Configuring Heterogeneous Replication server for MSSQL

Recommendations First

Sybase recommends that ECDA for ODBC, and the target database reside on the same machine.
MSSQL database should be set to capability mode 2005(90) - I am not sure that this is true. Early testing has shown that we can replicate to a MSSQL server DB set to 2008(100).

System requirements

Repserver, 512MB RAM, 380MB disk
ECDA, 512MB RAM, 300BM disk

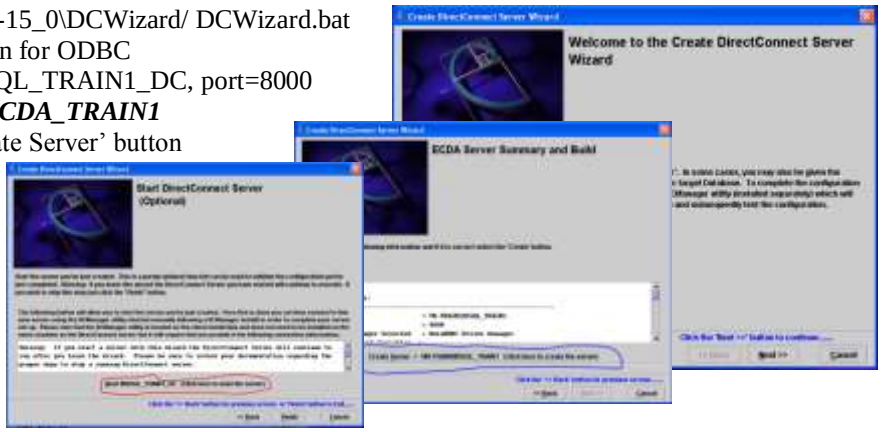
Configuring ECDA Option for ODBC

1. Open ODBC Driver manager from Control panel
2. Add System DNS connection for SQL Server driver called ***ECDA_TRAIN1***
3. Server name is VM_TRAIN\MSSQL_TRAIN1
4. Use 'SQL Server authentication', userID=sa, password=oranges1.

*note: in some clients systems, we have found that we had to setup the ODBC connection using a Domain User account and use a User DNS.

Configuring and starting the DirectConnect server

1. Start %SYBASE%\DC-15_0\DCWizard/ DCWizard.bat
2. Select the ECDA Option for ODBC
3. Set server name= MSSQL_TRAIN1_DC, port=8000
4. Enter service name = ***ECDA_TRAIN1***
5. Finally, select the 'Create Server' button
6. Start the service



7. Edit C:\sybase\DC-15_0\servers\MSSQL_TRAIN1_DC\cfg\ dcany.cfg to contain the following:-

```
[Service Library]
{Client Interaction}

[ECDA_TRAIN1]
{ACS Required}
ConnectionSpec1=ECDA_TRAIN1
{Client Interaction}
EnableAtStartup=yes
TransactionMode=long
SendWarningMessages=yes
{Target Interaction}
Allocate=connect
SQLTransformation=passthrough
ReturnNativeError=yes
{Catalog Stored Procedures}
CSPColumnODBCVersion=3
```

Note1: the section name and ConnectionSpec1 settings are set to the name of our service. This name **must** match the name given to the ODBC connection.

Testing

1. Start DirectConnect service, `C:\sybase\DC-15_0\bin>DCStart.bat -SMSSQL_TRAIN1_DC`
2. Connect using 'isql' from a MS-DOS session

```
C:\sybase\DC-15_0>DC_SYBASE.bat
C:\sybase\DC-15_0>isql -Usa -Poranges1 -SECDA_TRAIN1
```

Verify the connection to the replicate Microsoft SQL Server database by obtaining the DBMS name and version number:

```
select @@sqldbmsname
go
SQLDbmsName
--------------------
Microsoft SQL Server
```

Well done. You have configured the DirectConnect server!

Create a replication maintenance user in Microsoft SQL Server

1. In the Sybase serer (primary)

```
sp_addlogin rep_maint, rep_maint_ps, pubs2
go
sp_role 'grant', replication_role, rep_maint
go
use pubs2
go
sp_addalias rep_maint, dbo
go
```

2. In MSSQL server (replicate)

```
CREATE LOGIN rep_maint WITH PASSWORD='rep_maint_ps',
DEFAULT_DATABASE=pubs2
go
use pubs2
go
CREATE USER rep_maint FOR LOGIN rep_maint;
go
EXEC sp_addrolemember 'db_owner', 'rep_maint';
go
```

Add databases to Replication System

Primary DB

Use *rs_init* to add primary database. This will configure the rep_agent and rs_* objects to the ASE database and add the connection to the repserver.

1. Start *rs_init* and select to 'Add a database' from the replication configuration dialogue
2. Give a server name of SYB_TRAIN1_RS
3. In the 'Database Information dialogue box fill in all the details and make sure you answer 'yes' to 'Will the database be replicated'. Accept the suggested user name and password for the maintenance user.
4. Finally, once you are happy that all the information, select 'Continue' from the 'Add database to replication system' dialogue box.

Replicate DB

With the DC still running, log into the RS server

```
create connection to ECDA_TRAIN1.pubs2
using profile rs_ase_to_msss;standard
set username rep_maint
set password "rep_maint_ps"
go
admin who
go
```

You should see similar entries to this:-

```
   28 DSI EXEC    Awaiting Command     105(1) ECDA_TRAIN1.pubs2
   27 DSI         Awaiting Message     105 ECDA_TRAIN1.pubs2
   26 SQM         Awaiting Message     105:0 ECDA_TRAIN1.pubs2
```

Get replicating

We will add a database, and then define a simple replication definition (repdef), which we will subscribe to in MSSQL Server.

Creating the primary database

ASE

```
disk init name="pubs2_data01", physname="C:\sybase_15\pubs2_data01.dat", size="20M"
go
sp_diskdefault  pubs2_data01, defaulton
go
sp_diskdefault master, defaultoff
go


cd $SYBASE/ASE-15_0/scripts
ISQL -iinstpbs2
```

MSSQL

```
Create small database using wizard, called pubs2
```

Create the Replication Definition

```
In ASE,
use pubs2
go
-- get the repdef from the table
1> sp__gen_repdef_ashley "create", publishers , LOGICAL_SRV, logical_db
2> go
create replication definition publishers_repdef
with primary at LOGICAL_SRV.logical_db
with all tables named 'publishers'
(
  pub_id char(4),
  pub_name varchar(40),
  city varchar(20),
  state char(2)
)
primary key (pub_id)
replicate minimal columns
```

Run the above output into the RS!

Synchronise the table data between ASE & MSSQL

In ASE,

```
-- First, reverse engineer the table
1> sp__revtable publishers
2> go
 -- Table_DDL
 ------------------------------------------------------------------------
 CREATE TABLE  publishers
 (
  pub_id    char(4)                     NOT NULL,
  pub_name  varchar(40)                 NULL,
  city      varchar(20)                 NULL,
  state     char(2)                     NULL
 )
 -- Index_DDL
 ------------------------------------------------------------------------
create unique clustered index pubind on dbo.publishers (pub_id)
```

Run the above output into MSSQL, pubs2 database, to create the empty table.

Use BCP to copy data:-

```
bcp pubs2..publishers out publishers.txt -SSYB_TRAIN1 -Usa -Poranges1 -c

msbcp pubs2..publishers in publishers.txt -Usa -Poranges1 -SVM-TRAIN\MSSQL_TRAIN1 -c
```

*note, I have changed the name of the Microsoft bcp.exe to msbcp.exe, to differentiate between the two versions of this utility.

Create Subscription to our new repdef

In RS,

```
create subscription publishers_sub
for publishers_repdef
with replicate at ECDA_TRAIN1.pubs2
without materialization

check subscription publishers_sub
for publishers_repdef
with replicate at ECDA_TRAIN1.pubs2
```

Mark the table for replication

This step is vital, yet very simple. I have lost track of the number of times I have forgotten to do this!
In ASE,

```
sp_setreptable publishers, true
```

Insert sample row

```
insert into publishers values('9999', 'Fancypants publishing', 'London', 'NA')
```

Now in ASE & MSSQL compare the returned rows from

```
Truncate tab
```

If they are the same, congratulations you have installed replication between Sybase and MSSQL, using a simple repdef-subscription model. If the results are not the same, go back and check each step, one at a time.

Supplementary Note I - How to automatically start DirectConnect server at boot time?

(From the manual DirectConnect for z/OS Option Server Administration Guide for DirectConnect)
Chapter 5 Setting up SSL and a Windows Service

Installing a DirectConnect server as a Windows service
DirectConnect no longer automatically creates the server as a Windows service. However, you can run a DirectConnect server as a Windows service. The following describes how to register, configure, start, stop, and remove DirectConnect as a Windows service.

Register DirectConnect as a Windows service
To register DirectConnect:
Execute the AddWinService.bat located in the DC-12_6\bin directory using your name for the "Windows service name" and the name of the DirectConnect server:
AddWinService <Windows_service_name> <server_name>

For example:
AddWinService dcservice dcserver

Configure DirectConnect as a Windows service
To configure DirectConnect to run as a Windows service:
Install DirectConnect on your local hard drive.
WARNING! Do not install DirectConnect under a clear case filesystem, or remotely, or in a shared file system, it may cause DirectConnect to fail during file I/O.

Turn off all server tracing and logging to the screen (console) by accessing the "Logging' and "Tracing" configuration properties.

Perform DirectConnect licensing as described in Chapter 2, "Sybase Software Asset Management (SySAM)," of the installation guide for your platform.

To ensure the environment created by DC_SYBASE.bat is also your system environment. Set SYBASE, SYBASE_ECON, SYBASE_SYSAM, and PATH accordingly.
Verify configuration by starting the DirectConnect sever using DCStart.bat at the command line. Verify that only the copyright notice is written to the screen (console). No licensing dialog should be displayed.

Start DirectConnect as a Windows service
To start DirectConnect as a Windows service:
Open the Windows Services control panel.
Right click on the DirectConnect Windows service.
Select start.

Stop DirectConnect as a Windows service
To stop DirectConnect as a Windows service:
Open the Windows Services control panel.
Right click on the DirectConnect Windows service.
Select stop.

Remove DirectConnect as a Windows service
To remove DirectConnect as a Windows service:
Using the Windows Services control panel, stop the DirectConnect service.
Execute the RemoveWinService.bat located in the DC-12_6\bin directory using the name of the DirectConnect Windows service. Enter the following:
RemoveWinService <Windows_service_name>

These bat files are in the directory is %SYBASE\DC-12_6\bin

Supplementary Note II – adding additional 'Services'

You can have multiple database connections against a single Direct Connect server. These are called (rather confusingly, Services). The following is and example of how to add a new 'Service' to an existing DC server.

1. Create new ODBC connection to target server (MSSQL) and specify the name of the target database.
2. Next, to help the DC know how to connect to the new service, add the following to the SQL.INI file. Note that the Port number is the same for the DC and all services that run under it

```
[EDCA_TRAIN2]
query=NLWNSCK, vm-train, 8000
win3_query=NLWNSCK, vm-train, 8000
```

3. Finally, configure the new 'Service' by adding the following to the dcany.cfg file.

```
[ECDA_TRAIN2]
{ACS Required}
ConnectionSpec1=ECDA_TRAIN2
{Client Interaction}
EnableAtStartup=yes
TransactionMode=long
SendWarningMessages=yes
{Target Interaction}
Allocate=connect
SQLTransformation=passthrough
ReturnNativeError=yes
{Catalog Stored Procedures}
CSPColumnODBCVersion=3
```

Supplementary Note III – recommended config values from Sybase engineering

Below are the appropriate DC/ECDA config settings for a replication environment. Please compare with your current settings and update accordingly.

Allocate
Controls when an access service allocates conversations with the target database system.
Syntax Allocate=[connect | request]
Default connect

Values

connect specifies an access service to allocate the conversation when the client connects, and holds it open for the duration of the client connection.

request specifies an access service to allocate a new conversation each time the client application sends a request, and deallocates the conversation after each request.

Note: There is a large performance penalty when using the request setting.

TransactionMode
Specifies whether the access service or the client application manages commit and rollback statements.

Syntax TransactionMode=[short | long]

Default short
Values:-
long specifies the access service to give commitment control to the client application.
short specifies the access service to issue a commit or a rollback after each request.

Comment: The access service holds open the connection to the data source until the client application issues a commit or rollback, or until the ClientIdleTimeout value is exceeded.

SQLTransformation
Specifies the mode the access service uses for SQL transformation.
Syntax SQLTransformation=[passthrough | sybase | tsql0 | tsql1 | tsql2]

Default passthrough
Values
passthrough specifies an access service to send all SQL statements to the database system as received, without transformation. A client application uses passthrough mode to gain direct access to DBMS capabilities.
sybase specifies an access service to perform SQL transformation of selected statements. It also allows the use of multi-part table names with the view command in SQL statements.

CSPColumnODBCVersion
Specifies ODBC version that catalog stored procedures results conform to.
This affects interoperability with ASE/CIS.

Syntax CSPColumnODBC Version = [2 | 3]

Default 3
Values
2 specifies ASE/CIS version 12.0.
3 specifies ASE/CIS version 12.5 and later.
This property affects interoperability with ASE/CIS.

ReturnNativeError
Allows a non-localized native error message and a native error severity to be returned to the client.

Syntax ReturnNativeError = [yes | no]

Default no
Values
yes specifies non-localized native error messages are returned to the client.
no specifies non-localized native error messages are not returned to the client.

SendWarningMessages
Specifies whether an access service returns warning messages to the client application.

Syntax SendWarningMessages=[no | yes]

Default no
Values
no specifies the access service not to return warning messages to the client application.
yes specifies the access service to return warning messages to the client application.

Supplementary Note IV – Tracing

"TraceOpenServer =59" in server.cfg file & restart DC

Then see resulting files:
%SYBASE%\%SYBASE_ECON%\SERVERNAME\log\srv.log
%SYBASE%\%SYBASE_ECON%\ SERVERNAME\log\SERVERNAME.trc