SYBASE®
An SAP® Company

Release Bulletin

Sybase CEP Option R4

DOCUMENT ID: DC01028-01-0400-05
LAST REVISED: November 2010

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

Contents

Accessing Current Release Bulletin Information

A more recent version of this release bulletin may be available on the Web. To check for critical product or document information added after the product release, use the Sybase® Product Manuals Web site.

To access release bulletins at the Sybase Product Manuals Web site:

1. Go to Product Manuals at *http://www.sybase.com/support/manuals/.*
2. Select a product and language and click Go.
3. Select a product version from the Document Set list.
4. Select the Release Bulletins link.
5. From the list of individual documents, select the link to the release bulletin for your platform. You can either download the PDF version or browse the document online.

Product Summary

Enclosed is Sybase CEP Option R4, which is compatible with several platform and operating system configurations.

Hardware	**O/S**
Linux-64 (AMD/Intel)	Red Hat 5.0 (AMD) Red Hat 5.0 (Intel) SUSE 10 (AMD)
Sun-64 (Sparc)	Sun Solaris 10
Sun-64(UltraSparc T2)	Sun Solaris 10
Sun-64 (AMD)	Sun Solaris 10
Windows (32,64) (Server/Adapters)	Windows 2003 Server (64-bit) XP Professional (32-bit and 64-bit)
Windows (32-bit) (Studio only)	Windows XP Professional Vista

The CEP package contains:

- Sybase CEP Studio – The graphical environment for defining streams, adapters, CCL queries and CCL modules, and for managing CCL projects at runtime.
- Sybase CEP Server – The software engine that actually processes and correlates data streams at runtime.
- Sybase CEP Input and Output Adapters – The components that translate different kinds of data to and from the format used by the server.

For CEP, customers receive three CDs. The first contains the ordered platform including Sybase CEP Studio, Sybase CEP Server, and Sybase CEP adapters, and the second contains the Sybase CEP FIX Adapter and its components. The final CD is for a Windows platform and only contains Studio.

New and Changed Functionality

Learn about the new functions and features of Sybase CEP R4 ESD#1 and Sybase CEP R4.

New and Changed Functionality for CEP R4 ESD#1

Learn about the new functions and features of Sybase CEP R4 ESD#1.

Added support for using the Excel Adapter with SSL-enabled CEP servers.

Added the ability to disable auto-compilation of projects in the Sybase CEP Studio by adding the following preference to `studio-preferences.xml`:

```
<preference name="C8/Studio/DisableAutoCcx value="true"/>
```

Enhanced the CSV Adapter to enable you to:

- append the time of the last row in the output file to the filename (using the **TimestampInFilenames** parameter)
- rotate files at predefined time intervals (using the **TimebasedRotate** parameter)
- specify the time of the first rotation (using the **TimeBasedRotateStartAt** parameter)
- specify how long to wait between file rotations (using the **TimeBasedRotateInterval** parameter)

Refer to the ADL file, `c8_write_to_csv_file.adl` for details.

For the RPC Plug-in for CSV files, added the ability to suppress runtime errors and continue processing when the referenced file does not exist. In the ReadCsvFile service section of the `c8-service.xml` file, set the **AllowNoFileExist** parameter to true. It is set to false by default.

Embedded Java Support

This feature provides the ability to write CEP plug-ins in Java.

This feature provides a Java Software Development Kit (SDK) for building CEP plug-ins: input and output in-process adapters (in both regular and guaranteed delivery mode), and remote procedure calls (RPCs). In-process adapters run within the Complex Event Processing (CEP) server process. Due to the communications overhead involved with out of process adapters, in-process adapters tend to have better performance.

In-process adapters and RPCs for Sybase CEP can be written in C/C++ or Java. The Java SDK and the C/C++ SDK are functionally equivalent. Thus, Java and C/C++ in-process adapters have functionally identical run-time behavior. But the C/C++ adapters may have slightly better performance.

This feature requires Java 1.5 or higher. The Sybase CEP package does not include a Java Runtime Environment (JRE). You must install an appropriate Java package on the same

machine as the Sybase CEP server to use the Embedded Java Support. After installing Java, you will need to make the Java Virtual Machine (JVM) shared library available. On Windows, add the directory containing the `jvm.dll` file to the *PATH* environment variable. On Linux or Solaris, add the directory containing the file to the *LD_LIBRARY_PATH* environment variable. If your environment uses the *LD_LIBRARY_PATH_64* environment variable, add the directory containing the `jvm.so` file to it.

The API for the Java SDK is described completely in its accompanying Javadoc.

When ESD#1 is applied to an existing CEP R4 installation, it puts a new`c8-server.conf.src` file in the `CEP_INSTALL/SybaseC8/server/conf` directory. This file includes a new section (that looks like the following) to enable the JVM for the CEP server.

```
<section name="Java">
    <!-- The flag control whether JVM will be enabled or not, default
value is false
       -->
     <preference name="JVMEnabled" value="false"/>

     <!-- The flag control whether Java adapter will be enabled or
not, default value
          is true
       -->
     <preference name="JavaAdapterEnabled" value="true"/>

     <!-- The flag control whether Java rpc will be enabled or not,
default value
          is true
       -->
     <preference name="JavaRpcEnabled" value="true"/>

     <!-- If ignoreUnrecognized is true, CreateJavaVM ignore all
unrecognized
          option strings that begin with "-X" or "_".  If
ignoreUnrecognized is false,
          CreateJavaVM returns Error as soon as it encounters any
unrecognized
          option strings; default value is true
      -->
     <preference name="ignoreUnrecognized" value="true"/>

    <!-- configure all vm parameters separated by " " except system
property which
          begin with "-D"
       -->
     <preference name="VMArgs" value="-verbose:jni"/>

     <! This subsection is used to configure the system property -->
     <section name="SystemProperty">
         <!-- The list of semicolon(;) separated paths which can be
used to search
              java classes
```

```
            -->
          <preference name="java.class.path"
           value="CONFIG_INSTALL_FOLDER/sdk/java5/lib/
log4j-1.2.15.jar;
           CONFIG_INSTALL_FOLDER/sdk/java5/lib/jaxrpc.jar;
           CONFIG_INSTALL_FOLDER/sdk/java5/c8-sdk-java5.jar;"/>
     </section>
</section>
```

To use the Embedded Java feature, copy this section of `c8-server.conf.src` file to the same spot in the `c8-server.conf` file (located in the same directory). Edit the `c8-server.conf` file:

- change the value of the JVMEnabled preference to true
- change the string CONFIG_INSTALL_FOLDER to the folder in which CEP is installed

When ESD#1 is installed, it puts a new `c8-service.xml` file, which includes an example configuration of a Java RPC plug-in, in the `CEP_INSTALL/SybaseC8/server/conf` directory. The example looks like the following code fragment:

```
<!-- sample service configuration for java rpc
     Please don't change the value of the parameters RpcLibrary,
RpcInitCallback,
     RpcExecuteCallback, RpcShutdownCallback
    Please change the value of parameter "JavaRpcClass" to your java
rpc's class name
    Please change the value of parameter "ClassPath" to the path that
contains your
     rpc class, if it already contained in class path configured in
c8-server.conf,
     you can simply remove this parameter
  -->
  <!--  <Service Name="TestJavaRpc" Type="REMOTESERVICE">
    <Description>Sample service configuration for Java Rpc</
Description>
    <Param Name="RpcLibrary">c8_java_rpc_lib</Param>
    <Param Name="RpcInitCallback">c8_java_rpc_initialize</Param>
    <Param Name="RpcExecuteCallback">c8_java_rpc_execute</Param>
    <Param Name="RpcShutdownCallback">c8_java_rpc_shutdown</Param>

    <Param Name="JavaRpcClass">com.sybase.c8.rpc.TestRpc</Param>
    <Param Name="ClassPath"></Param>
  </Service>
  -->
```

An example ADL file for a Java in-process adapter, `c8_java_adapter_demo.adl`, is located in the `CEP_INSTALL/SybaseC8/server/plugin` directory.

User developed RPC plug-ins or in-process adapters can be a set of `.class` files or a single `.jar` file. A plug-in or adapter can have dependencies on other `.jar` files. Locations of all the files must be added to the classpath. Multiple locations are separated by semicolons on Windows and by colons on Linux and Solaris. The classpath to find the file(s) for user-

developed Java RPC plug-ins and in-process adapters can be set using any of the following methods:

- changing the value of the java.class.path preference in the `c8-server.conf` file
- setting the"ClassPath" parameter for the Java RPC plug-in in the `c8-services.xml` file
- changing the "ClassPath" parameter's default value in the in-process adapter's ADL file

These methods of setting the classpath are provided because this feature does not use the *CLASSPATH* environment variable to load the Java adapter's classes or jars.

When searching for in-process Java adapters, classpaths are read in the following order:

- First, the classpath specified by the ClassPath parameter in the in-process adapter's ADL file is read.
- If nothing is found, the classpath specified by the java.class.path preference in the `c8-server.conf` file is read.
- If nothing is found, "ClassNotFoundException" exception is thrown.

When searching for a Java RPC, classpaths are read in the following order:

- First, the classpath specified by the ClassPath parameter in the `c8-services.xml`file is read.
- If nothing is found, the classpath specified by the java.class.path preference ins the `c8-server.conf` file is read.
- If nothing is found, a "ClassNotFoundException" exception is thrown.

ESD#1 includes the following additional example files:

- `c8-in-process-adapters.jar` is a binary of the example Java in-process adapter.
- `c8-in-process-adapters-src.jar` contains the source files for the example Java in-process adapter (unzip before using).
- `c8-rpcs.jar` is a binary of the example Java RPC plug-in.
- `c8-rpcs-src.jar` contains the source files for the example Java RPC plug-in (unzip before using).

Kdb+ Driver

The kdb+ driver allows you to read information from and write information to a kdb+ database server.

Kdb+ is an in-memory and on-disk database produced by KX Systems. Kdb+ databases can be accessed using the proprietary q language. You may configure the kdb+ database as a "service" to access it from Sybase CEP Engine.

Sybase CEP provides a special-purpose kdb+ driver using KX Systems' proprietary APIs. You may get data in and out of kdb+ using either:

- A database subquery, or
- The Database Statement

Note: The Read from Database Adapter, the Poll from Database Adapter, and the Write to Database Adapter do not work with the kdb+ driver. CCL database subqueries are the preferred way to retrieve data from kdb+ into Sybase CEP Engine.

Configuring Connections to Databases with User Name, Password, and Database Source Name

Configure CCL queries to interact with a kdb+ database using a database user name, password, and Database Source Name (DSN).

To set up a connection from within Sybase CEP Engine.

1. In the `c8-services.xml` file, create a separate service entry section for every kdb+ database you wish to connect to. This entry provides information to enable Sybase CEP Server to communicate with the kdb+ database server. To connect to multiple kdb+ database servers or to multiple kdb+ databases on a single server, create a separate section for each kdb+ 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 customize one of these sections, or make a copy of it 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, and may contain a character string consisting of letters, numbers, underscores, dots and colons. Ensure that the service name begins with a letter, and that it is the same name as the name of the database used in your CCL queries. For example, if the 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. (Optional) Add a description of your service entry in the "Description" element.
5. Set the "DBDriverType" preference to "DBDriverKDB" to connect to a kdb+ database.
6. Set the "DBDriverConnectString" preference to the information necessary to connect to the external database. For DBDriverKDB, the "DBDriverConnectString" is the name of the host name of the kdb+ server.

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

   Note: If your "DBDriverConnectString" contains a semicolon (;), forward slash (/), question mark (?), colon (:), at sign (@), ampersand (&), equal sign (=), plus sign (+),

dollar sign ($), or comma (,), the character must be preceded by a back slash escape character (\).

7. For kdb+ services, set the "Port" preference to the port number of the kdb+ server.
8. (Optional) Set the "CacheMaximumAge" preference to specify how long Sybase CEP Server keeps a cached copy of the information it reads from the kdb+ database.
9. Set the "Username" preference to the user name that is used when communicating with the kdb+ database.

   Note: 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 specified in the "Username" preference. The password is unencrypted.
11. Set any desired optional preferences for your database service entry in the `c8-services.xml` file.

Configuration Preference Example

Review an example of configuration preferences for a kdb+ driver.

Example of an entry that uses 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>
```

Ensure that the kdb+ database you wish to connect to is running properly before attempting to connect to it from Sybase CEP Engine.

Datatype Mapping Between kdb+ and Sybase CEP Engine

Q has a larger set of datatypes than CCL. Sybase CEP Engine automatically converts kdb+ datatypes to compatible CCL datatypes when data is sent from kdb+ to Sybase CEP Engine.

You may explicitly cast values stored in CCL datatypes to q datatypes.

Conversion Between CCL and Q

Learn how to convert CCL to q datatypes, q types to CCL, and q values to CCL rows.

If you wish to convert a CCL datatype to a q datatype other than the type to which it automatically converts, explicitly cast the CCL datatype as the desired q type, as described in EXECUTE STATEMENT DATABASE.

Table 1. How CCL datatypes convert to q datatypes

CCL datatype	Q datatype
BLOB	list of byte
BOOLEAN	boolean
FLOAT	float
INTEGER	int
INTERVAL	long
LONG	long
STRING	list of char
TIMESTAMP	datetime
XML	list of char

Table 2. How q datatypes convert to CCL datatypes

Q datatype	CCL datatype
boolean	BOOLEAN INTEGER
byte date int minute month time second short	INTEGER LONG Time types are set to their q integer representation.
long	INTEGER (On overflow, if value is greater than MAX_INT, the value is set to NULL.) LONG
real float	FLOAT

Q datatype	CCL datatype
char list of char symbol	STRING
datetime	TIMESTAMP
Other types	Conversion not supported

When a database subquery includes a statement using the SQL dialect of q (as opposed to regular q syntax), data is read into CCL from kdb+ as it would be from any database: each row returned by the SQL dialect of q is read into a row in Sybase CEP Engine with the same number of columns. When a database subquery includes a statement using simple q (not its SQL dialect), the results returned by the q statement are mapped as shown in the following table.

Table 3. Q value mapping to CCL rows and columns

Q	CCL
Single value (atom)	One column, one row.
Simple list	One column, multiple rows.
Dictionary	Two columns, multiple rows.
Flip	Multiple columns, multiple rows (the same as from the SQL dialect of q).

Reading Data from kdb+

Use a database subquery to read data from kdb+ into Sybase CEP Engine.

To get data from kdb+ into Sybase CEP Engine, use a database subquery. For example:

```
INSERT INTO OutStream
SELECT
    DbResult.StockSymbol,
    DbResult.Price,
    DbResult.Volume
FROM
    PulseStream,
    (DATABASE "MyKdbService"
     SCHEMA (StockSymbol STRING,
             Price FLOAT,
             Volume INTEGER)
     [[select StockSymbol, Price, Volume from trades]]
    ) as DbResult
;
```

This query tells Sybase CEP Server which database to connect to by specifying:

```
DATABASE "MyKdbService"
```

Ensure the service name, *MyKdbService* is defined in the `c8-services.xml` file.

Sybase CEP Server extracts the query

```
select StockSymbol, Price, Volume from trades
```

from the database subquery and sends this query to the kdb+ server, which executes the query and returns the requested data. Use the q language to write the query that you send to the kdb+ server. The query may use parameters.

Writing Data to kdb+

Use the CCL language's Database Statement to write data to a kdb+ database.

The Database Statement uses a mix of CCL and q syntax.

EXECUTE STATEMENT DATABASE clause

Specifies the name of an external relational database connection and the SQL or q statements that should be executed against the database.

Syntax

```
EXECUTE STATEMENT DATABASE "service" [[statements]]
   SELECT…
```

Table 4. Components

service	The name of a database service defined in the file c8-services.xml. For more information about configuring Sybase CEP Engine database services, see the *Sybase CEP Installation Guide* .
statements	The statements to be executed by the database software.

Usage

The EXECUTE STATEMENT DATABASE clause is the first clause of the Database statement. It indicates the database that is impacted when the CCL query in the Database statement generates output, and specifies the SQL or q statements that should be executed against the database in that event.

Sybase CEP Engine sends the text you specify (as statements) to the external database for execution after replacing references to CCL columns and parameters with the appropriate values. Since Sybase CEP Engine does not interpret the statements beyond replacing the column and parameter references, you may specify any statement supported by your database. However, Sybase CEP Engine ignores any return values. To retrieve data from an external database, use a database subquery.

To include the value of a CCL column in your database statement, preface the column name with a question mark (**?**stream_or_alias.column). To include the value of a CCL parameter, preface the parameter name by a question mark and a dollar sign (**?$**parameter).

ON ERROR extension

Use the ON ERROR extension to handle database errors that may cause Sybase CEP Engine to stop project execution.

The syntax for the EXECUTE STATEMENT DATABASE clause with the ON ERROR extention is:

```
EXECUTE STATEMENT DATABASE "service" [[statements]]
   ON ERROR [error_insert_clause] CONTINUE
   SELECT…
```

Additional syntax related to the error_insert_clause:

```
error_insert_clause: INSERT INTO error_stream_name error_select_list
error_stream_name: Name of an output stream or local stream.
error_select_list: SELECT {error_expression} [,...]
error_expression: A CCL expression.
```

The error_expression can only contain:

- References to columns in the stream specified in the stream_clause.
- Operators, constant literals, and scalar functions.
- CCL parameters in the project. These should be prefaced with $.
- The ERROR_MESSAGE() built in.

The ON ERROR extension is optional. The error_insert_clause for the ON ERROR extension is also optional. If the ON ERROR extension is not provided and errors occur, the behavior of this clause is determined by the setting of the IgnoreErrors property in the c8-services.xml service configuration file. By default, this property is set to 'false', which means that if an error occurs, the query is aborted.

Only the INSERT INTO statement or the EXECUTE STATEMENT DATABASE clauses are legal within the ON ERROR clause.

When the ON ERROR extension is provided without the error_insert_clause, errors found in the execution of this clause are ignored and CEP Engine continues to execute the project. Subsequent tuples arriving in any of the input streams of the from_clause continue to trigger execution of this clause. If these executions generate errors, they are also ignored by the CEP Engine.

When the ON ERROR extension is provided with the error_insert_clause, errors found in the execution of this clause are ignored and CEP Engine continues to execute the project. However a new tuple is also inserted into the errorstream (specified by error_stream_name). This tuple contains the fields selected in the error_select_list clause. The timestamp of this tuple is the same as the timestamp of the tuple in the from_clause that triggered the execution of this clause. These tuples are inserted in the errorstream in the same order as that of the original tuples in the from_clause that triggered the execution of this clause.

The ON ERROR extension cannot be used with both an EXECUTE STATEMENT DATABASE/EXECUTE REMOTE PROCEDURE clause and a REMOTE SUBQUERY/ PROCEDURE clause in the same CCL statement. This causes errors. To avoid errors, the CCL statements must be broken into separate statements using an intermediate local stream.

Set the MaxRetries parameter in the service definitions for Database 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, therefore there is no retry.

Kdb+ and q

EXECUTE STATEMENT DATABASE clauses that modify kdb+ databases can contain statements in q language.

Important: The q statement cannot contain newline characters. Everything between the brackets ([[ ]]) must be entered on a single line.

When you pass values from the Database statement's data source to the q statements in the EXECUTE STATEMENT DATABASE clause, you must enter the values as a space-separated list that appears immediately after the parameterized q statement. Sybase CEP Engine passed the values to the final q statement as a single general (multi-type) list. This list takes the place of the right side of the q statement (the part of the statement after the operator). For example, consider the following q statement:

```
5+(1;2;3)
```

To pass column values for the three parameters, you would write it like this:

```
5+ ?MyStream.IntColumn1 ?MyStream.IntColumn2 ?MyStream.IntColumn3
```

If you pass a list of parameters to the q (or Q-SQL) statement, the final statement must be written in the brackets to accommodate the parameters. For example:

```
{[params] select Price, Volume from trades where StockSym = `
$params[0],
   Volume > `$params[1]} ?ParamStream.StockSym ?ParamStream.Volume
```

Sybase CEP Engine automatically converts CCL datatypes to a limited set of q datatypes. To convert a CCL datatype to another kdb+ type, you must explicitly cast the CCL type as the

desired kdb+ type in your q statement. For example, the following Database statement casts the values from four columns to datetime, symbol, real, and short in q:

```
EXECUTE STATEMENT DATABASE "MyKDBService"
[[`trades insert (`datetime ; ` ; `real ; `short) $ ?Ts ?StockSym ?
Price ?Volume]]
SELECT GETTIMESTAMP (QueryIn) AS Ts,
       StockSym AS StockSym,
       Price AS Price,
       Volume AS Volume
FROM QueryIn;
```

Note: Specify all types for all of your parameters, even to cast a single value.

Examples

This query updates a table called MyTable in a database called MyDatabase. The Symbol and Price columns (aliased as CurSymbol and CurPrice) are selected from a Sybase CEP stream called StreamIn. The SQL query matches symbols in the Symbol column of MyTable with those of the Symbol column in StreamIn, and records the corresponding price from the Price column in StreamIn to a column called Price in MyTable.

```
EXECUTE STATEMENT DATABASE "MyDatabase"
  [[UPDATE MyTable
    SET MyTable.Price = ?CurPrice
    WHERE MyTable.Symbol=?CurSymbol]]
SELECT
  StreamIn.Symbol AS CurSymbol, StreamIn.Price AS CurPrice
FROM StreamIn;
```

This example inserts column values from the ActiveStrategies window and MyTrades streams, as well as a calculated timestamp into TradingDatabase:

```
CREATE WINDOW ActiveStrategies
SCHEMA (Symbol STRING, Strategy STRING)
KEEP LAST PER Symbol;

CREATE INPUT STREAM MyTrades
SCHEMA (Symbol STRING, Quantity INTEGER, Price FLOAT);

EXECUTE STATEMENT DATABASE "TradingDatabase"
   [[INSERT INTO Positions
      VALUES (?Symbol, ?Ts, ?Quantity, ?Price, ?Strategy) ]]
SELECT *, GETTIMESTAMP(MyTrades) as Ts
FROM MyTrades, ActiveStrategies
WHERE MyTrades.Symbol = ActiveStrategies.Symbol;
```

This example demonstrates how errors are logged when the ON ERROR extension and insert_clause are provided with this clause:

```
Example:
EXECUTE STATEMENT DATABASE "StockTradeDB"
[[
```

```
    INSERT INTO TradeTable VALUE(?Symbol, ?Price, ?Volume, ?Ts)
]]
ON ERROR
INSERT INTO DBWriteErrorStream
SELECT Symbol, Ts, ERROR_MESSAGE()
CONTINUE
SELECT
Symbol  as Symbol,
Price as Price,
Volume as Volume,
GETTIMESTAMP(InTrades) as Ts
FROM
InTrades KEEP EVERY 1 minute;
```

where the schema for DBWriteErrorStream is (Symbol String, Ts TimeStamp, ErrMsg String).

Using Q

For both the Database Statement and database subquery, you may use any q snippet that you can use on a q console.

When parameters are passed, consider it as Sybase CEP Engine appending (<param1>;<param2>;<param3>) to the end of the statement.

In addition to the q results (called "flips" in kdb+ terminology), Sybase CEP Engine can read single atomic values, simple lists, and simple dictionaries. For example, if you have a variable called *var1* in your kdb+ instance, you may read the value of that variable into Sybase CEP Engine. In this example, the database subquery is *[[var1]]*, the schema of your expected database output has a single column, and the datatype of *var1* is compatible with the datatype of the column into which the data is read.

If *var1* is a simple list, it is read the same way as a single atomic value, with each element in the list being read as a separate row. Simple dictionaries are read the same way as simple lists, but with two values per row: the key and the value. To read dictionaries, your expected schema in Sybase CEP should contain a string for the key, and a compatible column for the value. Lists of lists, and dictionaries containing lists as values, are not supported.

Sybase CEP Adapter for SL Enterprise RTView

The Sybase CEP Adapter for SL Enterprise RTView® (RTView Adapter) acts as a Sybase CEP data source for Enterprise RTView.

RTView Adapter is a java software component that is used to subscribe to Sybase CEP Server data, and deliver the data to the Enterprise RTView Dashboard for display. It also allows you to send data from the RTView Dashboard to the Sybase CEP Server.

System Requirements

Learn what components are needed to run RTView Adapter.

To use the RTView Adapter, install java 1.5 (or higher) on your client machine, along with a licensed version of SL Corp's RTView Enterprise Dashboard software version 5.1 or higher. See the *Enterprise RTView* documentation.

Installation

Learn about the files and steps required to install RTView Adapter.

Prerequisites

Prior to installing and using the RTView Adapter on your client machine, install the Enterprise RTView software from SL Corp. Also, ensure that you have access to an installed Sybase CEP Server.

The RTView Adapter consists of several jar files and some simple script files. The most notable ones are:

- `c8ds.jar`
- `c8-sdk-java5.jar`
- `start_builder.bat start_builder` (for Unix)
- `start_viewer.bat start_viewer` (for Unix)

The `c8ds.jar` and `c8-sdk-java5.jar` files are used to support communications between the RTView Adapter and the Sybase CEP Server.

To install the RTView Adapter:

Task

1. Create a new folder and unpack all the files into it.
2. Create an environment variable called *C8_DASHBOARD* and set its value to the location of the folder created in step 1.
3. Ensure that the *RTV_HOME* environment variable is set and is pointing to the 5.1 or later version of SL RTView installation.

Starting a RTView Display Builder Using the RTView Adapter

Follow steps provided to start the RTView display builder.

To start a RTView display builder using the RTView Adapter:

1. Ensure that the Sybase CEP Server that the RTView Adapter attempts to connect to, is up and running.
2. (Windows) Open a DOS command prompt and type in the following command:

%C8_DASHBOARD%\bin\start_builder.bat <project_folder> [<rtv_file_name> "<rtview options>"]

Note: Create the `<project_folder>` before starting the builder. The `<rtv_file_name>` is optional and needs to be supplied only when opening an preexisting `.rtv` file. The `<rtview_options>` are options that are passed to RTView as is. Enclose all the options in double quotes.

Note: You may choose to create a preconfigured desktop shortcut to run the command above.

3. (Linux/Unix) Type in the following command:

   $C8_DASHBOARD/bin/start_builder.bat <project_folder> [<rtv_file_name> '<rtview_options>']

   Note: Create the `<project_folder>` before starting the builder. The `<rtv_file_name>` is optional and needs to be supplied only when opening an preexisting `.rtv` file. The `<rtview_options>` are options that are passed to RTView as is. Enclose all the options in single quotes.

   Note: You may choose to create a preconfigured desktop shortcut to run the command above.

Starting a RTView Display Viewer Using the RTView Adapter

Learn how to start the RTView display viewer.

To start the RTView display viewer using the RTView Adapter:

1. Ensure that the Sybase CEP Server that the RTView Adapter attempts to connect to, is up and running.
2. (Windows) Open a DOS command prompt and type in the following command:

   %C8_DASHBOARD%\bin\start_viewer.bat <project_folder> <rtv_file_name> ["<rtview_options>"]

   Note: The `<project_folder>` and `<rtv_file_name>` must exist before starting the viewer. The `<rtview_options>` are passed to the RTView viewer as is. Enclose the options in double quotes.

   Note: You may choose to create a preconfigured desktop shortcut to run the command above.

3. (Linux/Unix) Type the following command:

   $C8_DASHBOARD/bin/start_viewer <project_folder> <rtv_file_name> ['<rtview_options>']

   Note: The `<project_folder>` and `<rtv_file_name>` must exist before starting the viewer. The `<rtview_options>` are passed to the RTView viewer as is. Enclose the options in single quotes.

Note: You may choose to create a preconfigured desktop shortcut to run the command above.

Creating a Connection to the Sybase CEP Server

Learn how to create a connection to the Sybase CEP Server.

You may create one or more connections to a single instance of the Sybase CEP Server or multiple instances of the Sybase CEP Server.

To create a connection to the Sybase CEP Server in the RTView builder:

1. Select the **Tools > Options** menu item in the Display Builder.
2. In the window that opens up, select **C8** from the available options on the left side.
3. In the C8 Connections tab, click **Add**.
4. Fill in the appropriate connection information and click **OK**.
5. Click **Apply** and **Save** to save the connection information in the configuration file. When you click **Save**, a dialog box appears confirming whether you want to save the connection to the RTView initialization file or the intialization file in the folder that you started from (for example. the folder name you provided to `start_builder` or `start_viewer` script. Sybase recommends that you click **No** to save the connection information in the folder you started from.

Updating an Existing Connection to the Sybase CEP Server

Follow steps to edit an existing connection to the RTView Adapter.

To update an existing connection to the Sybase CEP Server:

1. Click on the **Tools > Options** menu item in the Display Builder.
2. In the window that opens up, choose **C8** from the available options on the left side.
3. In the C8 Connections tab, double-click the connection you wish to edit. The Add C8 Connection window appears.
4. Fill in the appropriate connection information and click **OK**.
5. Click **Apply** and **Save** to save the connection information in the configuration file.
6. When you click **Save**, a dialog box appears confirming whether you want to save the connection to the RTView initialization file or the intialization file in the folder that you started from (for example. the folder name you provided to `start_builder` or `start_viewer` script. Sybase recommends that you click **No** to save the connection information in the folder you started from.

Important: Once you have edited the connection, restart the RTView Builder or restart the Sybase CEP Server for the changes to take effect.

Subscribing to the Sybase CEP Server

To view streaming data from the Server, define a cache and use it as the data source to the table.

You may specify certain properties in the cache. These properties are necessary for viewing updates and deletes to the data:

- **valueTable –** Table containing the inserts and updates for the Stream you wish to view. To set this property, right-click on the `valueTable` property of the table object, and select **Attach To Data > C8**. A window appears where you may choose a predefined connection. Next, choose the stream and columns in the Streams you wish to view. You may choose the columns you wish to include in the Cache as long as the index columns specified have been selected.
- **indexColumNames –** Use this to specifiy the key columns for the Streams. You may specify this by separating the column names by a semicolon. For example, KeyCol1;KeyCol2. Note that the column names are case sensitive.
- **deleteTable –** Table that contains the delete records. You may follow the steps above to set the `valueTable` property, except for choosing the Stream name. Since the Server does not have a specific delete table, a virtual table name is used. The virtual name is derived by prefixing the Stream name chosen for the valueTable property with a "!". For example, if the stream name chosen is 'Positions', then the deleteTable will be '!Positions'. Note that there is no such stream in the Server, so type in the StreamName rather than choosing it from the drop down list.
- **maxNumberOfRows –** This value is 0 by default and it means that there is no history of records maintained for an index. Therefore, only the most recently arrived row for an index is displayed. This is sufficient for objects that display only the most recent data such as Tables, and Pie Charts, but for having a history may be necessary for objects that display trends.

  Setting this value to anything greater than 0 enables the history mechanism. For every key value, there will be N rows of history maintained, where N is the number of rows specified for this option.

The cache needs to be created in a separate rtv file and this file be imported into the main rtv file by selecting **Tools > Options > Caches selection**. Also, note that if any changes are made to the cache after it has been imported into the main display file, the cache needs to be refreshed again by selecting **Tools > Options > Refresh Selection**.

To use the cache in a table object:

1. Right click on the `valueTable` property of the table object.
2. Select **Attach To Data > Cache**.
3. A new window displays. Choose the cache name.
4. For the Stream, choose 'Current'.

5. Next, choose the columns that need to be displayed.

Note: There is an additional column available in the cache, called the time_stamp column. Displaying this column is optional.

Publishing to the Sybase CEP Server

To set up publishing, associate a custom command with a Control Object such as a Button in the Display Builder.

The RTView Adapter supports publishing data to the Sybase CEP Server from a dashboard display. To set up publishing:

1. Right-click the `actionCommand` property of the control, under the 'Interaction' category.
2. Select **Custom Command** > **C8**.
3. A new window appears. Type in a publish command using the following syntax:

   <conn_name>.publish ## <workspace_name.project_name.stream_name> ## <col_value_1> [## <col_value_2> ...]

   where

   - **## –** Is the argument delimiter. You may configure this to be anything using the **Add C8 Connection** box. To get to this box, select **Tools** > **Options** in the Builder application, then select **C8** from the list of options in the left, and click **Add**. Note that a space before and after the delimiter is required or the command does not work as expected.
   - **<conn_name> –** Is the name of a predefined connection to use for publishing.
   - **<workspace_name.project_name.stream_name> –** Is the target stream and it is composed of the concatentation of the workspace, project and stream name.
   - **<col_value_...> –** Is the value to specify for each column. There must be as many column values as there are columns in the target Stream. If you need to specify an empty string, change the `Null Value` property in the Add C8 Connection window to a different value. For any type other than Strings, an empty value (for example, `''`) always represents a NULL.

     To specify date and timestamp values, use the same format as the one specified in the `Date Format` and `Timestamp Format` property in the Add C8 Connection window. The format specification is the same as the one used by the Java SimpleDateFormat object.

     Variables, which are bound to appropriate controls, such as text boxes, radio buttons, and combo boxes, are typically used to supply the column values.
4. If the publish command is successful, there is no response. If there is an error, a message box appears with the appropriate error message.

Running the Provided RTViewer Adapter Example

Learn how to run the example provided with RTView Adapter.

To run the provided example:

1. Start Sybase CEP Server with the provided PortfolioValuation project, and use port 6789.
2. (Windows) To start the sample display in the RTView viewer, open a DOS command prompt and use:

   %C8_DASHBOARD%\bin\start_viewer.bat %C8_DASHBOARD%\examples examples.rtv

   Note: To start the example RTView builder, replace the **start_viewer** command with the **start_builder** command.
3. (Linux/Unix) Type:

   $C8_DASHBOARD/bin/start_viewer $C8_DASHBOARD/examples examples.rtv

   Note: To start the example RTView builder, replace the **start_viewer** command with the **start_builder** command.
4. Follow the on-screen instructions.

Known Issues and Limitations

Learn about the known issue and apply workarounds for RTView Adapter.

Table 5. Known issues for RTView adapter

CR#	Description
N/A	**Modifcations to existing connections to the Sybase CEP Server are not saved automatically.** **Workaround:** Restart the builder or the Sybase CEP Server for the changes to take effect.
N/A	**When subscribing to a stream with dates, the dates will show up in local time instead of GMT. Therefore, the dates may appear different than what is stored in the Server, depending on the timezone you are in.** **Workaround:** Ensure that the timezone option is specified as GMT when running the **run_builder** or **run_viewer** command. If the `start_builder` and `start_viewer` scripts are used, this is not an issue.
N/A	**The project and subscribe option is not currently supported.** **Workaround:** All other options are supported.

CR#	Description
N/A	**All error messages are displayed in the console only, except for the publication related messages.** **Workaround:** N/A.
N/A	**In the Add C8 Connection window, if OK is clicked before entering the connection name, an empty popup box shows up. This is a RTView issue and needs to be resolved by SL.** **Workaround:** N/A.
N/A	**The adapter does not currently support displaying null columns from a stream subscription or publishing data with null fields to a stream.** **Workaround:** Some users may work around this issue by having an additional stream translate the null columns for each tuple into a special value (for example, convert null integers into a -1) to indicate null, and have their RTView project subscribe to this stream instead and display accordingly.
N/A	**The adapter does not currently support displaying BLOB datatype. Publishing blob data is also not supported.** **Workaround:** The adapter always displays <<Binary Data>>.
N/A	**The time portion of a timestamp field is not recognized properly during publishing, even when the corresponding format is specified in `C8OPTION.ini`.** **Workaround:** N/A.

New and Changed Functionality for CEP R4

Learn about the new functions and features of Sybase CEP R4.

Analytic Functions

Learn about the addition of new SQL functions.

To support the analysis of large volumes of time-series data stored as rows within a table, Sybase CEP provides a new set of analysis functions and capabilities:

- ANSI Standard Analytic Functions
- Industry Standard Analytic Functions
- Weighted Aggregation Functions
- Enhanced Date Time Functions

These functions are implemented as aggregate or accumulate functions in CCL; they are not implemented as UDFs. Functions that require numeric data types can accept integer, long, float, timestamp, and interval datatypes.

Installer Changes

Sybase CEP allows for silent, GUI, and console installs.

Silent installs are recommended for installing multiple identical or almost identical copies of Sybase CEP. Silent installs are not recommended for installations that require excessive customization.

Sybase CEP now provides a wizard-based installer for in-process and out-of-process adapters. For out-of-process adapters, there are two options: standalone and dependent mode. In standalone mode, the adapter installer installs the Sybase CEP common libraries that are needed by the adapter. This mode of installation is used when installing the adapter to a system that doesn't have CEP Server or Studio installed.

Dependent mode can be used if the system has Sybase CEP Server or Sybase CEP Studio installed. In this case, the adapter installer does not install the Sybase CEP common libraries and the adapter uses the common libraries that are installed as part of CEP Server or Studio. Dependent mode requires Sybase CEP Server or Sybase CEP Studio to be of the same version as the adapter being installed. If an incompatible version of Sybase CEP Server or Sybase CEP Studio exists on your workstation, use standalone mode to install the adapter.

Note: Some adapters, such as Tibco, WSMQ, and MSMQ, do not provide a wizard-based installer. Consult your adapter manual for a full list of features.

For information on installation and configuration, see the *Sybase CEP Installation Guide*. For information on adapter installation, see the *Sybase CEP Adapter Installation Guide*

FIX Adapter

The new Sybase CEP FIX Adapter allows the Sybase CEP Server to send and receive FIX messages.

Sybase CEP FIX Adapter is an out-of-process C/C++ adapter that facilitates the direct exchange of FIX messages with a counter party FIX server. The adapter also enables writing and reading of FIX messages to/from a file or database.

Sybase CEP FIX Adapter can function as an input or an output adapter, and can perform both functions simultaneously in the same adapter instance.

Sybase CEP FIX Adapter is licensed separately from the Sybase CEP Server and Sybase CEP Studio. For information on FIX adapter configuration and customization, see the *Sybase CEP FIX Adapter Guide*.

RAP Adapter Enhancements and Updates

This section discusses the increased functionality of the RAP - The Trading Edition® adapter.

RAP adapter is now available as both an in-process and out-of-process adapter. The in-process adapter is only available as an output adapter.

The following improvements have been made:

- Sybase CEP R4 is compatible with multiple versions of RAP. To configure Sybase CEP for the RAP R4 adapter, run the `c8-use-rap-r4.sh` script.
- The RAP adapter can automatically convert Sybase CEP datatypes to RAP datatypes.
- The RAP out-of-process adapter is compatible with UAF.

The current instructions in *Sybase CEP Integration Guide> Sybase CEP Adapters> Adapters Supplied by Sybase CEP> RAP - The Trading Edition Adapters> Setting up UA for RAP - The Trading Edition Adapters* are no longer applicable. Use the following instructions instead:

Unified Agent (UA) is a Java based agent that gathers statistics from a RAP publisher. It uses JMX technology and runs parallel to the server it monitors or administers.

Sybase recommends that you copy the `bin/rap/ua` directory and use a different `ua` directory for each running instance of a RAP adapter.

To start the UA agent and configure the RAP in-process or out-of-process adapter to package UA statistics:

1. In `bin/rap/ua/bin/myenv`, set appropriate values for the following envars as specified in the "TODO" instructions at the top of the file:
   - SYBASE_UA
   - SYBASE_JRE
   - UA_RMI_PORT

   Note: UA_RMI_PORT in myenv should match the port property value in `service-config.xml`.
2. In `bin/rap/ua/plugins/com.sybase.rap.rapservice/agent-plugin.xml`, set appropriate values for the following properties as specified in the file:
   - rap.service_host
   - rap.service_port
   - rap.service_startCommand
   - rap.service_pluginsubtype
   - rap.service_plugindisplayname

   If the `bin/rap` directory link points to the `rap-r4` directory, set the following properties in `agent-plugin.xml` also:
   - rap.service_logFile
   - rap.service_configFileDir

- rap.service_componentTemplateDir

3. In `/rap/ua/services/RMI/service-config.xml`, set the appropriate value for the following property as specified in the file:
   - port

Note: The AdminPort node value in the adapter `publisher.xml` should match the rap.service_port property value in `agent-plugin.xml`.

For more information on configuring and using the RAP adapter, see *Sybase CEP Integration Guide> Sybase CEP Adapters> Adapters Supplied by Sybase CEP> RAP - The Trading Edition Adapters*.

Guaranteed Delivery for WebSphere and MSMQ Adapters

The WebSphere and MSMQ adapters now support guaranteed delivery of messages within Sybase CEP.

Guaranteed delivery ensures that every Sybase CEP row is received by its destination at least once, as long as the software components are running.

For more information on implementing Guaranteed Delivery for your adapter, see the *Sybase CEP Integration Guide* and the appropriate adapter guide.

Infiniband Support

Sybase CEP now provides communication support using Infiniband.

Infiniband is a high-performance, low-latency network that communicates using Socket Direct Protocol (SDP). This allows you to listen to SDP ports and addresses.

A new flag in the `c8-server.conf` file, named EnableAutoSDP, is set to enable SDP connection based on the hostname, ListenTo, and ManagerUri settings.

For more information on configuring Sybase CEP for an Infiniband environment, see the *Sybase CEP Integration Guide* .

Database Statement and RPC Error Handling

Sybase CEP R4 uses new syntax to improve error handling when executing database procedures and RPCs.

Four separate clauses within CCL allow control to escape to a database or remote procedure:

- EXECUTE STATEMENT DATABASE
- EXECUTE REMOTE PROCEDURE
- DATABASE Subquery
- REMOTE Subquery

By default, the Sybase CEP Server treats some errors returned by a database or RPC as fatal errors, which halts the execution of the Sybase CEP Server project. This behavior can be

overridden by setting the IgnoreErrors preference to TRUE in the `c8-services.xml` configuration file.

Additionally, a new parameter is allowed in the service definition for Database and RemoteService types. This parameter allows for automatic retry of the remote execution in case of an error.

For full documentation of the new syntax and its functionality, see the *Sybase CEP Integration Guide* .

Data Type Conversions

Sybase CEP is able to perform datatype conversions.

Known Issues

Learn about known issues and apply workarounds for Sybase CEP.

Known Issues for Sybase CEP Installation

Learn about known issues and apply workarounds for Sybase CEP installation.

Table 6. Sybase CEP installation issues

CR#	Description
N/A	**Launching `setup.exe` directly from the winzip file causes the installation to hang.** **Workaround:** Due to dependencies between the Sybase CEP installation files, extract the contents of the zip file `cepr4_winx86_32.zip` to a temporary location before launching `setup.exe`.
N/A	**Some Sybase applications and utilities are unable to support spaces in their directory structures, so using spaces can result in errors.** **Workaround:** When installing , avoid using spaces in your destination directory.

Known Issues for Sybase CEP Studio

Learn about known issues and apply workarounds for Sybase CEP Studio.

Table 7. Sybase CEP studio issues

CR#	Description

592594	**When starting on a Solaris platform with default locale settings (LC_ALL=C), an error may occur.**
	Workaround: Set the locale to something compatible with GTK:
	`% setenv LC_ALL en_US.ISO8859-1`
	`% locale`
	`LANG=`
	`LC_CTYPE="en_US.ISO8859-1"`
	`LC_NUMERIC="en_US.ISO8859-1"`
	`LC_TIME="en_US.ISO8859-1"`
	`LC_COLLATE="en_US.ISO8859-1"`
	`LC_MONETARY="en_US.ISO8859-1"`
	`LC_MESSAGES="en_US.ISO8859-1"`
	`LC_ALL=en_US.ISO8859-1`

Known Issues for FIX Adapter

Learn about known issues and apply workarounds for FIX adapter.

Table 8. FIX adapter issues

CR#	Description

N/A	**Due to a known issue with the console installer, installing the FIX Adapter in dependent mode on a Solaris x86 platform causes the installation to hang.**
	Workaround: Use the GUI Installer to install the FIX Adapter in dependent mode.
	Alternate workaround: Perform a console installation using standalone mode. This is possible even if you already have Sybase CEP Server installed on the machine.
	Note: On all platforms, if you choose to install the FIX adapter in a location other than the `/opt/sybase/SybaseC8/enterprise-adapters` directory, you may see an error message reported in the install log file:
	`Install Directory: /opt/sybase/SybaseC8/`
	`Status: ERROR`
	`Additional notes: ERROR - java.lang.Exception: Failed to create directory: /opt/sybase/SybaseC8`
	Please ignore this message.

Known Issues for RAP Publisher

Learn about known issues and apply workarounds for RAP publisher.

Table 9. RAP publisher issues

CR#	Description
632323	**When RAP publisher is run from a Solaris workstation that is partitioned into Solaris zones, the RDS packet size is set to zero.** A new `libpublisher.so` file is required. **Workaround:** Contact Sybase customer service to obtain the latest RAP R4 patch on Solaris.

CR#	Description
632462	**The RAP publisher fails to start.** **Workaround:** When installing Sybase CEP R4 ESD#1 and using RAP adapter, edit the `publisher.xml` file: • From `SybaseC8/server/examples/FeaturEExamples/Adapters/RapInProcessAdapter/RAPSetting/rap-config/publisher.xml`, remove the line: `<PacketWindowSize>20</PacketWindowSize>`

Known Issues for TIBCO Adapter

Learn about known issues and apply workarounds for TIBCO adapter.

Table 10. TIBCO adapter issues

CR#	Description
N/A	**The TIBCO adapter corrupts BLOB type data.** **Workaround:** Use other datatypes when working with the TIBCO adapter.

Known Issues for Unpacking Tar Files

Learn about known issues and apply workarounds for unpacking tar files on Solaris platforms with GNU tar.

Table 11. Unpacking tar files on Solaris issues

CR#	Description
587935	**If the GNU tar command is not used when extracting Sybase CEP-related files on Solaris platforms, file names are truncated and unpacked improperly.** **Workaround:** To extract Sybase CEP-related files on Solaris platforms, use the GNU tar command. The name of the GNU tar binary is either `tar` or `gtar`. Run **tar --version** to find out if a binary is GNU tar or not.

Known Issues for RAP Adapter

Learn about known issues and apply workarounds for RAP adapter.

Table 12. RAP adapter issues

CR#	Description
632698	**The datetime2 and time2 datatypes for RAP adapter are not supported by Sybase CEP.** **Workaround:** N/A.

Known Issues for Sybase CEP Server

Learn about known issues and apply workarounds for Sybase CEP server.

Table 13. Sybase CEP server issues

CR#	Description
634215	**The Sybase CEP server may crash when DL is enabled and DL `FileNameFormat` has an invalid regular expression.** **Workaround:** Ensure that the regular expression you provide is valid.

Known Issues for Tracer Logic

When tracer logic is enabled, CEP Server may send C3606 error messages directly to the error log.

Warning C3606 indicates when the server discards a message that has a timestamp set prior to the current processing time.

Table 14. Tracer Logic Issues

CR#	Description

634070	**Sybase CEP (versions R3 ESD1, R4, and R4 ESD 1) does not display C3606 warnings when they occur. Users can only see this warning in the error log and, therefore, may miss the message if they are not monitoring the error log.**
	The issue is related to the tracer logic that is used for latency performance monitoring. The tracer logic is enabled by default.
	Workaround: Disable tracers in the c8-server.conf file.
	Change the "EnableTracers" value from the default of "true" to "false" as follows:
	`<section name="c8/Performance">`
	`  ... ...`
	`  <!--Enable/disable support for tracer messages to automatically calculate latency-->`
	`  <preference name="EnableTracers" value="false"/>`
	`  ... ...`

Documentation Clarifications

Read about updates, corrections, and clarifications to the documentation released with Sybase CEP.

Table 15. Sybase CEP documentation changes

CR#	Description
629917	**`c8_write_to_rap.adl` file** The `c8_write_to_rap.adl` filename for RAP In-Process adapter has changed to `c8_sybase_rap_publisher.adl`. Modify each instance of this filename to reflect this change.
N/A	**Sybase CEP games** The documentation mentions a number of games that are installed with the Sybase CEP R4 Studio component. These games are no longer included with the Sybase CEP product.
592593	**Sybase CEP library** `c8_studio` no longer requires the libiconv.so.1 library on Solaris platforms. The new library is libXinerama.so.1. The documentation for this revision references the SMClinconv package where it should say SUNWxwplt.
589213	**Guaranteed delivery** Guaranteed Delivery is only available in synchronous mode. This is not specified in the Sybase CEP documentation.
N/A	**Excel adapter** The *CEP Excel RTD Adapter Guide* incorrectly states that the ProgID of the Sybase CEP RTD plugin program is `C8.RTD`. The correct ProgID is `Sybase.C8.RTD`.
616961	**Starting the JMS Adapter** The *Integration Guide* has a mistake in the command string for starting the JMS Adapter: it is missing the slash between sybase and c8 in the command string. It should be`java com/sybase/c8/adapter/JMSadapter -topic=test1 -GD=true`.

CR#	Description
623789	**Integration Guide** The current instructions in *Sybase CEP Integration Guide> Sybase CEP Adapters> Adapters Supplied by Sybase CEP> RAP - The Trading Edition Adapters> Setting up UA for RAP - The Trading Edition Adapters* are no longer applicable. Use the instructions below instead: Unified Agent (UA) is a Java based agent that gathers statistics from a RAP publisher. It uses JMX technology and runs parallel to the server it monitors or administers. Sybase recommends that you copy the `bin/rap/ua` directory and use a different `ua` directory for each running instance of a RAP adapter. To start the UA agent and configure the RAP in-process or out-of-process adapter to package UA statistics: **1.** In `bin/rap/ua/bin/myenv`, set appropriate values for the following envars as specified in the "TODO" instructions at the top of the file: • SYBASE_UA • SYBASE_JRE • UA_RMI_PORT **Note:** UA_RMI_PORT in myenv should match the port property value in `service-config.xml`. **2.** In `bin/rap/ua/plugins/com.sybase.rap.rapservice/agent-plugin.xml`, set appropriate values for the following properties as specified in the file: • rap.service_host • rap.service_port • rap.service_startCommand • rap.service_pluginsubtype • rap.service_plugindisplayname If the `bin/rap` directory link points to the `rap-r4` directory, set the following properties in `agent-plugin.xml` also: • rap.service_logFile • rap.service_configFileDir • rap.service_componentTemplateDir **3.** In `in/rap/ua/service/RMI/service-config.xml`, set the appropriate value for the following property as specified in the file:

CR#	Description
	• port **Note:** The AdminPort node value in the adapter `publisher.xml` should match the rap.service_port property value in `agent-plugin.xml`.

Technical Support

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support.

If you have any questions about this installation or if you need assistance during the installation process, ask the designated person to contact Sybase Technical Support or the Sybase subsidiary in your area.

Other Sources of Information

Use the Sybase CEP Getting Started CD, the installed documentation, and the Sybase Product Manuals Web site to learn more about your product.

- The Sybase CEP Getting Started CD contains the *Release Bulletin* , the *Installation and Configuration Guide* , and the *Sybase CEP Adapter Installation Guide* in PDF format, and may also contain other documents or updated information not included on the product CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The Sybase Product Manuals Web site provides documentation that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/ Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

  To access the Sybase Product Manuals Web site, go to Product Manuals at *http://www.sybase.com/support/manuals/*.

Sybase Certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

Finding the Latest Information on Product Certifications

Details on product certifications are located on the Sybase website.

Point your Web browser to Technical Documents at *http://www.sybase.com/support/techdocs/*.

1. Click Certification Report.
2. In the Certification Report filter select a product, platform, and timeframe and then click Go.
3. Click a Certification Report title to display the report.

Finding the Latest Information on Component Certifications

Point your Web browser to Availability and Certification Reports at `http://certification.sybase.com/`.

1. Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
2. Select Search to display the availability and certification report for the selection.

Creating a Personalized View of the Sybase Web Site (Including Support Pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1. Point your Web browser to Technical Documents at *http://www.sybase.com/support/techdocs/*.
2. Click MySybase and create a MySybase profile.

Sybase EBFs and Software Maintenance

Find the latest information on EBFs and software maintenance

1. Point your Web browser to the Sybase Support Page at *http://www.sybase.com/support*.
2. Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
3. Select a product.
4. Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

   Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.
5. Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Sybase White Papers

As white papers on Sybase CEP Option R4 become available, they are posted on the Sybase Web site.

Find the latest Sybase CEP white papers:

1. Point your Web browser to Technical Documents at *http://www.sybase.com/support/techdocs/*.
2. Click "White Paper - Technical".
3. In the Product filter select CEP and then click "Go".
4. Click a white paper title to display the document.

Sybase CEP Option Free Download Terms

Refer to the root directory of your Sybase CEP R4 Server CD, Sybase CEP R4 Studio CD, or FIX Adapter R4 CD to find the Free Download Terms document for this release.

This document, named `CEPOption.R4.FreeDownloadTerms.pdf`, supersedes the Free Download Terms document included in the installer (`SybaseCEPOptionThirdPartyComponentsUserNotice.pdf`).

Accessibility Features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

The Sybase CEP Option R4 documentation complies with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

For information about accessibility support in the Sybase IQ plug-in for Sybase Central™, see "Using accessibility features" in Chapter 1, "Introducing Sybase IQ" in *Introduction to Sybase IQ*. The online help for Sybase IQ, which you can navigate using a screen reader, also describes accessibility features, including Sybase Central keyboard shortcuts.

Note: You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and MixedCase Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at *http://www.sybase.com/accessibility*. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.