. The driver sets the default value for this option to CS_NETIO=CS_DEFER_IO.

If the driver detects that CS_OPT_CHAINXACTS is true, the driver does not issue “begin transaction” commands.

- OCSServer Address and OCSServerName – specifies the server to connect to. Only one of the parameters is necessary. Values set in the external configuration file overrride these settings.
- OCSNoTransactions – (optional) prevents the driver from issuing transaction commands. This is useful for achieving “autocommit” functionality. The default value is false, which means the driver can issue transaction commands.
- OCSFetchMultipleResultSets – (optional) governs whether multiple result sets are fetched at once. All result sets must have identical schema. The default value is false, which means the result sets are fetched independently.
- OCSInlineTimestampParamFormat – (optional) specifies a string format to convert the timestamp into when parameter inlining is required for an SQL statement, such as "exec sp_some_stored_proc ?CepVar". The default value is YYYY-MM-DD HH24:MI:SS.FF.

Note: If you are working on a Solaris platform, ensure that the LD_LIBRARY_PATH_64 environment variable is not set. Setting the variable will cause an error.

For more information on database drivers and external configuration files, consult the *Open Server 15.5 Client/Library C Reference Manual.*

Supported Data Types

Use only supported ASE data types to communicate with the ASE Database.

The ASE database driver supports a number of data types. The following table lists the supported data types and how they convert to CCL types.

ASE v15.5 Data Type	CCL Type
REAL	FLOAT
DOUBLE PRECISION	FLOAT
DATE	TIMESTAMP
TIME	TIMESTAMP
SMALLDATETIME	TIMESTAMP
DATETIME	TIMESTAMP
SMALLMONEY	FLOAT
MONEY	FLOAT
CHAR	STRING
VARCHAR	STRING

UNICHAR	STRING
UNIVARCHAR	STRING
NCHAR	STRING
NVARCHAR	STRING
DECIMAL	FLOAT
NUMERIC	FLOAT
VARBINARY	STRING
SMALLINT	INTEGER
INTEGER	INTEGER
TINYINT	INTEGER
BIGINT	LONG
UNSIGNED SMALLINT	INTEGER
UNSIGNED INT	INTEGER
UNSIGNED BIGINT	LONG
FLOAT	FLOAT

Enabling Access to Public Windows as External Databases

A description of how to use CCL-enabled public windows to communicate with external databases.

The public windows feature creates a special CCL window, with which you may do the following:

- Use a public window in CCL statements in all the same ways you can use a regular named window (see "Create Window Statement" in the Sybase CEP *CCL Reference Guide* .) A public window used in this way should be located in the same project as the CCL queries that use it.
- Query a public window with Sybase CEP SQL queries, from one of several locations:
    - From Sybase CEP Studio. See the Sybase CEP *Studio Guide* for more information. This method may only query public windows contained in the main module or submodules of the project in which you are working.
    - From external applications using Sybase CEP SDKs or from the c8_client command-line utility. For more information, see the Sybase CEP *Integration Guide* .

- From a database subquery. This method of querying a public window treats the public window as a "table" in an external "database". The public window you are querying may be located in the same project as the database subquery, or in a different project.
- From DB and PollFromDB adapters, which are described in the Sybase CEP *Integration Guide* .

For more information about Sybase CEP SQL, see the Sybase CEP *CCL Reference Guide* .

If you plan to access a public window with adapters, CCL database subqueries, or one of the Sybase CEP SDKs, you must first create one or more database service entries for the Sybase CEP Engine projects containing the public windows, as described here.

1. For every project that contains a public window to which you want to connect, add a database service entry to the `c8-services.xml` file.

The `c8-services.xml` file contains a sample service entry section for this purpose. You can either customize this section, or make a copy of it elsewhere in the file and customize the copy.

Here is the sample public window database service entry that appears in the `c8-services.xml` file. As with other examples in this guide, some items appear in italics. These are stand-in values, for which you must provide the actual information. For example, replace

```
user-name
```

with the appropriate user name.

```
<!-- Sample Service Configuration for Public Window  -->
    <Service Name="
MyPublicWindowDB
" 1 Type="DATABASE"2 >

      <Description>
        Sample Service Configuration for DB Subqueries (Public
Windows)
      </Description>3
     <!-- DBDriverType should be set to DBDriverPublicWindows -->
      <Param Name="DBDriverType">DBDriverPublicWindows4</Param>
      <!-- DBDriverConnectString should be set to:
           ccl://<your-manager-host>:<your-manager-port>/Project/
                 <your-workspace-name>/<your-project-name>
           with <your-workspace-name> and <your-project-name>
           corresponding to the workspace and name of the
           top-level module containing the Public Window(s)
           that you want to query -->
      <Param Name="DBDriverConnectString">
   \
              ccl://mymanager:1234/Project/MyWorkspace/MyProject5
      </Param>
      <!-- Set to an integer > 0 to enable caching -->
      <Param Name="CacheMaximumAge">06</Param>
      <!-- Set your Username and Password Appropriately if
        your server configuration requires authentication
      -->
      <Param Name="Username">
user-name7
</Param>
      <Param Name="Password">
password8
</Param>
    </Service>
```

1	Provide a unique service name for the public window service in the "Service Name" preference on this line. This name is case-sensitive, must begin with a letter and may contain letters, digits, underscores, dots and/or colons. The service name should be the same name as the name of the "database" service you use in your CCL queries. For example, if your database subquery is:
	`...`
	`    FROM (DATABASE "PublicWindow1" SCHEMA "valuation.ccs"`
	`    SELECT * FROM Stocks WHERE T.symbol=Stocks.symbol)`
	The database name (**PublicWindow1**) in the database subquery should match the service name in the `c8-services.xml` file.

2	Set the "Type" preference on this line to **`DATABASE`** .
3	Optionally, add a description of your service entry in the "Description" element.
4	Set the "DBDriverType" preference on this line to DBDriverPublicWindows. This preference indicates the type of driver used to communicate with the external "database".
5	Enter the ccl: URI of the project that contains the public window in the "DBDriverConnectString" preference on this line.
6	This line allows you to specify how long the Sybase CEP Server should keep a cached copy of the information that it read from the public window. For more information about caching, see *"Caching Data from an External Database, RPC Server, or Public Window"* on page 73.
7	If the public window being set up as a service is using the Sybase CEP Server's user authentication feature, indicate the user name that Sybase CEP Engineshould use when communicating with the Sybase CEP Server that contains the public window in the "Username" preference on this line. Note that this value is unencrypted: anyone who has access to the `c8-services.xml` file will be able to read the user name.

8	If the public window being set up as a service is using the Sybase CEP Server's user authentication feature, indicate the password that Sybase CEP Engine should use when communicating with the Sybase CEP Server that contains the public window in the "Password" preference, on this line. Like the user name above it, the password is unencrypted.

If you want to connect to public windows in more than one project, create a separate section for each project.

Here is an example of how you might set the configuration preferences.

```
<Service Name="MyPublicWindowDB" Type="DATABASE">
     <Description>Sample Service Configuration for DB Subqueries
(Public Windows)</Description>
     <!-- DBDriverType should be set to DBDriverPublicWindows -->
      <Param Name="DBDriverType">DBDriverPublicWindows</Param>

      <Param Name="DBDriverConnectString">ccl://mymanager:1234/
Project/MyWorkspace/MyProject</Param>
      <!-- Set to an integer > 0 to enable caching -->
      <Param Name="CacheMaximumAge">0</Param>
      <!-- Set your Username and Password Appropriately if
        your server configuration requires authentication
      -->
      <Param Name="Username">Helen</Param>
      <Param Name="Password">Troy450BC</Param>
    </Service>
```

2. Set any desired optional preferences for the service entries you just created in the `c8-services.xml` file.

   Optional preferences are discussed further in *"Setting Optional Preferences for Services"* on page 71. (Note that "MaxCallExecutionTime" and "DisableAbort" preferences do not apply to queries of public windows.)

Make sure that the project to which you are connecting is running properly before attempting to connect to it from another project.

Enabling Remote Procedure Calls

A description of the software components you need to use Sybase CEP Engine RPC functionality and how the components work together.

Sybase CEP Remote Procedure Call (RPC) functionality allows CCL statements to execute a function or procedure that is running outside the Sybase CEP Server process, either on the same computer as the Sybase CEP Server or on a different computer. For information about remote procedure call plugins and instructions for creating your own plugins, see the Sybase CEP *Integration Guide* .

To use the Sybase CEP Engine RPC functionality, install and use the following software components:

- An external executable program that includes a procedure or function that can be called remotely. The program must be capable of receiving and responding to requests from external applications. This guide refers to such programs as RPC servers. The RPC server may run on the same computer as Sybase CEP Server or on a different computer.
- A running Sybase CEP Server instance.
- An RPC plugin that interfaces between Sybase CEP Server and the RPC server.
- A running Sybase CEP Engine project that includes a CCL Remote Procedure Statement or remote subquery that invokes the plugin. For more information about this CCL statement and subquery, refer to the Sybase CEP *CCL Reference Guide* .
- A service entry section of the `c8-services.xml` file that describes the RPC plugin and allows the Sybase CEP Server to call it. Every RPC plugin should be described in a separate service entry of the `c8-services.xml` file.

These components work together as follows:

1. The CCL Remote Procedure Statement or remote subquery invokes the RPC plugin by specifying the plugin's unique service name (as identified in the `c8-services.xml` file) and may also pass values to the plugin, which are then passed on to the RPC server.

   The Remote Procedure Statement sends remote procedure calls to an external non-CCL, non-relational database destination, but cannot retrieve any data from this remote service into CCL. The remote subquery is a subquery within a CCL Query Statement that sends Remote Function Calls to an external non-CCL, non-relational database service, and retrieves the resulting data into the CCL query. A remote subquery does not modify the data in the external service. For more information about the Remote Procedure Statement and the remote subquery, see the Sybase CEP *CCL Reference Guide* .
2. Sybase CEP Server uses the service name and the information defined for the service in the `c8-services.xml` file to determine which RPC plugin to use and which function(s) in that plugin to call. Sybase CEP Server may also read some plugin configuration parameters (also defined in the `c8-services.xml` file service entry) and make these available to the plugin, which can read them by calling appropriate function(s) in the API.
3. The RPC plugin communicates with the RPC server and invokes the remotely-callable procedure. When appropriate, the plugin passes remote procedure parameter values from the CCL query to that RPC and/or receives values back from the RPC and makes them available to the CCL query.

The RPC Plugin

A description of the RPC Plugin and its functionality.

An RPC plugin is a compiled `.dll` file (on Microsoft Windows) or `.so` file (on UNIX-like operating systems). This type of library can be called by Sybase CEP Server, and is able to communicate with an RPC server. The plugin receives information (such as the body of an email message, a stock symbol, or an employee ID number) from a CCL statement and passes the information to the remote procedure, as well as receives information from the remote procedure (when appropriate) and makes it available to the CCL statement.

Sybase CEP Engine includes example HTTP and SOAP RPC plugins. You can also create your own plugins. For more information about RPC plugins, see the Sybase CEP *Integration Guide* .

Remote Service Entries in the c8-services.xml File

The `c8-services.xml` file requires a section for every remote service before you can use the service in a CCL query.

Your specific service entry depends on the RPC plugin and remote procedure you are using. All remote service entries, however, require the following information:

- A unique service name.
- A type of `REMOTESERVICE`.
- The name of the plugin library for the RPC plugin.

The names of the initialize, execute and shutdown function calls within the `.dll` or `.so`.

Optionally, the service entry may also include the names and functions of configuration parameters for the functions in the RPC plugin (for example, the HTTP URL of the RPC server). See the Sybase CEP *Integration Guide* for more information about the RPC plugin name, its initialize, execute and shutdown calls and its optional parameters.

The `c8-services.xml` file contains an example of a remote service entry, using the SOAP RPC plugin. The Sybase CEP *Integration Guide* includes a detailed description of the example plugin. You can use the example plugin to send two zip codes to an RPC server and receive the distance between these two zip codes. You can also use the SOAP RPC plugin section of the `c8-services.xml` file to create a service entry for your own RPC plugin.

To create a service entry for your plugin, make a copy of the SOAP RPC plugin section within the `c8-services.xml` file and replace the values as explained in this section.

```
<Service Name="GetZipCodeDistance"1 Type="REMOTESERVICE"2>
<Description>
 Computes the distance between two US zip codes.
</Description>3
<!-- RPC Plugin -->
<Param Name="RpcLibrary">c8_rpc_http_soap_lib4</Param>
<Param Name="RpcInitCallback">c8_rpc_soap_initialize5</Param>
<Param Name="RpcExecuteCallback">c8_rpc_soap_execute6</Param>
<Param Name="RpcShutdownCallback">c8_rpc_soap_shutdown7</Param>
<!-- SOAP/RPC Plugin -->
<Param Name="HttpURI">8http://teachatechie.com/GJTTWebservices/
ZipCode.asmx
</Param>
<Param Name="HttpTimeout">50</Param>
<Param Name="HttpKeepAlive">true</Param>
<Param Name="HttpEnableLogging">true</Param>
<Param Name="SoapURI">http://teachatechie.com/</Param>
<Param Name="SoapMethod">GetDistance</Param>
<Param Name="SoapAction">
 "http://teachatechie.com/GetDistance"
```

```
</Param>
</Service>
```

1. Provide a unique service name for the remote service in the "Service Name" preference on this line. This name is case-sensitive, must begin with a letter and may contain letters, digits, underscores, dots and/or colons. The service name must be the same name as the name of the remote service you use in your CCL queries.
2. Set the "Type" preference on this line to **REMOTESERVICE**.
3. (Optional). Add a description of your service entry in the "Description" element.
4. Set the "RpcLibrary" preference on this line to the name of the library used by your RPC plugin. For the Sybase CEP SOAP and HTTP RPC plugin, use c8_rpc_http_soap_lib.

   When specifying the library name:

   - Do not include the `.dll` or `.so` file name extension. Sybase CEP Server adds the appropriate extension automatically. On UNIX-like operating systems, Sybase CEP Server also adds a `lib` prefix (such as in `libc8_rpc_http_soap_lib.so`) if necessary.
   - Do not specify the library path. Both Microsoft Windows and UNIX-like operating systems automatically search for the library in the default library locations for the operating system.
5. Set the "RPCInitCallback" preference on this line to the name of the initialize function used by the plugin. If you are using the SOAP RPC plugin, the initialize function is **c8_rpc_soap_initialize**. For the HTTP RPC plugin, use c8_rpc_http_initialize.
6. Set the "RPCExecuteCallback" preference on this line to the name of the execute function used by the plugin. If you are using the SOAP RPC plugin, the execute function is **c8_rpc_soap_execute**. For the HTTP RPC plugin, use c8_rpc_http_execute.
7. Set the "RPCShutdownCallback" preference on this line to the name of the shutdown function used by the plugin. If you are using the SOAP RPC plugin, the shutdown function is **c8_rpc_soap_shutdown**. For the HTTP RPC plugin, use c8_rpc_http_shutdown.
8. Optionally, use this and subsequent lines to specify plugin configuration parameters used by the plugin's functions. Here is a list of the configuration parameters used by the SOAP and HTTP RPC plugins. For detailed information regarding the configuration parameters available with the SOAP and HTTP RPC plugins, see the Sybase CEP *Integration Guide* .

HttpURI	The HTTP URI to which the plugin should send the request.
HttpTimeout	The HTTP connection timeout.
HttpKeepAlive	The service should use keep-alive connections.
HttpEnableLogging	The service should log arguments and return values for all plugin calls.
SoapURI	The URI of the SOAP resource.
SoapMethod	The name of the SOAP method to call.

SoapAction	The SOAPAction HTTP header.

You can also configure the remote service entry for a number of additional settings, as described in *"Setting Optional Preferences for Services"* on page 71.

Enabling Adapter-Specific Preferences

A description of the requirements for using an SMTP Email Output Adapter. The steps in this section require you to modify the `c8-services.xml` file.

If you are using an SMTP Email Output Adapter, the following section must be included in the `c8-services.xml` file:

```
<Service Name="
SMTP-service-name
"1 Type="SMTP"2>
 <Description>
service-description3
</Description>
 <Param Name="Server">
server-hostname4
</Param>
 <Param Name="Port">
port-number5
</Param>
 <Param Name="Username">
SMTP-user-name6
</Param>
 <Param Name="Password">
SMTP-password7
</Param>
</Service>
```

1	Set the "Service Name" preference on this line to the same value as the "SMTPService" preference of your adapter. See the Sybase CEP *Integration Guide* for more information about setting up and using the SMTP Email Output Adapter.
2	Set the "Type" property on this line to **`SMTP`**
3	(Optional). Enter a description of this service in the "Description" element.
4	Set the "Server" preference on this line to the host name of the computer on which the SMTP service is running.
5	(Optional). Set the "Port" preference on this line to the port number you want to use to contact the SMTP service. The default value for this preference is 25.

6	If the SMTP service you are using requires user and password authentication, the "Username" preference must specify the user name you are using to connect to the SMTP service.
7	If the SMTP service you are using requires user and password authentication, the "Password" preference must specify the password you are using to connect to the SMTP service.

Setting Optional Preferences

You can configure service entries in the `c8-services.xml` file with a number of optional settings.

Some of these preferences are only applicable to certain types of services, for example, databases, while others may be set for any service. To configure one of these optional preferences for your service, enter the preference, along with the appropriate value, anywhere within your service entry.

Setting Maximum Call Execution Time

The MaxCallExecutionTime preference defines the maximum amount of time that Sybase CEP Engine allows for the database or remote procedure to execute a call.

You can set a "MaxCallExecutionTime" preference for remote services and database services (except database services that establish a connection to a kdb+ database).

- You can set "MaxCallExecutionTime" to any valid CCL INTERVAL literal (see the Sybase CEP *CCL Reference Guide* for more information about valid INTERVAL literals).
- If you set this preference to 0, or set the "" preference to **`TRUE`** , Sybase CEP Engine does not limit the time it allows for the call (see the description of *"Disabling the Call Abort Feature"* on page 71).
- If you do not set the "DisableAbort" preference, and specify an integer value for "MaxCallExecutionTime", the "MaxCallExecutionTime" setting specifies the amount of time allowed for the call. Sybase CEP Engine terminates any call that exceeds its allowed maximum value. A terminated call is *not* retried.

The following example sets the maximum call execution time to five minutes.

```
<Param Name="MaxCallExecutionTime">5 MINUTES</Param>
```

Use "MaxCallExecutionTime" only in conjunction with direct CCL connection to a database or remote server (using the CCL Database Statement, database subquery, Remote Procedure Statement, or remote subquery). This preference cannot be set for adapters.

Disabling the Call Abort Feature

A description of the process for enabling the "DisableAbort" preference for drivers that cannot safely abort in-process requests when a Sybase CEP Engine project stops or times out.

By default, Sybase CEP Server automatically aborts pending remote calls to databases or external services when a Sybase CEP Engine project is stopped, or if the call times out (see the description of *"MaxCallExecutionTime"* on page 71). Some services, such as ODBC drivers, do not fully support this functionality, and may occasionally cause problems when an in-progress call is aborted. If your driver is having problems aborting in-process requests, you can enable the "DisableAbort" preference. This preference disables the effects of "MaxCallExecutionTime" and forces a project to wait for the completion of all initiated calls to remote services and databases before stopping. If you enable this feature, Sybase CEP Engine project shutdown may take longer, as it waits for all outstanding requests to be completed. The default setting for this preference is **`FALSE`** .

The following example enables the "DisableAbort" preference:

```
<Param Name="DisableAbort">true</Param>
```

"DisableAbort" has no effect on kdb+ databases and on public windows.

Setting the Time Zone Conversion Preference

Describes how to set the "Timezone" preference, which you can use to convert time types from another application or program to the correct time zone of your database or RPC server before passing the values to Sybase CEP Engine.

This conversion is necessary when converting from a relative time type, such as a DATE datatype in an Oracle database to Sybase's absolute **TIMESTAMP** datatype.

The preference setting allows any standard time zone abbreviation. The default value for "Timezone" is UTC (Coordinated Universal Time).

The following example sets the time zone for the service to Pacific Standard Time.

```
<Param Name="Timezone">PST</Param>
```

For more information about timezones, see the Sybase CEP *Integration Guide* .

Enabling Tracing

Describes the "EnableTracing" preference and its functionality.

When you set the "EnableTracing" preference to `TRUE`, Sybase CEP Engine logs all database, RPC server, or public window requests for the service to the Sybase CEP Server log as info level messages. Here is an example:

```
<Param Name="EnableTracing">true</Param>
```

Info level logging must be enabled when you enable tracing. This is the default setting for logging. However, if you have reset the logging level in the "SybaseC8/Logging/LogWriters" section of the `c8-server.conf` file, set this preference back to its default before enabling tracing. For more information about Sybase CEP Server logging, see *"Specifying Sybase CEP Server Log Location"* on page 37.

Note that turning on tracing may degrade performance, and tracing is recommended only during debugging.

The default setting for the "EnableTracing" preference is `FALSE`.

Ignoring Database or RPC Server Errors

A description of the "IgnoreErrors" preference and its functionality.

By default, Sybase CEP Engine treats some errors returned by a database or remote procedure call as fatal errors, which halt the execution of the Sybase CEP Engine project. You may override this behavior by setting the "IgnoreErrors" preference to **`TRUE`** . Here is an example:

```
<Param Name="IgnoreErrors">true</Param>
```

This setting causes the Sybase CEP Server to treat *all* errors generated by the database or remote procedure call as warnings. Following the error, Sybase CEP Server drops the data triggering the query, but the project continues running.

Note: Use this setting with utmost caution, since it causes syntax errors returned by the database or remote call to be ignored by Sybase CEP Server. Sybase recommends that you check the Sybase CEP Server log for any errors or warnings generated by the database or RPC server.

The default setting for the "IgnoreErrors" preference is **`FALSE`** .

Setting Automatic Retry

Set the MaxRetries parameter in the service definitions for "Database" or "RemoteService" to determine how many times Sybase CEP Engine should retry statement execution after experiencing errors.

The parameter in the service file is written as follows:

```
<Param Name="MaxRetries">some_number</Param>
```

where some_number is an integer between 0 and 255.

If the MaxRetries parameter is set to a value greater than 0, say N, then every time there is an error in an Execute Statement or Subquery statement that subscribes to this service definition, the Sybase CEP Engine automatically retries the execution of that statement up to a maximum of N number of times. If the execution of the statement fails for all the N number of times, an error tuple is inserted into the errorstream.

The default value for MaxRetries is 0, that is, by default there is no retry.

Caching Data from an External Database, RPC Server, or Public Window

Sybase CEP Engine sometimes experiences delays when evaluating database subqueries and remote subqueries. To eliminate this delay, set caching preferences to enable Sybase CEP Server to cache values received from external databases and remote servers.

Sybase CEP Engine is optimized to evaluate CCL statements very quickly. However, database subqueries (which join a Sybase CEP Server data stream to an external relational database table or to a public window) and remote subqueries (which join the Sybase CEP Engine data stream to a remote procedure call server) can significantly slow query execution if every row in the data stream is sent to an external database or remote procedure server.

To eliminate this delay, Sybase CEP Server caches values received from external databases and remote servers. When database, public window, or remote procedure data is cached, Sybase CEP Server only queries the external database, public window, or RPC server when a new row arrives in the joined data stream and the cache no longer contains a matching queryresult. Otherwise, Sybase CEP Server uses the result stored in the cache, and does not send the row to the external database, remote server, or public window.

The following three preferences control the operation of the cache. Sybase CEP Engine uses these preferences only in conjunction with direct CCL connection to a database, remote server, or public window. Adapters do not use these preferences:

- "CacheMaximumAge".
- "CacheMaximumMemoryUsage".
- "CacheMaximumSize".

You can set these caching preferences either within the c8-services.xml file, or by using the CCL CACHE clause, inside a database subquery, or remote subquery. If you set caching preferences in both places, Sybase CEP Engine uses the settings of your CACHE clause, and ignores the settings in the c8-services.xml file. You can also set the CACHE clause to clear the cache whenever a row arrives in a specified data stream or named window, regardless of the caching policies. For more information, see the "CACHE Clause" section of the Sybase CEP *CCL Reference Guide* .

"CacheMaximumAge" determines whether caching takes place and specifies the time interval during which data from the database subquery, RPC server, or public window is cached. See the Sybase CEP *CCL Reference Guide* for more information about supported interval specifications. The maximum setting for this preference is 100 years. If you set "CacheMaximumAge" to 0, or do not specify this preference, then Sybase CEP Server does not cache data and all other caching preferences are ignored. If you specify a value of -1, caching is infinite with respect to age, in which case you must set another caching preference to limit the cache size in another way.

As a general rule, if the data in the database, public window or RPC server changes infrequently and you have enough memory to cache the data, then you should enable data caching and set to a longer interval. However, if the data changes frequently and you need the most up-to-date data possible, then you should set caching to a short duration or disable it completely.

In the following example, "CacheMaximumAge" is set to 30 seconds. This means that, for any CCL rows received by the query within the 30 seconds, Sybase CEP Engine retrieves the value for the corresponding database, public window, or RPC value from the cache, instead of sending it to and retrieving it from the database, public window, or RPC service itself.

```
<Param Name="CacheMaximumAge">30 SECONDS</Param>
```

Note: Sybase CEP Server does not automatically detect when an external database table or public window is updated or when data changes on the RPC server. The database or RPC server does not inform Sybase CEP Engine about updates and Sybase CEP Server does not "poll" the external server for changes. Sybase CEP Server only queries the external server when a new row arrives in a Sybase CEP Engine stream and the data in the cache has expired.

"CacheMaximumMemoryUsage" controls the cache contents based on the size of the cache. This preference specifies a maximum number of bytes for the cache. The default value is 0 (unlimited). This preference can be set to an integer of any size to specify the maximum number of bytes for the cache. The value is specified as a 32-bit integer on 32-bit systems, or as a 64-bit integer on 64-bit systems. Sybase CEP Engine enforces the maximum byte size separately for every database subquery or remote subquery.

When the cache exceeds the size you specified, Sybase CEP Engine begins deleting items from the cache from least to most-recently used, until it removes enough items to bring cache size below the maximum. The following example sets the maximum cache size for each database subquery, remote subquery, or public window to one Megabyte (1Mb).

```
<Param Name="CacheMaximumMemoryUsage">1048576</Param>
```

Note: Since "CacheMaximumMemoryUsage" limits the cache to an actual number of bytes, consider issues such as data fragmentation when setting this preference.

"CacheMaximumSize" limits the size of the cache to the specified number of rows. The default value is 0 (unlimited). You can also set this preference to an integer of any size (taking into account your system's limitations) to specify the maximum number of rows the cache can hold. Sybase CEP Engine enforces the maximum row number separately for every database subquery, remote subquery, or public window.

When the cache exceeds the number of rows you specify, Sybase CEP Engine begins deleting rows, from least to most-recently used, until it removes enough items to bring cache size below the maximum. The following example sets the maximum number of rows in the cache to 100 for every database subquery, remote subquery, or public window.

```
<Param Name="CacheMaximumSize">100</Param>
```

Cache policies set by "CacheMaximumAge", "CacheMaximumMemoryUsage", and "CacheMaximumSize" interact in the following way:

"CacheMaximumAge" Setting	"CacheMaximumMemoryUsage" Setting	"CacheMaximumSize" Setting	Comments

-1	0 or omitted (defaults to 0)	0 or omitted (defaults to 0)	Not permitted. If you set "CacheMaximumAge" to infinite size, you must limit the cache by byte size and/or number of rows. Sybase does not permit unbounded infinite caching.
-1	positive integer	0 or omitted	Cache is limited by maximum number of bytes only.
-1	0 or omitted	positive integer	Cache is limited by maximum number of rows only.
-1	positive integer	positive integer	Cache is limited by maximum number of bytes and maximum number of rows.
0 or omitted (defaults to 0)	any setting	any setting	Caching is turned off. "CacheMaximumMemoryUsage" and "CacheMaximumSize" settings (if any) are ignored.
interval specification	0 or omitted	0 or omitted	Cache is limited by row age only.
interval specification	0 or omitted	positive integer	Cache is limited by row age and number of rows in the cache.
interval specification	positive integer	0 or omitted	Cache is limited by row age and byte size.
interval specification	positive integer	positive integer	Cache is limited by row age, byte size and number of rows.

Caching preferences affect only the service for which you specify them.

Setting Concurrency Preferences

Set concurrency properties for database, public window, and remote service entries.

By default, Sybase CEP Engine establishes a separate connection with a database, public window, or RPC server for every CCL query that requires such a connection. Concurrency preferences enable you to override this default behavior by allowing multiple CCL queries using the same service to share a single connection, or by using several connections to process a single query.

Concurrency settings apply only to direct CCL connections to a database, public window, or RPC server. Adapters do not use these preferences. The two preferences used to specify concurrency preferences are:

- "MaxPoolSize".
- "MaxParallelInvocationsPerCall".

The "MaxPoolSize" preference establishes a pool within which multiple CCL queries may share a single connection to the RPC server, public window, or database server specified by the service entry. The "MaxPoolSize" indicates the maximum number of permitted connections that all queries running at the same time on a single Sybase CEP Server instance share. The maximum pool size is 1024. A "MaxPoolSize" of 0 indicates an unlimited pool. This is the default maximum size of the pool setting.

The following example establishes a maximum pool size per Sybase CEP Server instance per service entry of ten.

```
<Param Name="MaxPoolSize">10</Param>
```

The "MaxParallelInvocationsPerCall" preference allows you to distribute the processing of a single query over multiple connections to the database server, public window or RPC server specified by the service entry. "MaxParallelInvocationsPerCall" establishes the maximum number of connections per service entry that a Sybase CEP Server instance can use to process a single CCL query. You can specify a preference with a range of integer values from 1 to 1024. The default value is 1.

The following example allows up to three concurrent connections for a single database subquery or remote subquery.

```
<Param Name="MaxParallelInvocationsPerCall">3</Param>
```

Sybase CEP Engine uses concurrent query execution settings only for CCL database subqueries and remote subqueries, which retrieve data from the remote server, database, or public window. CCL Database Statements and Remote Procedure Statements, which write information to the database, or remote service, do not use these settings.

Note: If you set "MaxPoolSize" to 0 (unlimited) the maximum size of the pool is calculated by multiplying the "MaxParallelInvocationsPerCall" by the number of queries connecting to the service from a given Sybase CEP Server instance. Sybase therefore recommends that you do not set "MaxPoolSize" to 0 when you set "MaxParallelInvocationsPerCall" to a value of greater than 1 to avoid running out of resources.

The following table shows connection behavior with various combinations of "MaxPoolSize" and "MaxParallelInvocationsPerCall" settings. In all cases, the table specifies established and dropped connections per Sybase CEP Server instance per service (that is, per CCL statements using the service and running on a single Sybase CEP Server instance).

"MaxPoolSize" Setting	**"MaxParallelInvocationsPerCall" Setting**	**Connections Established**
0 (default setting -- unlimited pool)	1 (default setting)	A single new connection for every CCL statement that uses the service.
0	**X** (where **X** is an integer between 2 and 1024)	**X** new connections for every CCL database subquery or remote subquery that uses the service. A single new connection for every CCL Database Statement or Remote Procedure Statement that uses the service.
Y (where **Y** is an integer between 1 and 1024)	1	A single new connection for every CCL statement that uses the service, up to the specified **Y** maximum. Any of the connections in the pool are available to any statement, but a single statement may use only one connection at any given time.
Y (where **Y** is an integer between 1 and 1024)	**X** (where **X** is an integer between 2 and 1024)	**X** new connections for every CCL database subquery or remote subquery that uses the service, up to the specified **Y** maximum. A single new connection for every CCL Database Statement or Remote Procedure Statement that uses the service, up to the specified **Y** maximum. Any of the connections in the pool are available to any statement, but a given database subquery or remote subquery may use only **X** connections at any given time, and a given Database Statement or Remote Procedure Statement can only use one connection at a time.

Database-Specific Service Preferences

A description of the database and public window entry preferences and their functionality.

The following additional preferences apply only to database and public window service entries. Sybase CEP Engine uses these preferences only for direct CCL connections to the database via the Database Statement or database subqueries. You cannot set these preferences for use with adapters.

- "DBWriteMaxBatchSize"

  This preference determines how many database writes Sybase CEP Engine may send to the database from a CCL Database Statement in a single batch. You can specify a value between 1 and 65536 for this preference. The default value is 1. "DBWriteMaxBatchSize" affects only requests that are already in the processing queue - Sybase CEP Server does not wait for the maximum number of requests to accumulate before executing them. Sybase CEP Engine does not support batching for **BLOB** and **XML** data types. Since public windows do not accept database writes, this preference does not affect public windows. "DBWriteMaxBatchSize" also has no effect on Oracle TimesTen connections.

  Note: Some ODBC databases (such as MySQL's ODBC driver) do not support batch sizes of greater than 1 for stored procedure calls, such as the **CALL** statement or **BEGIN ...END** blocks. These databases may not issue an error for the unsupported batch size, but may simply ignore subsequent data after the first call in the batch. The MySQL ODBC driver also does not support batching for **INSERT**, **DELETE** and **UPDATE** statements.

  If you are using a "DBWriteMaxBatchSize" of more than one 1, in conjunction with an ODBC database connection and are using stored procedure calls, verify that your database is receiving all the expected data from your batch before using this setting in a production environment. Do not set a "DBWriteMaxBatchSize" of more than one 1 when using a MySQL ODBC driver.

- "DBReadNoCommit"

  By default, Sybase CEP automatically issues an SQL **COMMIT** after every executed database statement, regardless of whether the statement is a read or a write. If this is affecting performance on your database reads, the "DBReadNoCommit" preference can be set to **`TRUE`** (from its default setting of **`FALSE`**) to disable the commits for database subquery execution only. Note that enabling this option prevents you from issuing **SELECT FOR UPDATE** and other SQL statements that require a **COMMIT**. Since public windows do not accept database writes, this preference does not affect on public windows.

- "DBDriverCharBufferSize"

  The "DBDriverCharBufferSize" preference specifies the size of the Sybase CEP Server buffer for sending *any* data to and receiving *any* data from an external database. This preference can be set to an integer between 1 and 65536. The default value is 4096. Sybase CEP Server generates errors in response to any attempt to send or receive data larger than the specified buffer size maximum (or larger than the default value, if no maximum is specified). Sybase CEP Engine does not support buffer sizes that exceed the 65536 maximum, even when external databases support such buffer sizes.

This preference does not affect public windows or connections to kdb+ databases, both of which supports an unlimited buffer size.

- "DBDriverNoBigintSupport"

  Some ODBC drivers (especially older pre ODBC 3.51 versions) do not have native support for 64-bit integers. Set the"DBDriverNoBigintSupport" preference to **`TRUE`** or **`FALSE`** to notify Sybase CEP Server whether or not the underlying ODBC driver or database supports native 64-bit integer transfers. A settings of **`TRUE`** indicates that the ODBC driver or database does not support native 64-bit integer transfers, and prompts Sybase CEP Server to transparently convert 64-bit integers into strings. Note that you may slow down data transfer by enabling this preference. The default setting for "DBDriverNoBigintSupport" is **`FALSE`** .

If you are not sure whether your ODBC driver supports 64-bit integers, set "DBDriverNoBigintSupport" to **`FALSE`** . If you then detect messages related to LONG, TIMESTAMP or INTERVAL data types, try changing this preference to **`TRUE`** and then restarting Sybase CEP Server.

You must set this preference to TRUE when using the Oracle TimesTen database.

Troubleshooting Configuration Settings

If the Sybase CEP Engine installation is not running properly, first check to see that Sybase CEP Server is running, and then check the log file.

Making Sure That Sybase CEP Server Is Running

Sybase CEP Server must be running in order for Sybase CEP Engine to work. To check if Sybase CEP Server is working, start Sybase CEP Studio, which automatically tries to connect to Sybase CEP Server.

If Sybase CEP Server is running properly and the current workspace is connected, the workspace icon (displayed in the left upper pane of Sybase CEP Studio) looks like this: 🖥 .

If Sybase CEP Server is not running, or if Sybase CEP Studio is not able to connect to it for some reason (such as due to a networking problem) then your current workspace appears in italics, with the words "not connected" next to it, and the icon looks like this: ❗ .

Checking the Sybase CEP Server Log

A Sybase CEP Server log file is generated every time the Sybase CEP Server is stopped and then restarted. If Sybase CEP Server or Sybase CEP Studio does not start properly, check the Sybase CEP Server log.

The latest log file is called `server.log`; older logs are saved as `server1.log`, `server2.log`, and so on.

The default location of the log files on Microsoft Windows is `C:\sybase\SybaseC8\Server\logs`

The default location of the log files on UNIX-like operating systems is located under the following subdirectory of the `c8` installation directory: `SybaseC8/server/logs.`

Using the Watchdog Utility

Sybase provides a watchdog monitoring utility with Sybase CEP Engine. To use this utility, or to learn more about it, see the `Server\watchdog` directory of your Sybase CEP Engine installation.

Enterprise Sybase CEP Engine Settings

A description of the settings for the Sybase CEP Engine.

High Availability

Describes settings for the Sybase CEP Engine High Availability feature, which supports enterprise installations of Sybase CEP Server in cases where a Sybase CEP Server stops operating due to a hardware failure or a software crash.

The Sybase CEP Engine High Availability (HA) feature supports enterprise installations of Sybase CEP Server in cases where a Sybase CEP Server stops operating due to a hardware failure or a software crash. In such cases, if you have enabled High Availability, another Sybase CEP Server instance replaces the malfunctioning Sybase CEP Server instance, without shutting down the application. This process is called *failover*.

Every Sybase CEP Server instance includes a single Container process, or a single Manager process, or both a Container and a Manager. Containers run queries and C/C++ input and output adapters, while the Manager monitors the Containers and manages their workload. Enterprise installations of Sybase CEP Server may include clusters of Sybase CEP Servers, consisting of any combination of multiple active and passive Containers, one active Manager and backup Managers.

Note: This is the precise definition of Sybase CEP Server and Sybase CEP Server cluster, and is relevant in discussions of High Availability, and in some other discussions of Sybase CEP Server configuration, and of distributed Sybase CEP Server solutions. However, in many other cases, Sybase documentation uses the term Sybase CEP Server more generically, to describe Sybase CEP Server functions performed either by a stand-alone Sybase CEP Server installation, or by a Sybase CEP Server cluster.

High Availability features can be set for both Containers and for Managers. For more information about High Availability as it pertains to adapter usage, see the Sybase CEP *Integration Guide* .

Understanding High Availability Features for Containers and Managers

The following sections explain how High Availability features work for Sybase CEP Server Container and Manager processes.

High Availability Configuration for Containers

A description of the High Availability configuration for Containers within Sybase CEP Server clusters, as well as their general functionality.

A Sybase CEP Server cluster includes several Containers and one active Manager. Sybase CEP Server may set its Containers to active or passive status. Active Containers are actively used by the Manager when it distributes the workload, while passive Containers remain in standby mode, with no workload until an active Container fails.

All Containers notify the Manager of their availability when they first start up. Containers also send a continuous periodic "heartbeat" to the Manager while they are running. The frequency of the heartbeat is determined by the number of seconds you specify in the Container's "HeartbeatFrequencySeconds" preference in the Container's `c8-server.conf` file.

The Manager detects Container failure when it does not receive three consecutive heartbeats from the Container. The Manager considers the first heartbeat as missing if the gap between heartbeats exceeds the interval set by "HeartbeatFrequencySeconds" multiplied by three. It considers the second heartbeat missing, if no heartbeat arrives after an additional "HeartbeatFrequencySeconds" interval. It considers the third heartbeat as missing if no heartbeat arrives within yet another "HeartbeatFrequencySeconds" interval. For example, if "HeartbeatFrequencySeconds" is set to two (2), the first heartbeat is assumed to be missing after six seconds; the second and third heartbeats are assumed to be missing after an additional two seconds each. A total of ten seconds elapse before the Manager initiates failover procedures.

After waiting the appropriate number of seconds, the Manager initiates failover procedures in the following order:

1. If an active Container fails momentarily and restarts immediately, the Manager reassigns the Container's workload back to the Container.

   This occurs if a Container fails and restarts before the Manager detects that the Container failed. The Manager does not attempt to restart a Container once it determines that the container has failed.
2. If an active Container fails and does not recover immediately, and one or more passive containers are available, the Manager activates a passive Container and reassigns the failed Container's work to the new Container.
3. If an active Container fails and does not recover immediately, and no passive Containers are available, the Manager redistributes the failed Container's workload among the remaining Containers.

When configuring Container High Availability features, you can specify the number of active Containers that the Manager should try to maintain at any given time and the maximum number of CCX modules that each Container is allowed to run. However, you cannot explicitly indicate to the Manager which Containers it should activate at a particular time.

High Availability Configuration for Managers

A description of the High Availability configuration for Managers within Sybase CEP Server clusters, as well as their general functionality.

A Sybase CEP Server cluster configured for Manager HA includes one active or primary Manager and one or more passive Managers. If the active Manager fails, one of the passive Managers becomes active. Managers within a Sybase CEP Server cluster communicate with one another by writing information to and reading information from a common directory, specified in the "StoragePath" preference. Two of the HA-specific files that reside in this directory are `mgr_primary_node` and `mgr_heartbeat`. The `mgr_primary_node` file indicates which Manager is currently functioning as the primary. The `mgr_heartbeat` file records the primary Manager's heartbeat (which you set with the "HeartbeatFrequencySeconds" preference in the `c8-manager-cluster.xml` file.)

Sybase CEP Server uses this heartbeat to determine whether or not the primary Manager is available and running properly. If the primary Manager fails, the other Managers in the Cluster detect that the Manager's heartbeat has stopped and initiate the selection of a new primary Manager.

Since the new primary Manager has a different IP address from the original Manager, the new primary Manager must "announce" itself to all the Containers in the Server Cluster, to make sure that communications intended for the Manager are routed to the right destination. To avoid the need to notify each individual Container, whenever a new Manager is activated, the Sybase system relies on a dynamic Domain Name Service (DNS) to route messages from the Containers to the current active Manager.

Containers send messages to a generic Manager hostname. The DNS server then translates the hostname to the IP address of the current active Manager. In the event of Manager failover, the new primary Manager notifies the DNS server that the generic host name should now point to the new Manager's IP address. In order for Manager HA features to work correctly, the DNS server must be running continuously and must be accessible to all Managers and Containers.

Note, however, that the DNS does not translate port numbers. Since Containers in a Sybase CEP Server cluster are configured for a specific Manager port, all Managers in the Cluster must use the same port.

How to Configure Sybase CEP Server for High Availability

A description of how to set up Container and Manager high availability features. The steps in this section require you to modify the `c8-server.conf` configuration file.

The following sections provide you with instructions for setting up Container and Manager High Availability features. Directory and file names appear in Microsoft Windows format. If you are running Sybase CEP Engine on a UNIX-like operating system, change file name and directory format accordingly.

Many of the steps in this section involve changes to the `c8-server.conf` configuration files of the Sybase CEP Servers included in your cluster. See "*Introduction to Configuring Sybase CEP Engine*" for the default location of this file within each Sybase CEP Server's installation directory.

Note: This chapter describes all the preferences that you must set to use Sybase CEP Engine's High Availability features and assumes that you are setting these preferences manually in the configuration files. If you are installing Sybase CEP Server on a UNIX-like operating system, however, you can use the `install-server.sh` Sybase CEP Server script to configure many High Availability settings automatically during installation.

Nevertheless, even if you plan on running Sybase CEP Server on a UNIX-like system and do not need to set all the High Availability preferences manually, Sybase recommends that you read the rest of this chapter before installing Sybase CEP Server to understand the various preference settings. After you familiarize yourself with the preferences, run `install-server.sh` once for every Manager or Container Server you are installing and make the appropriate selections when prompted by the script. After running the script, edit the configuration files to set the remaining preferences, as discussed in this chapter.

Preparing to Configure High Availability Features

A description of the steps to complete before you configure High Availability features for Sybase CEP Engine.

Before you configure High Availability features for Sybase CEP Engine, complete the following steps:

1. In the `c8-server.conf` configuration file of every Sybase CEP Server in your cluster, designate whether you want Sybase CEP Server to act as a Manager, Container, or both Manager and Container. To do this, set the following preferences:
   a. For every Sybase CEP Server in your cluster that you want to designate as a Container, set the "SybaseC8Server/Container/Enabled" preference to **`true`** :

```
<section name="SybaseC8/Server">
        ...
        <section name="Container">
  ...
  <preference name="Enabled" value="true"/>
```

   b. For every Sybase CEP Server that is enabled as a Container, that you do *not* also want to enable as a Manager, set the "SybaseC8/Server/Manager/Enabled" preference to **`false`** :

```
<section name="SybaseC8/Server">
        ...
        <section name="Manager">
  ...
  <preference name="Enabled" value="false"/>
```

   c. For every Sybase CEP Server in your cluster that you want to enable as a Manager, set the "SybaseC8/Server/Manager/Enabled" preference to **`true`** :

```
<section name="SybaseC8/Server">
        ...
        <section name="Manager">
  ...
  <preference name="Enabled" value="true"/>
```

d. For every Sybase CEP Server that is enabled as a Manager that you do *not* also want to enable as a Container, set the "SybaseC8/Server/Container/Enabled" preference to **`false`** :

```
<section name="SybaseC8/Server">
        ...
        <section name="Container">
  ...
  <preference name="Enabled" value="false"/>
```

2. Write down the host name and IP address of every Sybase CEP Server in your cluster, and note whether each Sybase CEP Server is enabled as a Manager, Container, or both. For example:

```
alpha 10.10.10.97 Manager
    bravo 10.10.10.98 Container
    ...
```

For machines that include multiple network cards and have multiple IP addresses, note both addresses.

3. Check the following preference settings in the `c8-server.conf` configuration file of every Sybase CEP Server in your cluster. You will find some of these settings already set appropriately by the installation process, but will need to change others manually.

a. Set the "SybaseC8/Server/Common/Hostname" preference to the host name you wrote down in Step 2. For example:

```
<section name="SybaseC8/Server">
        <section name="Common">
   <!-- the server's hostname -->
   <preference name="Hostname">alpha</preference>
```

b. Set the "SybaseC8/Server/Common/ListenTo" preference to the IP address or host name on which you want the Sybase CEP Server to "listen". The default setting for this preference is 0.0.0.0, which indicates that Sybase CEP Server should listen to all available IP addresses:

```
<section name="SybaseC8/Server">
        <section name="Common">
   ...
   <!-- the server's hostname or IP that it would listen to -->
   <preference name="ListenTo">0.0.0.0</preference>
```

Some computers have multiple network cards, and, consequently, multiple IP addresses. If you want to limit the addresses to which Sybase CEP Server listens, list the addresses, or their associated host names here, instead of 0.0.0.0. If you are listing multiple addresses or host names, separate the entries by commas, for example:

```
<section name="SybaseC8/Server">                      <section
name="Common">
   ...
   <!-- the server's hostname or IP that it would listen to -->
   <preference name="ListenTo">
      12.13.14.15, 12.13.200.1</preference>
```

c. If more than one Sybase CEP Server uses the same host name or IP address, set each Sybase CEP Server to use a different port number, as shown in the next step. However, two Managers that use the same host name or IP address cannot both be part of a High Availability cluster, since all Managers in such a cluster must have the same port number.

d. Set the "SybaseC8/Server/Common/Port" preference to the port number you want each Sybase CEP Server to use, for example:

```
<section name="SybaseC8/Server">
        <section name="Common">
   ...
   <!-- the server's port number that it would listen to -->
   <preference name="Port">6789</preference>
```

If you plan to configure the Manager failover High Availability, set the port for all Managers in your intended cluster to the same port number.

1. If any of the Sybase CEP Servers in your cluster are running on the same machine, set the "SybaseC8/Logging/LogWriters/FileLogger/Filename" preference to a unique file name for that machine, to ensure that each Sybase CEP Server has its own log file.

   For example:

```
<section name="SybaseC8/Logging/LogWriters">
   ...
   <section name="FileLogger">
     ...
     <preference name="Filename">
     C:\ProgramFiles\SybaseC8\Server\logs\Container1.log
     </preference>
```

2. If you plan on configuring the Manager failover High Availability features (in which multiple Sybase CEP Servers are configured as Managers, though only one of these is active at a time) create a `conf` and `storage` directory, as explained here.

   (If you plan on using only one Manager in your Sybase CEP Server cluster, and do not plan on enabling Manager failover, you can skip this step.)

   a) Create a directory which all Managers are able to read and to which all of them will be able to write.

If the Managers are running on different machines, all the machines must be able to connect to the shared directory.

b) Create two shared subdirectories named `conf` and `storage` under the shared directory you created in the previous step.

For example, if the shared directory is `C:\public\shared\home\C8MgrShared`, the two subdirectories should be: `C:\public\shared\home\C8MgrShared\conf` and `C:\public\shared\home\C8MgrShared\storage`.

c) Write down the names of these directories.

The Managers use these directories to communicate with one another.

3. Create a shared storage directory accessible to all the Containers in your cluster.

This directory may be the same as the `storage` directory you created in the previous step, or you may create separate storage directories for your Containers. This directory will be used by Containers to store data and prevent data loss during Container failover.

4. If you plan on configuring the Manager failover High Availability features, write down the following information about your DNS system.

(If you plan on using only one Manager with your Sybase CEP Server cluster, and do not plan on enabling Manager failover, you can skip this step.)

a) Write down the name of your DNS Server.

For example:

```
DNSServ1
```

b) Write down the IP address of your DNS.

For example:

```
10.10.10.254
```

c) Write down the DNS zone for your Managers.

For example:

```
myzone.myco.com
```

d) Create and write down the generic host name alias that will be used to access the primary Manager, as explained in *"Configuring Managers for Manager or Container High Availability"* on page 90.

For example:

```
C8MgrShared
```

5. Make sure that the clocks of all computers included in your Sybase CEP Server cluster remain synchronized - for example, by using Network Time Protocol (NTP).

Configuring Managers for Manager or Container High Availability

Describes the steps you must complete for all individual Sybase CEP Servers acting as Managers in your cluster when configuring Manager or Container High Availability.

Complete these steps individually for all Sybase CEP Servers acting as Managers in your cluster, whether or not you plan to run multiple Managers. Some of these values have already been set appropriately by the installation program, but other preferences will need to be set manually. Refer to the information from *"Preparing to Configure High Availability Features"* on page 86 while you work on this section.

1. If you plan to configure Sybase CEP Engine High Availability features for Manager failover, set the "SybaseC8/Server/Common/ManagerURI" preference in the `c8-server.conf` configuration file to the Manager's Uniform Resource Identifier (URI). To use only one Manager with no Manager failover, set this preference to the local address of the Manager.

The Manager URI uses the following format: **`http://`** *generic-host-name-alias:port* or **`https://`** *generic-host-name-alias:port* where *`generic-host-name-alias`* is the generic host name alias you set up in Step 7d of *"Preparing to Configure High Availability Features"* on page 86, and port is the port number set for the Managers in your cluster in Step 3c of *"Preparing to Configure High Availability Features"* on page 86. If SSL is enabled on Sybase CEP Servers in the cluster, then all Manager URIs must start with **`https://`** instead of **`http://`** . For more information about configuring SSL features, see *"Configuring SSL Support"* on page 39.

If you are using multiple Managers in your cluster, and the "ManagerURI" preference is currently set to the local address of the computer where the Manager is installed, replace the local address with the URI. For example, you might change this preference:

```
<section name="SybaseC8/Server">
    <section name="Common">
   ...
   <!-- the URI of the Manager this server is using -->
   <preference name="ManagerUri">
     http://localhost:6789</preference>
```

to this:

```
<section name="SybaseC8/Server">
    <section name="Common">
   ...
   <!-- the URI of the Manager this server is using -->
   <preference name="ManagerUri">
     http://C8MgrShared:6790</preference>
```

As explained in *"Configuring Managers for Manager or Container High Availability"* on page 90, the Domain Name Service (DNS) uses this URI to route messages from

Containers to the current active Manager. All the Managers in your cluster must refer to the same Manager URI.

2. Specify how many active Containers you want the Manager to try to maintain at any given time in the "Manager/HighAvailability/MinActiveContainerNative" preference of the `c8-server.conf` configuration file. For example:

```
<section name="Manager">
    ...
    <section name="HighAvailability">
   <!-- After this number of active native containers is
        reached, all containers will be made passive -->
   <preference name="MinActiveContainersNative" value="0"/>
```

   The default setting for this value is 0, which specifies unlimited active Containers. If you set the value to 0, the Manager activates all available Containers. If you set the value to another number, the Manager attempts to limit the number of Containers to the number you specify in "MinActiveContainersNative".

   Under normal circumstances, the Manger sets all other available Containers to passive mode until they are needed for failover. However, the "LoadLimit" setting, which specifies the maximum number of CCX modules (that is, adapters or query processors) that a given Container can simultaneously execute, may affect the number of active Containers. If a Container's "LoadLimit" setting does not permit it to take on additional workload, the Manager attempts to activate another available passive Container, even if the number of active Containers specified in the "MinActiveContainersNative" has already been reached.

Complete these steps for every Sybase CEP Server acting as Manager in your cluster only if you are using multiple Managers and setting up Manager failover features. If you plan to use a single Manager with no failover features, skip the rest of this section and go to *"Configuring Containers for High Availability"* on page 95 for instructions on configuring Containers for High Availability features.

1. Set the "SybaseC8/Server/Common/StoragePath" preference in the `c8-server.conf` configuration file to the full path and name of the storage directory created in Step 5b of *"Preparing to Configure High Availability Features"* on page 86.

   For example:

```
<section name="SybaseC8/Server">
<section name ="Common">
   ...
   <!-- the path on disk, under which the server
        components (directory, workspace mgr,
        qp state, etc) should persist -->
   <preference name="StoragePath"
     value="C:\public\shared\home\C8MgrShared"/>
```

This preference must be set to the same directory for all Managers in a cluster to enable the Managers to communicate with one another.

2. If the "Enabled" preference in the "SybaseC8/Server/Manager/HighAvailability/ManagerCluster" section of the `c8-server.conf` file is enclosed in comment marks, remove the comment marks.
3. Set the "Enabled" preference to **true**:

   ```
   <section name="Manager">
      ...
      <section name="HighAvailability">
        ...
        <section name="ManagerCluster">
          <!-- The flag that enables/disables high
            availability configuration for manager
            server (multiple manager nodes) -->
          <preference name="Enabled" value="true"/>
   ```
4. If the "NodeURI" preference, in the "SybaseC8/Server/Manager/HighAvailability/ManagerCluster" section of the `c8-server.conf` file is enclosed in comments marks, remove the marks.
5. Set the "NodeURI" preference to an IP address that uniquely identifies the particular Sybase CEP Server in your cluster.

   This address can be used to identify the current cluster node, regardless of whether it is currently serving as the primary Manager. For example:

   ```
   <section name="Manager">
      ...
      <section name="HighAvailability">
        ...
        <section name="ManagerCluster">
          ...
          <preference
             name="NodeUri"value="206.45.31.158"/>
   ```

   While all Managers in a cluster must have the same "ManagerURI" setting, every Manager within the cluster must have its own unique "NodeURI", which must be specified in both the `c8-server.conf` file and in the `c8-manager-cluster.xml` file.
6. Set up the cluster configuration file. All Managers in a cluster must refer to this file, which, together with the storage path setting, enables communication among Managers. The `conf` subdirectory of the `server` directory in every Sybase CEP Server installation directory contains a sample cluster configuration file, called `c8-manager-cluster.xml`. However, all the Managers in your cluster must be able to access the *same* cluster configuration file in a common location.

   a) Place a copy of the `c8-manager-cluster.xml` file in the shared `conf` directory created in Step 5b of *"Preparing to Configure High Availability Features"* on page 86.

b) Change the path and file name setting of the "ConfigurationPath" preference of every Manager in the cluster to the set the configuration path to the full path and file name of the `c8-manager-cluster.xml` file.

   For example:

```
<section name="Manager">
   ...
   <section name="HighAvailability">
     ...
     <section name="ManagerCluster">
       ...
       <preference name="ConfigurationPath" value=
         "C:\public\shared\home\C8MgrSharedc8-manager-
cluster.xml"/>
```

   This preference is located in the "SybaseC8/Server/Manager/HighAvailability/ManagerCluster" section of the Manager's `c8-server.conf` file. If the preference is enclosed in comment marks, remove the marks when setting the storage path.

c) Configure the cluster configuration file to include information about all the available Managers in your cluster.

   See the comments in the `c8-manager-cluster.xml` file itself for instructions.

7. In the `c8-server.conf` file, specify information for the plugin that notifies the DNS when a new Manager becomes the active Manager.

A plugin is an SO (Shared Object) file on a UNIX-like operating systems, or a DLL (Dynamic Link Library) on Microsoft Windows.

If an active Manager fails, the remaining (passive) Managers hold an election and elect a new Manager as the primary. The plugin you specify here is then called by the new primary Manager. This plugin executes a program, such as `nsupdate`, and passes the necessary parameters to the program. The program in turn notifies the DNS of the new Manager's IP address and related information. Each Manager in your Sybase CEP Server cluster requires its own command file because the Managers all have unique IP addresses.

a) Remove the comments around the "ManagerFailoverDDNSPPlugin" section of the `c8-server.conf` file.

b) Set the "MessageGroup" preference in the "ManagerFailoverDDNSPlugin" section to **"ManagerInfo"**:

```
<!-- Sample command-line plugin configuration -->
        <!--
        <section name="ManagerFailoverDDNSPlugin">
   ...
    <preference name="MessageGroup" value="ManagerInfo"/>
```

c) Set the "MessageName" preference in the "ManagerFailoverDDNSPlugin" section to **"ManagerHAPromotedToPrimary"**.

This event occurs on every Manager in the cluster when a backup Manager node is promoted to be the new primary (active) Manager:

```
<!-- Sample command-line plugin configuration -->
        <!--
        <section name="ManagerFailoverDDNSPlugin">
   ...
    <preference name="MessageName"
   value="ManagerHAPromotedToPrimary"/>
```

d) Set the "CommandName" preference to the program you want the plugin to execute.

   For example:

```
<!-- Sample command-line plugin configuration -->
        <!--
        <section name="ManagerFailoverDDNSPlugin">
    ...
    <preference name="CommandName" value="/usr/bin/nsupdate"/>
```

e) Set the "CommandArgument#" preference values, where # is a number from 1 to 1024. These settings specify the parameters that the plugin should pass to the program specified in "CommandName". The first argument must always be specified under "CommandArgument1". List additional arguments in order, under "CommandArgument2", "CommandArgument3" and so on.

   The parameters specified under command arguments depend on the program being used. In many cases one of these parameters is the name of a text file that includes appropriate commands for updating the hostname-to-IP address mapping on the DNS. By default, the "ManagerFailoverDDNSPlugin" section of the `c8-server.conf` file includes an example text file path under "CommandArgument4", although you are not required to use this preference for hostname-to-IP address mapping.

   Note also that the text file is not generic to all Managers. Since the exact instructions are different for every Manager (because every Manager uses a different IP address), every Manager must reference its own command file.Here is an example of command argument settings:

```
<!-- Sample command-line plugin configuration -->
<!--
<section name="ManagerFailoverDDNSPlugin">
   ...
   <preference name="CommandArgument3" value="-v"/>
   <preference name="CommandArgument4" value="C:\public
\shared\home\C8MgrShared\conf\nsupdate_datafile1.txt"/>
```

   In this example, the plugin passes two parameters to the program. The first of these is the -v preference, which instructs `nsupdate` to read a file. The second preference is the name of a text file that contains commands to change the DNS map. For example, if alpha's IP address is 10.10.10.97, its full name is alpha.myzone.myco.com, the TTL is ten (10) seconds, and the DNS server name is DNSServ1, and the `nsupdate` commands in the `nsupdate_datafile.txt` for alpha may look as follows:

```
server DNSServ1
zone myzone.myco.com
update delete alpha.myzone.myco.com A
update add alpha.myzone.myco.com 10 A 10.10.10.97
show
send
```

Note that DNS caches names and reuses the names stored in the resolver cache. Make sure that the TTL ("Time to Live") setting of your DNS record is set to a short time period (10 or 20 seconds) to prevent old Manager names from being cached and reused. Refer to your DNS documentation for more information about setting the TTL

Some DDNS commands require an authentication key, which may be stored in a file. If your system is configured to require an authentication key, provide the full path and name of the key file in one of the "CommandArgument#" preferences in the `c8-server.conf` file. By default, the "ManagerFailoverDDNSPlugin" section of the `c8-server.conf` file includes a sample text file path entry will be entered as a setting for the "CommandArgument2" preference.

Configuring Containers for High Availability

A description of the steps you must complete for all individual Sybase CEP Servers acting as Containers in your cluster when configuring Container High Availability.

Some values have already been set appropriately by the installation program, but other preferences will need to be set manually. Refer to the information from *"Preparing to Configure High Availability Features"* on page 86 while you work on this section.

1. Set the "SybaseC8/Server/Common/ManagerURI" preference in the `c8-server.conf` configuration file of every Container, to the same setting specified for Manager(s) in Step 1 of *"Configuring Managers for Manager or Container High Availability"* on page 90.

   If the cluster includes multiple Managers, this preference is the DNS URI alias. Otherwise, if the cluster includes only one Manager, with no Manager failover features, this setting is the local address of the Manager.

2. In the "Container" section of each Container's `c8-server.conf` configuration file, set the "HeartbeatFrequencySeconds" preference to the rate at which you want the Container to send heartbeats to the Manager, for example:

   ```
   <section name="Container">
      ...
       <!-- the frequency with which container sends
         SOAP heartbeat messages to the manager -->
      <preference name="HeartbeatFrequencySeconds">5</preference>
   ```

The default setting for this preference is five (5). A setting of zero (0) disables heartbeats. Make sure that this setting is not disabled when configuring High Availability.

3. Set the "SybaseC8/Server/Common/StoragePath" preference in the `c8-server.conf` configuration file for all the Containers in your cluster to the directory designated in Step 6 of *"Configuring Managers for Manager or Container High Availability"* on page 90.

   If preventing query data or state information loss during failover is important to you, set the "StoragePath" to the same shared directory for all Containers. If Persistence is enabled for your query module, the "StoragePath" preference for all Containers must be set to the same location.

   Make sure to specify the complete directory path and file name for the "StoragePath". For example:

   ```
   <section name="SybaseC8/Server">
      <section name="Common">
         ...
         <!-- the path on disk, under which the server
           components
         (directory, workspace mgr, qp state, etc) should
           persist -->
         <preference name="StoragePath"
           value=C:\Program Files\SybaseC8\Server\storage/>
   ```

4. If you are using file-based adapters, such as the CSV, Binary or XML input or output adapters, set the "SybaseC8/Adapters/ReadWriteBaseFolder" preference for all the Containers to the same shared directory, located on one machine and accessible to all the Containers in your cluster.

   This preference sets all the Containers in your Sybase CEP Server cluster to share read and write access.

   This directory is initially set in the adapters base folder setting during installation. By default, every Sybase CEP Server has its own unique adapters base folder. If the "SybaseC8/Adapters/ReadWriteBaseFolder" default settings are not changed for the Containers in your cluster, then you must copy the `.csv`, binary or `.xml` data files for your adapters to every adapters base folder included in your Cluster.

High Availability Troubleshooting Tips

A description of common high availability errors and troubleshooting tips.

If the Containers in your Sybase CEP Server cluster cannot connect to the active Manager and trying to establish connection with the Manager alias (the name specified in the "ManagerURI" preference) by using the ping tool generates the following error message:

```
Ping request could not find host
hostname
. Please check the name and try again.
```

you may be experiencing one of the following problems:

- The Manager process on Sybase CEP Server may not be running. Start the Sybase CEP Server instance on which the Manager process is installed and try connecting again.
- The Manager may not be configured correctly. Double-check and correct the following settings:
  - Make sure that the Manager's host name is set to the actual name of the machine and includes the proper domain.
  - Make sure that the Manager's port number is set to the correct value.
  - If you are using multiple Managers, make sure that each Manager's "ManagerUri" preference is set to its Uniform Resource Identifier (URI), not to its actual host name.
  - Make sure that the Manager's"NodeUri" preference is set to its host name and port number, not to its URI alias.
- The DNS server may not be working correctly. If multiple DNS servers are present (one "master" and one or more "slaves") the master may not be sending updates to the slave(s) promptly enough. Fix the problem and try connecting to the Manager again.
- The plugin settings in the "ManagerFailoverDDNSPlugin" section of the `c8-server.conf` configuration file may be incomplete or incorrect. Correct the plugin settings and try again.

If one or more Containers fail almost immediately after starting, and generate a message like the following (the actual file name in your error message may differ from the one listed in this example):

```
Error: Can not rename file
'C:\Program Files\SybaseC8\Server\Storage\manager.xml.
1155153553187500'     to
'C:\Program Files\SybaseC8\Server\Storage\manager.xml'
('The file is busy (-5936,0)').
```

some of the Containers in your cluster may be incorrectly enabled as Managers. Follow the steps listed in Step 1 of *"Preparing to Configure High Availability Features"* on page 86 for all the Containers in your cluster to resolve the problem.

Configuring Access Restrictions to Sybase CEP Engine

You can configure Sybase CEP Server to restrict specific actions and resources to specified IP addresses, host names, users and/or groups, thus protecting the desired Sybase CEP Engine resources from unauthorized use.

Setting Up Access Restrictions

By default, Sybase CEP Server is installed with no access restrictions. A description of how to configure access restrictions.

1. Create an `.xml` access control list (ACL) file specifying one or more access rules for your Sybase CEP Server.

   See *"Creating an ACL File"* on page 99 for more information. The `SybaseC8\Server\conf` directory contains a sample `c8-acl.xml` ACL file.

2. Set the "ACLFile" preference in the "SybaseC8/Security/AccessControl" section of the `c8-server.conf` file to the full path and name of the ACL file you created in the previous step, for example:

```
<section name="SybaseC8/Security">
  ...
  <section name="AccessControl">
     ...
     <preference name="ACLFile" value="C:\ProgramFiles
         \SybaseC8\Server\conf\myacl.xml"/>
     ...
  </section>
  ...
```

3. In the `c8-server.conf` file, add (or de-comment existing) information to specify which authentication method you want to use.

   See *"Sybase CEP Engine Authentication Plugins"* on page 112 for information about available methods.

4. Enable the *shared secret* functionality for any computers in your Sybase CEP Server cluster to which you want to grant connection access.

   A shared secret is a single predefined character string, which you specify for all Sybase CEP Servers in a cluster. The shared secret provides your cluster with an extra measure of security. When you enable this feature, the shared string must be exchanged between any two Sybase CEP Engine computers attempting to establish a connection.

   To enable the shared secret, set the "SharedSecret" preference of the "SybaseC8/Security/AccessControl" section of your `c8-server.conf` file to your desired shared secret, for example:

```
<section name="SybaseC8/Security">
  ...
  <section name="AccessControl">
     ...
     <preference name="SharedSecret" value="sharedsecret123/>
     ...
  </section>
  ...
```

The same shared secret string must be specified for all the computers in the cluster.

5. Once all the computers in your cluster have the shared secret in their `c8-server.conf` file, enable the shared secret feature in your ACL file by specifying a rule that includes the <SharedSecret/> subject, as explained in *"Creating an ACL File"* on page 99.
6. If you want to change the frequency with which the access control list is reloaded by Sybase CEP Engine, change the "FileReloadIntervalSeconds" preference, located directly under the "ACLFile" preference of the `c8-server.conf` file, to the desired reload interval.

   The default reload frequency is 30 seconds.

Note: To prevent security breaches to your access restriction policies, caused either by users tampering directly with the ACL file, or by redirecting the `c8-server.conf` file to point to a different ACL file, set file permissions to restrict user access to both the `c8-server.conf` file and the ACL file to which it points.

Creating an ACL File

A decription of how to create ACL files for use with Sybase CEP Engine. This section introduces access control ACL files.

This section discusses how to create access control ACL files for use with Sybase CEP Engine and how these files are used by Sybase CEP Engine. For tips on designing policies, see the *Designing Sybase CEP Access Control Policies* article available on the Sybase website.

The ACL file is composed of a single *policy set*, which contains one or more *policies*, each of which contains one or more *rules*. Every policy set must begin with a System Rule that enables shared secret use on the Manager and Container processes of your Sybase CEP Server or Sybase CEP Server cluster. The System Rule looks like this:

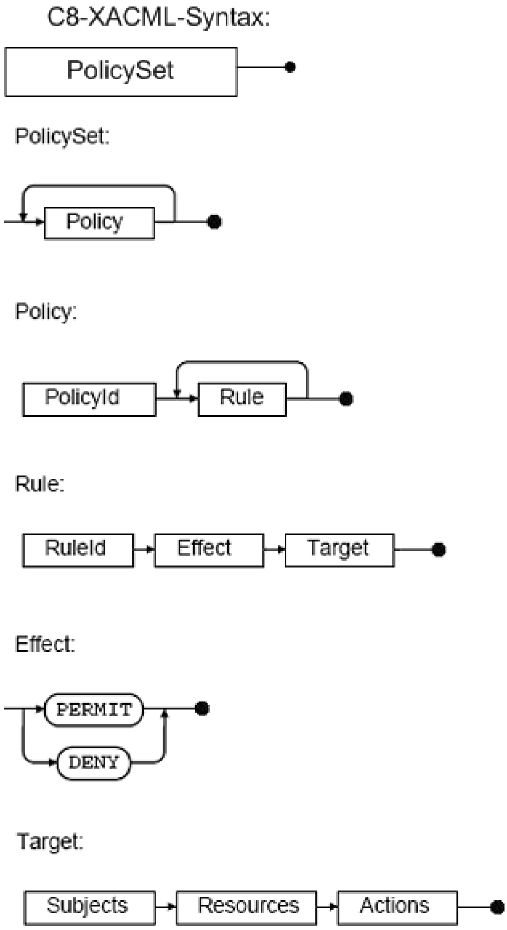
```
<PolicySet xmlns='
http://www.sybase.com/access_control/">
<Policy PolicyId="FirstPolicy">
   <Rule Effect="Permit" RuleID="SystemRule">
      <Target>
         <Subjects>
            <SharedSecret/>
         </Subjects>
         <Actions>
            <AnyAction/>
         </Actions>
         <Resources>
            <AnyResource/>
         </Resources>
      </Target>
   </Rule>
   ...
</Policy>
...
</PolicySet>
```

After this initial rule, you can create other rules inside the same policy, and other policies, as required. For an explanation of policy set, policy and rule components and syntax, see *"ACL File Components"* on page 100 .

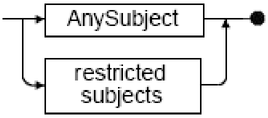
ACL File Components

Diagrams providing details on the syntax and usage of ACL file components.

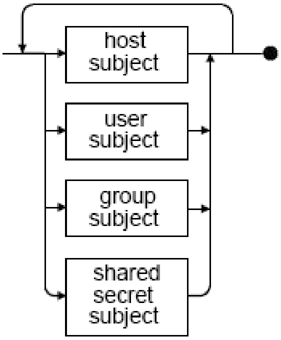
Sybase CEP Engine access control ACL files use a simplified version of the eXtensible Access Control Markup Language (XACML). The following diagram illustrates the overall structure of the IP-access ACL file:



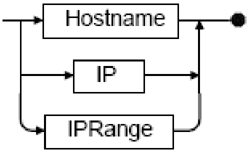
Subjects:



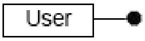
restricted-subjects:



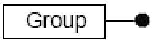
host-subject:



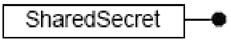
user-subject:



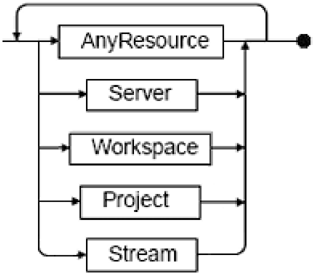
group-subject:



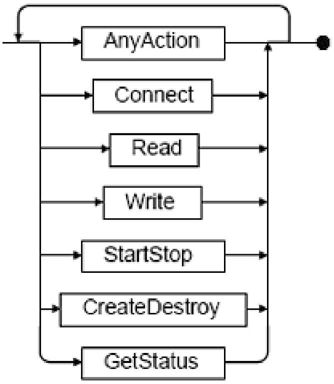
shared-secret-subject:



Resources:



Actions:



The rest of this section discusses in detail the syntax and usage of the components shown in the diagram. The syntax illustrations in the following discussions use several conventions:

- Items appearing in italics are stand-in values, for which you should replace with the actual information. For example, replace *policy-name* with the appropriate name for a policy.
- Square brackets ([ ]) show optional items.
- A pipe character (|) separates members in a list of items, from which you can choose only one item.
- The ellipsis marking inside square brackets ([...]) represents an optional repetition of the previous component type. You may repeat these components as many times as you want..

PolicySet

Syntax:

```
<PolicySet xmlns="
http://www.sybase.com/access_control/">
  policy
  [policy]
  [...]
</PolicySet>
```

Description:

The entire contents of the ACL must be contained in the "PolicySet" element, which specifies the CEP Engine access control policy set. The "PolicySet" element contains one or more "Policy" elements.

Policy

Syntax:

```
<Policy PolicyId="
policy-name
">
  rule
  [rule]
  [...]
</Policy>
```

Description:

The "Policy" element defines a single access policy. An ACL file's "PolicySet" contains one or more "Policy" elements. The "Policy" element requires:

- A "PolicyId" property, which specifies a unique name for the policy.
- One or more "Rule" elements.

Rule

Syntax:

```
<Rule RuleId="
rule-name
" Effect="
effect
">
  target
</Rule>
```

Effect: Permit or Deny

Description:

The "Rule" element defines a single rule for an access policy. A single "Policy" element contains one or more "Rule" elements. The "Rule" element requires:

- A "RuleId" property, which specifies a unique name for the rule.
- An "Effect" property, which specifies whether to permit or deny certain actions, which certain IP addresses, host names, users, or groups try to perform on certain Sybase CEP Engine resources. Note that the "Effect" property specifies only Permit or Deny. The specifics of what the rule is permitting or denying are included in the "Target" element.
- A single "Target" element.

Target

Syntax:

```
<Target>
  <Subjects>
      <AnySubject/> |
restricted-subjects
   </Subjects>
  <Resources>

resource
      [
resource
]
      [...]
  </Resources>
  <Actions>

action
      [
action
]
      [...]
  </Actions>
</Target>
```

restricted-subjects: Come in four varieties, as listed here. You can specify more than one subject.

Restricted Subject Type	Syntax Description

`host-subject`	May include any of the following: • <Hostname>hostname</Hostname> where hostname is a case-insensitive POSIX regular expression that matches a host name. For more information about POSIX regular expressions, see *http://www.boost.org/doc/libs/1_41_0/libs/regex/doc/html/boost_regex/syntax/basic_syntax.html*. • <IP>IP-address</IP> where IP-address is an IP address in xxx.xxx.xxx.xxx format • <IPRange>IP-address-range</IPRange> where IP-address-range is specified either in bitmask notation or range notation. • *bitmask notation:* Specifies an IP address in *XXX.XXX.XXX.XXX/YY* format, in which *XXX.XXX.XXX.XXX* is a bitmask and *YY* is the number of bits, counted from left to right, that are considered in the screening of IP addresses. This subject specification encompasses all addresses for which the first *YY* bits in the incoming IP address match the first *YY* bits of the bitmask you specified in *XXX.XXX.XXX.XXX*. Sybase CEP Engine ignores all bits to the right of those considered by *YY*. For example, if you specify an IP range as 144.227.192.2/ 16, this range accepts all addresses starting with 144.227 as valid subjects. • *range notation:* Specifies an IP address range in *IP1 - IP2* format, for example, 10.10.10.10 - 10.10.10.20. All IP addresses within the range are included in the subject specification.
`user-subject`	<User>user-name</User> where user-name is a case-insensitive POSIX regular expression that matches a defined user name.
`group-subject`	<Group>group-name</Group> where group-name is a case-sensitive POSIX regular expression that matches a defined user group name.
`shared-secret-subject`	<SharedSecret/>

resource: Resources can be set to any of the following values. Multiple resources can also be specified.

- <AnyResource/>
- <Server/>
- <Workspace>workspace</Workspace> where workspace is a case-insensitive POSIX regular expression that matches the name of a workspace on Sybase CEP Server.
- <Project>project</Project> where project is a case-insensitive POSIX regular expression that matches the name of a project in a workspace, using the workspace-name/project-name format.

- <Stream>data-stream</Stream> where data-stream is a case-insensitive POSIX regular expression that matches one or more Sybase C8 streams, using the workspace-name/project-name/[submodule-name][.../]data-stream-name format.

  You can get information used in specifying the "Stream" resource in the ACL file from Sybase CEP Studio, where it is displayed in the Properties View when the workspace, project or stream are selected. The CCL URI displayed in Sybase CEP Studio uses the following format:

  ```
  ccl://host-name:port/Stream/path
  ```

  The data-stream specification in the "Stream" resource uses the path portion of this URI, for example:

  ```
  myworkspace/myproject/instream
  ```

action: Actions can be set to any of the following values. Multiple actions can be specified.

- <AnyAction/>
- <Connect/>
- <Read/>
- <Write/>
- <StartStop/>
- <CreateDestroy/>
- <GetStatus/>

Description:

The "Target" element is contained within the "Rule" element. The "Target" contains three elements, which answer the following questions within the rule:

- The "Subjects" element answers the question of which hosts, users and/or user groups are permitted or denied access to a resource.
- The "Resources" element answers the question of which resources are being made accessible or inaccessible to the subjects.
- The "Actions" element answers the question of which actions pertaining to the resource are permitted or not permitted to the subject.

Specify the "Subjects", "Resources" and "Actions" elements, using the syntax detailed in the *syntax* section. These settings are further explained in this section:

- "Subjects": The empty "AnySubject" element (<AnySubject/>) creates a rule that applies to any host, user, or user group that attempts to perform the specified action on the specified resource. To restrict the rule to certain hosts, users, or user groups, use one or more host-subject, user-subject, and/or group-subject elements instead of "AnySubject".

  Specify a special empty shared-secret-subject (specified as <SharedSecret/>) once at the beginning of the ACL file to enable Sybase CEP Server connectivity.

host-subject elements include "Hostname", "IP" and "IPRange" elements. You can specify any number of these in any combination in a single "Target" element.

Note: Specifying Subjects to Connect to the Local Computer: Because of differences in the way that Microsoft Windows and UNIX-like operating systems handle connections to the local machine, Sybase advises that you configure subjects connecting to the local machine either with both the loopback IP address 127.0.0.1 and the external IP address, or with both localhost and the external host name.

user-subject elements include the "User" element. You can specify any number of "User" elements in a single "Target" element.

Note: User credentials are sent over the network as unencrypted text. To protect the user credentials from network analyzers, enable the Secure Socket Layer (SSL). See *"Configuring SSL Support"* on page 39 for more information about enabling SSL.

group-subject elements include the "Group" element. You can specify any number of "Group" elements in a single "Target" element.

If you specify multiple subjects in a single "Target", the rule applies to any host, user, or group that meets the criteria for at least one subject specification from each of the included groups. For example, a "Target" that includes a "Hostname" subject, an "IPRange" subject, and two "User" subjects applies to either of the indicated users when the computer from which they try to access the resource either matches the "Hostname" or falls within the "IPRange".

Here are some examples of "Subjects" elements:

```
<Subjects>
  <AnySubject/>
    </Subjects>
<Subjects>
  <Hostname>.*\.sybase\.com</Hostname>
    </Subjects>
<Subjects>
  <IP>192.168.10.10</IP>
  <IPRange>192.168.0.12/24</IPRange>
    </Subjects>
<Subjects>
  <IP>192.168.10.10</IP>
  <User>jdoe</User>
  <Group>qagroup</Group>
    </Subjects>
```

- "Resources": The empty "AnyResource" element (specified as <AnyResource/>) creates a rule that applies to any Sybase CEP Server resource requested by one of the subjects. To restrict the rule to certain resources, use the "Server", "Workspace", "Project", and/or "Stream" elements, in any combination. (The "Server" resource setting is typically used in conjunction with the "Connect" action, to permit or deny access to Sybase CEP Server.)

  If you identify multiple resources, the subjects are allowed or denied access to any of the specified resources. For more information about how a resource hierarchy is used to grant or deny access to various resources, see *"ACL Policy Set Implementation"* on page 108.

Here is an example of a "Resources" element:

```
<Resources>
 <Stream>Production/.*/StreamIn</Stream>
 <Stream>Production/.*/StreamOut</Stream>
    </Resources>
```

- "Actions": The "AnyAction" element (specified as <AnyAction/>) creates a rule that applies to any actions that the subject attempts on the specified resource. To restrict the rule to certain actions, use the "Connect", "Read", "Write", "Start/Stop", "Create/Destroy", and/or "GetStatus" elements, in any combination. If you identify multiple resources, the subjects are allowed or denied access to any of the specified actions. All elements contained inside an "Actions" element are empty. Here is an example of an "Actions" element:

  ```
  <Actions>
    <StartStop/>
      </Actions>
  ```

  Action requests are interrelated. The ability to perform certain actions depends on access to certain other types of actions:

  - Connection to a resource should be enabled in order to allow any other action on that resource. Connections can be enabled explicitly by using the "Connect" action, or by using the "AnyAction" action. If the connection is not enabled, the subject is not allowed to perform actions on the resource, even if a rule states that those actions are permitted.
  - To enable "GetStatus" for a status stream, the subject should also be enabled to "Read" from the status stream.
  - To enable SOAP calls to access resources, the resource should also be enabled to "GetStatus", "StartStop" or "Read".

ACL Policy Set Implementation

When access control is enabled, Sybase CEP Engine checks incoming requests for actions on resources against the rules contained in a policy. This section describes the general rules that apply to all policy sets.

- Sybase CEP Engine checks rules in hierarchical order of resources, from the most specific to the most general.

  For example, if an explicit rule permitting or denying access to a project is defined, the rule is applied. If a matching rule for the project is not specified, but a rule for the project's workspace is defined, Sybase CEP Engine applies the workspace rule. If a rule for the workspace is not defined either, then Sybase CEP Engine searches the ACL file for a rule applying to the Sybase CEP Server on which the project is located.
- On each level of the resource hierarchy, Sybase CEP Engine applies the first matching rule, if one is found, and ignores any subsequent rules. A matching rule is a rule that applies to the subject and permits or denies the subject the desired action type on the desired resource.

For example, if one rule is defined that allows a particular user connect privileges to a specified project, and a second rule, later in the file, that denies access to the same user and project, Sybase CEP Engine uses the first rule.

- Sybase CEP Engine always considers "AnyResource" to be a match for any resource at any level in the hierarchy.

  For example, if one rule is defined that denies a particular host name connection privileges to "AnyResource", and a second rule, later in the file, that permits the same host name connection privileges to a particular workspace, Sybase CEP Engine considers the "AnyResource" rule to be the first match, and denies access to the workspace.
- Similarly, Sybase CEP Engine always considers "AnyAction" to be a match for any requested action and "AnySubject" to be a match for any subject requesting the action.
- If Sybase CEP Engine does not find an applicable rule, it denies the subject access to the action on the resource.

The following table shows the actions and resources required to complete specific tasks, and the order in which the rules controlling access to these resources are checked. Where the requested task requires permission to perform more than one type of action, the rules must permit all the required action types, or must permit "AnyAction". Note that the rules permitting an action must appear earlier in the ACL file than any rule(s) denying the action. Likewise, rule(s) permitting access to a resource must appear earlier in the file than any rule(s) denying access to the resource. Otherwise, the action, or access to a resource is denied.

Table 2. Rule Implementation Order

When a Subject Tries to Perform This:	Search for a Match on These Actions:	Search in Order, for a Match on One of These Resources:

Connect to Sybase CEP Server and see a list of workspaces.	`<AnyAction/>`
	or
	`<Connect/>`
	and
	`<GetStatus/>`
		`<AnyResource/>`
		If not found:
		`<Server/>`

When a Subject Tries to Perform This:	Search for a Match on These Actions:	Search in Order, for a Match on One of These Resources:

Create or destroy a workspace on Sybase CEP Server.	`<AnyAction/>`
	or
	`<Connect/>`
	and
	`<CreateDesroy/>`
		`<AnyResource/>`
		If not found:
		`<Workspace>workspace-name</Workspace>`
		If not found:
		`<Server/>`

Connect to a workspace on Sybase CEP Server.	`< AnyAction/>`
	or
	`<Connect/>`
	and
	`<GetStatus/>`
		`<AnyResource/>`
		If not found:
		`<Workspace>workspace-name</Workspace>`
		If not found:
		`<Server/>`

Start or stop a project in a workspace.	`< AnyAction/>`
	or
	`<Connect/>`
	and
	`<StartStop/>`
		`<AnyResource/>`
		If not found:
		`<Project>workspace-name/project-name</Project>`
		If not found:
		`<Workspace>workspace-name</Workspace>`
		If not found:
		`<Server/>`

When a Subject Tries to Perform This:	Search for a Match on These Actions:	Search in Order, for a Match on One of These Resources:

Read from a stream or write to a stream in a project in a workspace.	`< AnyAction/>`
	or
	`<Connect/>`
	and
	`<Read/>`
	or
	`<Connect/>`
	and
	`<Write/>`
		`<AnyResource/>`
		`If not found:`
		`<Stream>/workspace-name/project-name/[submodule-name/][.../]stream-name</Stream>`
		If not found:
		`<Project>workspace-name/project-name</Project>`
		If not found:
		`<Workspace>workspace-name</Workspace>`
		If not found:
		`<Server/>`

Query a public window in a project in a workspace (requires both "Connect" and "Read" permission).	`<AnyAction/>`
	or
	`<Connect/>`
	and
	`<Read/>`
		`<AnyResource/>`
		If not found:
		`<Project>workspace-name/project-name</Project>`
		If not found:
		`<Workspace>workspace-name</Workspace>`
		If not found:
		`<Server/>`

Sybase CEP Engine Authentication Plugins

Describes Sybase CEP authentication plugins. Sybase CEP Engine provides a user-level SDK for creating authentication plugins that you can then enable to verify user identities and determine user membership in groups.

The Sybase CEP Engine distribution also includes three built-in sample plugins:

- *Plain Text Apache-Format htpasswd Plugin* on page 113
- *LDAP Authentication Plugin* on page 115
- *Pluggable Authentication Modules (PAM)* on page 115

The `c8-server.conf` file includes sections for each of these plugins, enclosed in comment marks. These sections and the plugins they enable are discussed in more detail in the following sections.

Before setting up authentication plugins, make sure that you have set up access restrictions for Sybase CEP Engine. See *"Setting Up Access Restrictions"* on page 98 for more information.

Creating and Configuring Authentication Plugins

A description of how to configure a plugin in the `c8-server.conf` file.

Authentication plugins are compiled `.dll` files (on Microsoft Windows) or `.so` files (on UNIX-like operating systems). Sybase CEP Server calls the methods in these libraries when it authenticates a user. For instructions on creating your own authentication plugins, see the Sybase CEP *Integration Guide* .

Once the plugin library is created, configure the plugin in the `c8-server.conf` file by creating a section for it in the "SybaseC8/Security/AccessControl/Authentication" section of the file. Here is an example authentication plugin entry:

```
<section name="C8/Security">
...
<section name="AccessControl">
 ...
 <section name="Authentication">
    <section name="Plugin">
       <preference name="LibraryName"
                  value="c8authplugin_demo_lib"/>
       <preference name="InitializeFunction"
                   value="c8authplugin_demo_initialize"/>
       <preference name="AuthenticateFunction"
                   value="c8authplugin_demo_authenticate"/>
       <preference name="ShutdownFunction"
                     value="c8authplugin_demo_shutdown"/>
    </section>
 ...
```

Include the following information in your plugin configuration:

1. Set the "LibraryName" preference to the name of the library containing your authentication plugin.

   Do not include the `.dll` or `.so` extension in the name, as Sybase CEP Server automatically adds this extension. If you are configuring the plugin on a UNIX-like operating system, also omit the `lib` prefix, which is also automatically added by Sybase CEP Server.
2. Set the "InitializeFunction" preference to the initialize function for the plugin.
3. Set the "AuthenticateFunction" preference to the authenticate function for the plugin.
4. Set the "ShutdownFunction" preference to the shut down function for the plugin.

The configuration file section that contains the authentication plugin entry must also define any parameters that you want to pass to the plugin functions, using the following syntax:

```
<preference name="
parameter-name
" value="
parameter-value
"/>
```

For example:

```
<preference name="MyPreference" value="MyParameterValue"/>
```

Plain Text Apache-Format htpasswd Plugin

A description of the Plain Text Apache-Format htpasswd plugin, which accesses Apache-format `htpasswd` and `htgroup` files to perform user and group authentication.

To use this plugin:

1. Create an `htpasswd` file.

   For details regarding this file and the Apache htpasswd utility, see

   ```
   http://httpd.apache.org/docs/2.0/programs/htpasswd.html
   ```

   Note: This plugin supports only apache's MD5 encryption format. Use the Apache

   ```
   htpasswd
   ```

   command with the

   ```
   -m
   ```

   option to create htpasswd records and ensure proper encryption.
2. If you are defining groups, create a `htgroup` file, using the following format for each group:

   ```
   group-name
   ```

```
:
user-name
,
user-name
,
user-name
```

where

```
group-name
```

is the name of a group and

```
user-name
```

is a user name associated with the group. You can define multiple groups, one under the other. Each group may include multiple users.

3. Configure the following entry in the "SybaseC8/Security/AccessControl/Authentication" section of the `c8-server.conf` file:

```
<section name="SybaseC8/Security">
    ...
    <section name="AccessControl">
   ...
   <section name="Authentication">
      <section name="Plugin">
         <preference name="LibraryName"
             value="c8_server_plugins_lib"/>
         <preference name="InitializeFunction"
             value="c8_auth_plugin_htpasswd_initialize"/>
         <preference name="AuthenticateFunction"
             value="c8_auth_plugin_htpasswd_authenticate"/>
         <preference name="ShutdownFunction"
             value="c8_auth_plugin_htpasswd_shutdown"/>
         <preference name="PasswordFilePath"
             value="
path-to-htpasswd-file
\htpasswd.txt"/>
         <preference name="GroupFilePath"
             value="
path-to-htgroup-file
\htgroup.txt"/>
       </section>
   ...
```

where

```
path-to-htpasswd-file
```

is the directory path to your `htpasswd.txt` file and

```
path-to-htgroup-file
```

is the directory path to your `htgroup.txt` file.

LDAP Authentication Plugin

A description of how to configure the LDAP Authentication plugin. These steps require you to modify the `c8-server.conf` file.

To configure this plugin, open the `c8-server.conf` file and remove the comment markings around the LDAP plugin entry in the "SybaseC8/Security/AccessControl/Authentication" section of the file. For a list of parameters supported by the LDAP Authentication Plugin, see the Sybase CEP *Integration Guide* .

Note: Make sure to enable SSL or Transport Layer Security (TLS)-protected connection on your LDAP server to prevent user credentials from being visible to network analyzers.

Pluggable Authentication Modules

A description of how to implement Pluggable Authentication Modules (PAM) on Unix-like operating systems.

PAMs enable you to change authentication techniques (for example, from password to retina scan) by merely changing a configuration file, rather than forcing you to recompile the program (for example with Sybase CEP Server). Before setting up PAM, make sure that you have set up access restrictions for Sybase CEP Engine. See *"Setting Up Access Restrictions"* on page 98 for more information.

To use PAM, you need to update your `c8-server.conf` file as explained in *"Creating and Configuring Authentication Plugins"* on page 112 and in this section. You also need to perform other steps, which are described in the Sybase CEP *Integration Guide* .

The default `c8-server.conf` file included with your Sybase CEP Engine contains a commented-out section with PAM-related configuration parameters:

```
<!-- Sample PAM authentication plugin configuration -->
<section name="Plugin">
<preference name="LibraryName" value="c8_server_plugins_lib"/>
<preference name="InitializeFunction"
value="c8_auth_plugin_pam_initialize"/>
<preference name="AuthenticateFunction"
value="c8_auth_plugin_pam_authenticate"/>
<preference name="ShutdownFunction"
value="c8_auth_plugin_pam_shutdown"/>
<!--
  Uncomment the following line and change 'value' to
  set the password prompt that PAM sends to applications.
  The defaultprompt is "Password: ". (Note the required
  space at the end.)  You should only need to do this if
  you encounter a system that does not use the default,
  and as such, it failing authentication.
 -->
 <!-- <preference name="PasswordPrompt" value="Password: "/> -->
</section>
```

Since Sybase CEP Engine includes the C8 PAM library, the names of the library and the functions in it are already included in the example entry. If you are using the C8 PAM library, you can simply remove the comment marks for the section, and do not need to change the configuration preferences.

Query Module Persistence

A description of the Query Module Persistence features, which offer information protection if Container failure occurs.

A stream-oriented query system such as Sybase CEP Engine stores data in memory, instead of in tables on a long-term storage medium, such as a hard disk drive. Under normal circumstances, Sybase CEP Engine automatically saves window and query state. In the event of Container failure, however, Sybase CEP Engine loses window state information, and may lose some rows of data during failover.

Query Module Persistence features minimize loss of rows and state information in the event of Container failure. When Persistence is turned on in a project, Sybase CEP Server stores information about the state of named windows, queries, adapters, and data streams on the disk drive, in the `storage` directory. The default location of the Storage directory is in SybaseC8/ Server/Storage. This allows a new Container to recapture most of the state leading up to the failure. Of course, the more data you write to the disk, the slower your performance is; turning on Persistence results in lower maximum performance, though the extent of the reduction depends in part on your query structure.

If you want a Container to replace another Container in the event of Container failure, and you have enabled Persistence, the replacement Container must be able to read the same storage - such as a disk drive - that the failed Container was using. Furthermore, if the storage device crashes, the replacement Container will not be able to take advantage of the Persistence feature. To ensure against loss of data due to hardware failure, you may wish to use a combination of RAID and dual-port disk drives.

For information about using Persistence settings in Sybase CEP Studio, see the Sybase CEP *Studio Guide* .

Note that, while enabling Persistence features drastically reduces the amount of information that may be lost if a Container fails, it does not ensure that Sybase CEP Engine retains all rows and state information. To ensure that no rows are lost in the event of Container failure, use Persistence features in combination with Guaranteed Delivery features, described in the Sybase CEP *Integration Guide* , the Sybase CEP *CCL Reference Guide* and the Sybase CEP *Studio Guide* .

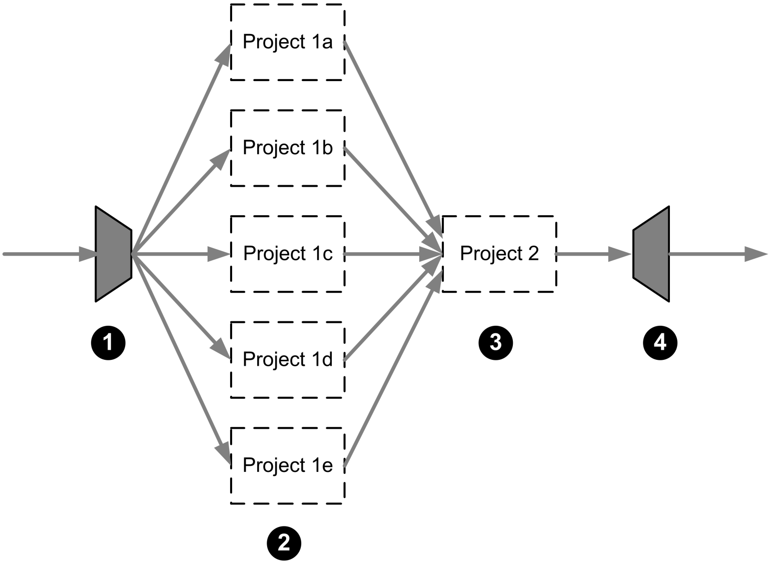
Parallel Queries

You can often dramatically improve the performance of large Sybase CEP applications by distributing the queries in your Sybase CEP Engine project into several sets of data and processing each set on a separate machine. This feature is called Parallel Queries, or parallel processing.

For example, consider a stock exchange that calculates daily averages for 5,000 stocks. If time is not an issue, all 5,000 stocks can be processed by a single project on one computer. However, performance will be greatly enhanced if the 5,000 stocks are distributed across 5,000 computers, with each computer calculating the average for a single stock. In this example, you might route prices for each stock to the appropriate computer as they arrive, then calculate the average for each stock and gather the final data once more on a single computer.

Here is a more realistic example. Suppose that you have five computers processing 5,000 stocks. You allocate 1,000 stocks to each computer. Arriving data is routed to the appropriate computer to calculate the daily average and to perform other calculations. In this scenario, the average calculation would be performed approximately five times faster than would be the case with only one computer doing all the work.

Here is an illustration of an application using parallel queries, with an explanation of the role played by each component:



1. A user-defined out-of-process adapter splits the incoming data into multiple streams and then sends each stream to one of the project instances.
2. Five identical project instances process the data in parallel.
3. When the processing is finished, each query processor project instance sends its data to another project, which is responsible for merging the data back into a single stream. This project may also include other queries that process the merged data.
4. Sybase CEP Engine then sends the merged data to an external destination through an output adapter.

Deciding When to Use Parallel Queries

Examples of situations where you can and cannot use parallel queries.

Not all problems can be partitioned as neatly as the average price calculation described in *"Parallel Queries"* on page 117. For example, suppose you are using a bank management application to partition all bank ATM (Automated Teller Machine) transactions and handle them in parallel projects. If you partition the transactions by bank branch, the application processes ATM records for each branch very quickly and keeps track of how much money is in the ATM of each branch. However, since different customers may use different bank branches, queries involving customer accounts are processed by computers that have all the information about a customer's transactions at one branch, but no information about the customer's transactions at other branches.

If, on the other hand, you partition the data by customer ID and assign one computer to every 10,000 customers, customer balances are calculated quickly, but queries involving the calculation of money balances in a particular branch cannot be performed by a single computer. In both cases, the different computers running Sybase CEP Server have to exchange information, which precludes the use of parallel queries.

Furthermore,you can improve performance by distributing your queries.

Divide data into separate streams and then quickly merge it back into one stream when the parallel queries finish their processing. The performance benefits you gain from parallel processing are reduced if the overhead of partitioning or recombining the streams is high.

The rest of this chapter assumes that you can partition your data into independent sets, in which each set contains all the data necessary to perform the computations you want to run.

Parallel Query Setup

A description of how to configure parallel queries.

1. Create a Sybase CEP Server cluster, consisting of two or more Sybase CEP Server instances running Container processes (preferably on different machines) all of which point to the same Manager URI.

For more information about Containers and Managers, see *"High Availability"* on page 83. For more information about configuring a Sybase CEP Server cluster, see *"How to Configure Sybase CEP Server for High Availability"* on page 85.

2. Set a multiple number of instances for the project you want to run.

   The project instances are duplicates of the project that run parallel to one another. When you run the project, Sybase CEP Server automatically starts the number of project instances you specify. If you are using Sybase CEP Studio, see the Sybase CEP *Studio Guide* for instructions on setting the number of project instances in the project's Properties View.

   Note: When you replicate a project in multiple instances, all data streams attached to the project are replicated too. Streams associated with each project instance have unique identifiers. To refer to a stream in a specific instance of the project, specify the Stream URI, with an additional `~ instance-number` at the end of the URI, for example:

   ```
   ccl://localhost:6789/Stream/Default/TwoWindowJoin/ MsftTrades~2
   ```

3. Use one of the Sybase CEP SDK to write and install an out-of-process adapter which partitions the incoming data and sends each partitioned set to an instance of the project.

   For information see the Sybase CEP *Integration Guide* .

   Alternately, you can partition data by explicitly publishing rows from queries in one project to data streams associated with specific instances of another project.

4. Create another Sybase CEP Engine project and configure it to merge the data output from the parallel queries, as well as to perform any necessary operations on the merged data.

   Note: Since the data from the multiple instances of the parallel output stream comes from different machines, the merged data may be out of order. To order the data correctly, set the stream data timestamp to the current time, or allow the messages to come out of order, and set the maximum out of order delay. For instructions on setting data stream properties, see the Sybase CEP *Studio Guide* .

Configuring the Container Load Limit

A description of how to configure the Container load limit. The Container load limit setting enables you to specify the maximum number of CCX modules that a given Container may simultaneously execute.

Once the Container reaches its load limit, the Manager process makes the Container unavailable for additional work until its load is reduced. This setting is particularly valuable if you are running a high-volume application requiring a parallel query solution. A low load limit prevents multiple instances of the same project from running on the same machine.

To configure the load limit, set the "SybaseC8/Server/Container/LoadLimit" preference in the Container's `c8-server.conf` file to the desired number. This preference can be set to accept non-negative integers. The default setting for the property is 0 (unlimited).

Note: If you run more than one set of parallel projects, the load distribution of your projects across the available machines may still be uneven. For example, one machine may run two instances of ParallelProject1, while another machine runs two instances of ParallelProject2.

Crash Recovery and High Availability

Crash recovery and High Availability features for parallel query applications operate the same way for single instance applications. If a Sybase CEP Server instance is running an instance of a parallel project and it fails, the project instance restarts on another Sybase CEP Server instance if High Availability features are enabled and another Container is available to take over the work.

Running Parallel Sybase CEP Applications

A description of how to run parallel Sybase CEP applications.

To start parallel Sybase CEP applications, execute these steps, in order:

1. Start the Sybase CEP Server instance(s) on which the Manager processes for your cluster are enabled.
2. Start the Sybase CEP Server instances on which the Container processes for your cluster are enabled.
3. Start the project containing the queries that merge the results from your parallel queries.
4. Start the project you have set up to run parallel queries.
5. Start the out-of-process adapter, or the project that supplies data to the parallel queries.

Migration Information

A description of how to migrate from previous versions to Sybase CEP R4. Includes instructions on how to migrate configuration files, projects, SDK applications, and adapters.

Migrating from Sybase CEP R3 to Sybase CEP R4

Migration steps for migrating from Sybase CEP R3 to Sybase CEP R4.

Do not uninstall Sybase CEP R3 prior to installing Sybase CEP R4. When installing Sybase CEP R4, do not install over your installation of Sybase CEP R3. Instead, install Sybase CEP R4 Server and Studio to a different folder and choose a new location for the adapters base folder.

Once installation of Sybase CEP R4 is complete, perform the following migration steps.

Note: Existing Sybase CEP R3 license files cannot be used with Sybase CEP R4. You will need to obtain a license file for Sybase CEP R4 from Sybase, and follow the provided instructions for installing the license file.

Configuration Files

If you want to retain your Server settings from Sybase CEP R3, copy the configuration files located in the Server configuration directory of your Sybase CEP R3 installation over to the corresponding directory in your Sybase CEP R4 installation.

In your configuration files, update directory paths to point to the Sybase CEP R4 installation directory instead of the Sybase CEP R3 directory.

Projects

To migrate your existing Sybase CEP R3 projects over to Sybase CEP R4:

1. Locate your Sybase CEP R3 project directories (for example, C:\Documents and Settings \user-name\My Documents\SybaseC8Repository\3.0) and copy these directories to the repository for Sybase CEP R4 (for example, C:\Documents and Settings\user-name\My Documents\SybaseC8Repository\4.0).
2. Redeploy your projects to the Sybase CEP Server R4 using the Sybase CEP Studio R4.

   If any of your projects use custom adapters, see the following section for steps required to migrate your custom adapters.

SDK Applications and Adapters

Existing adapters and applications written using the Sybase CEP R3 SDK can run without code changes on Sybase CEP R4. However, it is recommended that you recompile your code against the SDK libraries and header files included with Sybase CEP R4. In addition, if you have written custom in-process adapters, copy the `.adl` files for your custom adapters located in the Sybase CEP R3 `studio\plugins` and `server\plugins` directories over to the corresponding directories in the Sybase CEP R4 installation.

Migrating from Sybase CEP R2, Coral8 5.5 or Coral8 5.6 to Sybase CEP R4

Migration steps for migrating from Sybase CEP R2, Coral8 5.5 or Coral8 5.6 to Sybase CEP R4.

Before starting your migration, familiarize yourself with these prerequisites and recommendations:

- Do not uninstall Coral8 5.5/Coral8 5.6/Sybase CEP R2 prior to installing Sybase CEP R4. However, when installing Sybase CEP R4, do not install over your installation of Coral8 5.5/Coral8 5.6/Sybase CEP R2. Instead, install Sybase CEP R4 Server and Studio to a different folder and choose a new location for the adapters base folder.
- In addition to the migration steps outlined in this document, consider using the migration utility located at *http://www.sybase.com/detail?id=1090893*.
- Be aware of differences in the way Coral8 5.5, Coral8 5.6, and Sybase CEP R4 are licensed. See *Migrating from Coral8 to Sub-capacity Environments* for details.
- Create a test plan for your migration. Once the migration is complete, test it thoroughly before moving it into production.
- Familiarize yourself with the changes and additions introduced in Sybase CEP R4. See *New and Changed Functionality* in the *Sybase CEP R4 Release Bulletin*.

  Be particularly aware of the change in behavior for database statement and RPC error handling. Sybase CEP R4 can interact with remote databases and RPCs supported by the execute database statement/RPC and join clause syntax, which accepts a remote database or RPC as a data source. If there are errors in the interactions, the project stops.

  To ensure your existing Coral8 projects function properly with this changed behavior, either:

  - Alter your CCL to include the ON ERROR clause or
  - Set the global IgnoreErrors flag to true in the given database's entry in the `c8-services.xml` file

Migrating from Coral8 to Sub-capacity Environments

If you are migrating from Coral8 and need to run Sybase CEP on machines with a core capacity that exceeds your licensed cores, please contact Sybase Customer Service for assistance. Customer Service will confirm your eligibility to receive supplemental licenses permitting Sybase CEP to operate successfully in such an environment. Please note that you will be required to limit your use of Sybase CEP to the number of licensed cores specified in your Sybase contracts.

The Coral8 Engine can run on servers where the total number of cores exceeds the number of licensed cores. In such cases, the Coral8 Engine establishes an affinity to a subset of the server's cores so that only the number of licensed cores is used. The Coral8 Engine determines the number of licensed cores from the Coral8 license file. Check your `coral8-server.conf` file to see if the **ProcessorAffinity** parameter is set. If it is, then this situation applies to you.

Sybase CEP license checking works differently than license checking in the Coral8 Engine. The Sybase CEP license check fails if it finds that the number of licensed cores falls below the total number of cores. Use the **cpuinfo** utility provided with Sybase CEP to determine the total number of actual cores and licenses expected by Sybase CEP. Refer to this sample output from the utility (emphasis added):

```
% $SYBASE_C8/server/SYSAM-2_0/bin/cpuinfo
```

```
Detected 32 logical processors, 8 cores, in 1 chip
```

On such a machine, Sybase CEP attempts to check out eight cores of licenses on startup. If the license check fails, Sybase CEP shuts down.

Note: Sybase CEP has a thirty day installation grace period which can mask license issues. During this period, Sybase CEP does not shut down upon failure of the license check, so review your Sybase CEP log file for license messages immediately following migration.

To proceed with a migration from Coral8 to a Sybase CEP environment that has a core capacity exceeding your licensed cores, first open a case with Sybase Customer Service to check your eligibility for supplemental licenses for your Sybase CEP products. If eligible, you will be provided with supplemental licenses that will enable the Sybase CEP license check to succeed.

Second, for compliance purposes, limit the actual number of cores Sybase CEP will use for processing to the number of cores specified in your license agreements with Sybase. To limit the cores Sybase CEP uses, configure the **ProcessorAffinity** parameter in the Sybase CEP `c8-server.conf` configuration file of each server. The parameter specifies which cores to leverage by number, starting with 0. Here is an example that restricts Sybase CEP to use just the first two cores on a machine for processing:

```
<preference name="ProcessorAffinity" value="0,1"/>
```

After setting the parameter, run a command such as **top** to verify that the `c8_server` processes only use the cores to which they are bound.

Configuration Files: Server

Compare your existing server configuration files to the new Sybase CEP R4 files. If you have made few modifications to your existing Coral8 5.5/Coral8 5.6/ Sybase CEP R2 files, you may edit the Sybase CEP R4 files to reflect the previous customizations and skip the steps below. Also, change preference paths from `Coral8/...` to `C8/...`. However, if the modifications to the server configuration files are significant, follow the steps below.

To retain your Server settings from Coral8 5.5/Coral8 5.6/Sybase CEP R2:

1. Copy the files in the Coral8 5.5/Coral8 5.6/Sybase CEP R2 Server configuration directory (for example, `C:\Program Files\Coral8\Server\conf`) to a temporary location.
2. Go to the temporary location and rename all the files from `coral8-`... to `c8-`....
3. For each renamed file:
   a) Change any XML namespaces containing *www.coral8.com* to *www.sybase.com*, and *schema.coral8.com* to *schema.sybase.com*.
   b) Change configuration section names containing *Coral8/…* to *C8/…*.
   c) Update directory paths referring to Coral8 to reflect the Sybase CEP R4 install directory.
   d) Change any service names starting with *Coral8…* to *C8…*.
4. Copy the updated configuration files to the Sybase CEP R4 Server configuration directory (`C:\Program Files\SybaseC8\Server\conf`) to overwrite the default configuration files.

   Important: Install Sybase CEP R4 the same as Coral8 5.5/Coral8 5.6 (container or manager). For example, if Coral8 was installed as a container, install Sybase CEP as a container also.

Configuration Files: Studio

Prerequisites

Installation of Sybase CEP R4 is complete.

Task

To retain your Studio settings from Coral8 5.5/Coral8 5.6/Sybase CEP R2:

1. Locate the Coral8 Studio preferences file `studio-preferences.xml` (for example, `C:\Documents and Settings\<user-name>\My Documents\Coral8 Repository\5.6`) and copy it over to the Sybase CEP R4 repository (for example,

`C:\Documents and Settings\<user-name>\My Documents \SybaseC8Repository\4.0`).

2. Edit the copied preferences file:
   a) Change the XML namespace from *http://schema.coral8.com/preference/2004/05* to *http://schema.sybase.com/preference/2004/05*.
   b) Update the preference name prefixes from *Coral8/…* to *C8/…*.
   c) Update any preferences that reference the Coral8 repository path (for example, `C:\Documents and Settings\<user-name>\My Documents\Coral8 Repository\5.6`) to use the Sybase CEP R4 repository path (for example, `C:\Documents and Settings\<user-name>\My Documents \SybaseC8Repository\4.0`).
3. To retain your workspace settings:
   a) Copy the files ending in "`.cce`" from the Coral8 repository (for example, `C:\Documents and Settings\<user-name>\My Documents\Coral8 Repository\5.6`) to the Sybase CEP R4 repository (for example, `C:\Documents and Settings\<user-name>\My Documents \SybaseC8Repository\4.0`).
   b) For each copied file, change the XML namespace from *http://www.coral8.com/cpl/2004/04/* to *http://www.sybase.com/cpl/2004/04/*.

   Note that the migrated Studio and workspace settings may have dependencies on projects being migrated as well.

License File

The existing Coral8 5.5/Coral8 5.6/Sybase CEP R2 license files cannot be used with Sybase CEP R4. Obtain a new license file from the Sybase Product Download Center, and follow the provided installation instructions.

Projects

Prerequisites

Installation of Sybase CEP R4 is complete.

Task

Migrate your Coral8 5.5/Coral8 5.6/Sybase CEP R2 projects over to Sybase CEP R4:

1. Locate your Coral8 5.5/Coral8 5.6/Sybase CEP R2 project directories (for example, `C:\Documents and Settings\<user-name>\My Documents\Coral8 Repository\5.6`) and copy these directories to the Sybase CEP R4 repository (for example, `C:\Documents and Settings\<user-name>\My Documents \SybaseC8Repository\4.0`).

2. For each copied project, edit the `.ccp` file and change the XML namespace from *http://www.coral8.com/cpl/2004/04/* to *http://www.sybase.com/cpl/2004/04/*.
3. Repeat step 2 for any `.ccl` and `.ccs` files located in the modules and schemas subdirectories.
4. Recompile the project to build new ccx so the project is ready for execution.

   Note: If your projects use custom adapters, refer to the section on SDK Applications and Adapters for additional steps required to migrate your custom adapters.

SDK Applications and Adapters

To facilitate migration, Sybase CEP R4 provides wrappers that allow applications and adapters written using the Coral8 5.5/Coral8 5.6/Sybase CEP R2 SDKs to function with Sybase CEP R4 without any code changes. The migration instructions for each SDK are given below.

SDK Applications and Adapters

Migration instructions for applications and adapters written using Coral8 5.5/Coral8 5.6/ Sybase CEP R2 SDKs.

SDK	Prerequisites	Migration Instructions
C/C++	Installation of Sybase CEP R4 is complete.	Adapters and applications written using the /Coral8 5.6/ C/C++ SDK do not require any code changes to work with Sybase CEP R4. However, a recompile against the Sybase CEP R4 libraries is required. For in-process adapters, a simple update of the `.adl` files is required. • Copy the `.adl` files for your custom adapters located in the /Coral8 5.6/ Studio plug-ins directory (for example, `C:\Program Files\Coral8\Studio\plug-ins`) over to the Sybase CEP R4 Studio plug-ins directory (for example, `C:\Program Files\SybaseC8\Studio\plug-ins`). • Edit each copied `.adl` file and change the XML namespace from *http://www.coral8.com/adl/2005/07/* to *http://www.sybase.com/adl/2005/07/*, and *http://www.coral8.com/cpx/2004/03/* to *http://www.sybase.com/cpx/2004/03/*. Your custom in-process adapters are now listed in Sybase CEP R4 Studio.
Java 1.5 SDK	Installation of Sybase CEP R4 is complete.	Adapters and applications written using the Coral8 5.5/ Coral8 5.6/Sybase CEP R2 Java 1.5 SDK do not require any code changes to work with Sybase CEP R4. However, a recompile against the Sybase CEP R4 libraries is required. Ensure that your Java classpath is referring to the `c8-sdk-java5.jar` included with Sybase CEP R4.

SDK	Prerequisites	Migration Instructions
Java 1.4 SDK	Installation of Sybase CEP R4 is complete.	The Coral8 5.5/Coral8 5.6/Sybase CEP R2 Java 1.4 SDK is not supported by Sybase CEP R4. Adapters written using the Coral8 5.5/Coral8 5.6/Sybase CEP R2 Java 1.4 SDK need to be updated to call into the Sybase CEP R4 Java 1.5 SDK APIs. However, a recompile against the Sybase CEP R4 libraries is required.
.NET 3.5 SDK	Installation of Sybase CEP R4 is complete.	Adapters and applications written using the Coral8 .NET 3.5 do not require any code changes to work with Sybase CEP R4. However, a recompile against the Sybase CEP R4 libraries is required. Ensure that your executable is referring to c8-4.0.0.dll included with Sybase CEP R4 instead of the Coral8 5.5/Coral8 5.6/Sybase CEP R2 Coral8.dll.
.NET 2.0 SDK	Installation of Sybase CEP R4Sybase CEP R4 is complete.	The Coral8 5.5/Coral8 5.6/Sybase CEP R2 .NET 2.0 SDK is not supported by Sybase CEP R4. Adapters written using the Coral8 5.5/Coral8 5.6/Sybase CEP R2 .NET 2.0 SDK need to be updated to call into the Sybase CEP R4 .NET 3.5 SDK APIs. Also, a recompile against the Sybase CEP R4 libraries is required.
Flex SDK	Installation of Sybase CEP R4 is complete.	Applications written using the Coral8 5.5/Coral8 5.6/Sybase CEP R2 Flex SDK do not require any code changes to work with Sybase CEP R4. However, a recompile against the Sybase CEP R4 libraries is required. Reference the Flex SDK source included with Sybase CEP R4.
Perl SDK	Installation of Sybase CEP R4 is complete.	Applications written using the Coral8 5.6/Sybase CEP R2 do not require any code changes to work with Sybase CEP R4. However, a recompile against the Sybase CEP R4 libraries is required. Reference the module (C8.pm) included with Sybase CEP R4.

SDK	Prerequisites	Migration Instructions
Python SDK	Installation of Sybase CEP R4 is complete.	Applications written using the Coral8 5.5/Coral8 5.6/Sybase CEP R2 Python SDK do not require any code changes to work with Sybase CEP R4. However, a recompile against the Sybase CEP R4 libraries is required. Update the PYTHONPATH to refer to the Python SDK directory in the installation (for example, `C:\Program Files\SybaseC8\Server\sdk\python`).

Migrating User-Defined Functions

Prerequisites

Installation of Sybase CEP R4 is complete.

Task

To migrate user-defined functions (UDFs) written for Coral8 5.5/Coral8 5.6/Sybase CEP R2 SDK to work with Sybase CEP R4:

1. Recompile your UDF code against the Sybase CEP R4 libraries, and copy the library files over to the Sybase CEP R4 Server "bin" directory.
2. Copy the `.udf` files for your user-defined functions from the Coral8 5.5/Coral8 5.6/Sybase CEP R2 Studio “plugins” directory (for example, `C:\Program Files\Coral8\Studio\plug-ins`) over to the Sybase CEP R4 Studio “plugins” directory (for example, `C:\Program Files\SybaseC8\Studio\plug-ins`) and Sybase CEP R4 Server "plugins" directory.
3. Edit each copied `.udf` file and change the XML namespaces from `http://www.coral8.com/udf/2005/04/` to `http://www.sybase.com/udf/2005/04/`, and from `http://www.coral8.com/cpx/2005/04/` to `http://www.sybase.com/cpx/2005/04/`.

Sybase CEP Server Events

Sybase CEP Server's manager process writes most messages reflecting the Server's status to special data streams called server status streams.

Every message appears as a row in a server status stream and contains the following information:

- SourceTimestamp: The message timestamp.
- MessageGroup: The group to which the message belongs.
- ObjectID: The object identifier for the group.

CclApplicationInfo Status Messages

A description of runtime properties for projects.

The ObjectId column for this group is the full path to the project, in the following format: *workspace-name/project-name*, where *workspace-name* is the name of the workspace, and *project-name* is the name of the project.

CCLApplicationInfo messages include:

Event Name	Frequency	Description
AutoRestartAttempt	Whenever the project or the Server Container on which it is running auto-restarts.	Shows the number of times the project has auto-restarted.
CPUUtilization	Every second.	Shows the fraction of CPU the project utilizes, where a value of one (1) indicates 100% of total available processor CPU.
DroppedMessages	Every second.	Shows the total number of dropped out-of-order and late rows for the project.
InputMessages	Every second.	Shows the number of input rows the project has received. This is the sum of output rows for all external input streams.
LastError	Whenever the project generates an error.	Shows the most recent error the project has generated.

Event Name	Frequency	Description
OutputMessages	Once per second.	Shows the number of output rows the project has sent. This is the sum of input rows for all external output streams.
PendingMessages	Once per second.	Shows the number of pending rows the project's queries need to process. This is a sum of pending rows for all CCX modules in the corresponding program.
PendingPersistentMessages	Once per second.	Shows the number of rows the project needs to save to the disk. This is the sum of pending rows for all CCX modules in the corresponding program.
RunningTime	Every second.	Shows how long the project has been running.
RpcFailedMessages	Every second.	Shows the total number of failed RPC calls for the project.
RpcSentMessages	Every second.	Shows the total number of successful RPC calls for the project.
State	Whenever the state changes.	Shows a state change. State changes include: • Unregistered • Unregistering • Registered • Registering • RegisterFailed. • Started • Starting • StartFailed

CclQueryInfo Status Messages

A description of runtime message properties for specific CCL queries.

The ObjectId column for this group is the full path to the query, in the following format: *workspace-name/full-ccl-path/statement-number*, where *workspace-name* is the name of the workspace, and *full-ccl-path* and *statement-number* are a path and statement number assigned to the query by the compiler.

CCLQueryInfo messages include:

Event Name	*Frequency*	*Description*
InputMessages	Every second.	Shows the number of input rows the query has received.
OutputMessages	Every second.	Shows the number of output rows the query has sent.
RpcFailedMessages	Every five seconds.	Shows the total number of failed RPC calls for the query.
RpcSentMessages	Every five seconds.	Shows the total number of successful RPC calls for the query.

ContainerInfo Status Messages

This section describes runtime message properties for specific container processes.

The ObjectId column for this group is the container's URI, in the following format: *http://hostname:port* or *https://hostname:port*, where *hostname* and *port* are the host name and port number on which the container is running.

ContainerInfo messages include:

Event Name	**Frequency**	**Description**
ContainerAdded	Whenever a Container is added	Generates the following message: `Event: Container added (value= value)` where *`value`* is active when an active Container has been added, or passive when a passive Container has been added.

Event Name	Frequency	Description
ContainerKilled	Whenever a Container is killed	Generates the following message: `Event: Killing container (value= reason)` where `reason` is the reason the Container was killed.

ContainerRemoved	Whenever a Container is removed	Generates the following message:
		`Event: Container removed (value= reason)`
		where
		`reason`
		is the reason the Container has been removed.

CPUTime	Every second	Shows how much CPU time in microseconds the Sybase CEP Serverprocess has used since it has started.
CPUUtilization	Every second	Shows the fraction of CPU the Sybase CEP Server process is using, where a value of one (1) indicates 100% of total available processor CPU.
LogErrorCount	Every second	Shows the total number of errors that have been logged in the Server log for a given Container. This number is reset upon Container restart.
LogWarningCount	Every second	Shows the total number of warnings that have been logged in the Server log for a given Container. This number is reset upon Container restart.
TotalMemory	Every second	Shows the total bytes of memory available to Sybase CEP Server.
UsedMemory	Every second	Shows the total bytes of memory used by Sybase CEP Server.

ManagerInfo Status Messages

This section describes runtime message properties for specific manager processes.

Messages are generated on the primary active manager and any passive managers.

The ObjectId column for this group is the manager's URI, in the following format: *http://hostname:port* or *https://hostname:port*, where *hostname* and *port* are the host name and port number on which the manager is running.

managerInfo messages include:

Event Name	Frequency	Description
managerHA PromotedToPrimary	Whenever a new manager is promoted to primary status.	Generates the following message: `Event: manager HA node promoted to primary (value=` *`reason)`* where `reason` is the reason for the promotion.
managerHA DemotedToBackup	Whenever a manager is demoted to backup status.	Generates the following message: `Event: manager HA node demoted to backup (value=` *`reason`*`)` where `reason` is the reason for the demotion.
managerHA ParticipatingInElection	Whenever a manager participates in the election of a new primary manager.	Generates the following message: `Event: manager HA participating in primary elections (value=` *`reason`* where `reason` is the reason for the participation.

Sybase CEP Engine Third-Party Software Dependencies

A description of third-party software supported by the current release of Sybase CEP Engine on various platforms.

Operating Systems

This release of Sybase CEP Engine was built to support the following version of each operating system:

Operating System	Version
Microsoft Windows (32-bit)	Microsoft Windows Server 2003 SP2 Microsoft Windows XP SP2 Microsoft Windows Vista SP1
Microsoft Windows (64-bit)	Microsoft Windows Server 2003 SP2
RedHat Enterprise Linux 4 (32-bit)	RedHat Enterprise Linux ES release 4 (Nahant Update 6)
RedHat Enterprise Linux 4 (64-bit)	RedHat Enterprise Linux ES release 4 (Nahant Update 6)
RedHat Enterprise Linux 5 (32-bit)	RedHat enterprise Linux Server release 5.2 (Tikanga)
RedHat Enterprise Linux 5 (64-bit)	RedHat enterprise Linux Server release 5.2 (Tikanga)
SUSE Linux Enterprise Server 10 (64-bit)	SUSE Linux Enterprise Server 10 (x86_64) Patch level 1
Sun Solaris 10 for x86 (64-bit)	Sun Solaris 10 v5.10
Sun Solaris 10 for SPARC (64-bit)	Sun Solaris 10 v5.10

Note: Future updates or patches to this release of Sybase CEP Engine will be built against the then-current version of each operating system, which may include service packs or updates later than those listed here.

Microsoft Windows (32-bit)

	Supported Version
Languages	
C/C++	Microsoft Visual Studio 2005 SP1
Java	Sun Java5 1.5.0.15
.NET Languages	.NET 3.5 or later
Perl	5.8.8
Python	2.5.1
Databases	
IBM DB2	9.1
MaxDB	7.6
Microsoft SQL Server	Built in Driver, Native Client 2005
MySQL	3.51.11
Oracle Database	Instant Client 10.1.0.3
PostgreSQL	8.0.1
Sybase ASE	15
Sybase IQ	Untested
TimesTen	7.0.5
Message Buses	
None	
Microsoft MQ	The version installed with Microsoft Windows
Tibco Rendezvous	7.5.4
Reuters RFA	6.0.2 E2

Microsoft Windows (64-bit)

	Supported Version
Languages	
C/C++	Microsoft Visual Studio 2005 SP1

	Supported Version
Java	Sun Java5 1.5.0.15
.NET Languages	.NET 3.5 or later
Perl	5.8.8
Python	2.5.1
Databases	
IBM DB2	Unsupported
MaxDB	Unsupported
Microsoft SQL Server	Built-in driver, Native Client 2005
MySQL	Unsupported
Oracle Database	Instant Client 10.1.0.3
PostgreSQL	Unsupported
Sybase ASE	Unsupported
Sybase IQ	Untested
TimesTen	Untested
Message Buses	
None	

Red Hat Enterprise Linux 4 (32-bit)

	Supported Version
Languages	
C/C++	gcc 3.4.6-9
Java	Sun Java5 1.5.0.15
Perl	5.8.5
Python	2.3.4
Databases	
IBM DB2	9.1
MaxDB	7.6

	Supported Version
Microsoft SQL Server	FreeTDS 0.64
MySQL	2.50.39-21.RHEL4.1
Oracle Database	Instant Client 10.1.0.3
PostgreSQL	7.3-8.RHEL4.1
Sybase ASE	15
Sybase IQ	12.7
TimesTen	Untested
Message Buses	
None	

Red Hat Enterprise Linux 4 (64-bit)

	Supported Version
Languages	
C/C++	gcc 3.4.6-9
Java	Sun Java5 1.5.0.15
Perl	5.8.5
Python	2.3.4
Databases	
IBM DB2	9.1
MaxDB	Unsupported
Microsoft SQL Server	FreeTDS 0.64
MySQL	3.51.14
Oracle Database	Instant Client 10.1.0.3
PostgreSQL	7.3-8.RHEL4.1
Sybase ASE	Unsupported
Sybase IQ	12.7
TimesTen	Untested

	Supported Version
Message Buses	
Sybase RAP - The Trading Edition	R2

Red Hat Enterprise Linux 5 (32-bit)

	Supported Version
Languages	
C/C++	gcc 4.1.2-42
Java	Sun Java5 1.5.0.15
Perl	5.8.8
Python	2.5.1
Databases	
IBM DB2	9.1
MaxDB	7.6
Microsoft SQL Server	FreeTDS 0.64
MySQL	2.50.39-21.RHEL4.1
Oracle Database	Instant Client 10.1.0.3
PostgreSQL	7.3-8.RHEL4.1
Sybase ASE	15
Sybase IQ	Untested
TimesTen	Untested
Message Buses	
None	

Red Hat Enterprise Linux 5 (64-bit)

	Supported Version
Languages	
C/C++	gcc 4.1.2-42
Java	Sun Java5 1.5.0.15

	Supported Version
Perl	5.8.5
Python	2.5.2
Databases	
IBM DB2	9.1
MaxDB	Unsupported
Microsoft SQL Server	FreeTDS 0.64
MySQL	3.51.14
Oracle Database	Instant Client 10.1.0.3
PostgreSQL	7.3-8.RHEL4.1
Sybase ASE	Unsupported
Sybase IQ	Untested
TimesTen	Untested
Message Buses	
None	

SUSE Linux Enterprise Server 10 (64-bit)

	Supported Version
Languages	
C/C++	gcc 4.1.2
Java	Sun Java5 1.5.0.15
Perl	5.8.8
Python	2.5.1
Databases	
IBM DB2	9.1
MaxDB	Unsupported
Microsoft SQL Server	FreeTDS 0.64
MySQL	3.51.21-0

	Supported Version
Oracle Database	Instant Client 10.1.0.3
PostgreSQL	08.01.0102-13.2
Sybase ASE	Unsupported
Sybase IQ	Untested
TimesTen	Untested
Message Buses	
None	

Sun Solaris 10 for x86 (64-bit)

	Supported Version
Languages	
C/C++	gcc 3.4.3
Java	Sun Java5 1.5.0.15
Perl	5.8.8
Python	2.4.4
Databases	
IBM DB2	8.1
MaxDB	Unsupported
Microsoft SQL Server	FreeTDS 0.64
MySQL	3.51.14
Oracle Database	Instant Client 10.2.0.1
PostgreSQL	8.2.3
Sybase ASE	Unsupported
Sybase IQ	Untested
TimesTen	Untested
Message Buses	
None	

Sun Solaris 10 for SPARC (64-bit)

	Supported Version
Languages	
C/C++	gcc 3.4.6
Java	Sun Java5 1.5.0.15
Perl	5.8.8
Databases	
IBM DB2	Unsupported
MaxDB	Unsupported
Microsoft SQL Server	FreeTDS 0.64
MySQL	3.51.14
Oracle Database	Full Client 10.2.0.3
PostgreSQL	8.2.3
Sybase ASE	Unsupported
Sybase IQ	Untested
TimesTen	Untested
Message Buses	
SybaseRAP - The Trading Edition	R2

Server Software

Microsoft Windows Server 2003 SP1 32-bit

	Supported Version
Databases	
IBM DB2	9.1
MaxDB	7.6
Microsoft SQL Server	Server 2005
MySQL	N/A
Oracle Database	10gR2

	Supported Version
PostgreSQL	N/A
Sybase ASE	15
Sybase IQ	N/A
TimesTen	7.05
Message Buses	
None	
LDAP Servers	
Fedora/RedHat Directory Server	N/A
Microsoft ADAM	The version installed with Microsoft Windows
OpenLDAP	2.2.13-7.4E

Red Hat Enterprise Linux AS/ES 4.0 Update 4 32-bit

	Supported Version
Databases	
IBM DB2	N/A
MaxDB	N/A
Microsoft SQL Server	N/A
MySQL	4.1.20-1.RHEL4.1
Oracle Database	10gR2
PostgreSQL	7.4.16-1.RHEL4.1 / 7.3-8.RHEL4.1
Sybase ASE	N/A
Sybase IQ	N/A
Message Buses	
None	
LDAP Servers	
Fedora/RedHat Directory Server	1.04

Red Hat Enterprise Linux AS/ES 4.0 Update 4 64-bit

	Supported Version
Databases	
IBM DB2	N/A
Microsoft SQL Server	N/A
MySQL	N/A
Oracle Database	N/A
PostgreSQL	N/A
Sybase IQ	12.7
Message Buses	
Sybase RAP - The Trading Edition	R2

Index