

Information Engineering Technology

# Install Guide - Multiple SQLServer CSE



Release 8.7.3

# Table Of Contents

<b>Multi-CSE Install for SQL Server .....</b>	<b>3</b>
Pre-Requisites .....	3
Understanding Connections (Single CSE).....	3
Understanding Connections (Multiple CSEs) .....	3
Preparing SQLServer .....	4
Considerations for Windows Server .....	5
Performing the Installs.....	7
Cluster Aggregate Set Definition Index.....	11
Remote Shell Daemon .....	11
Shared Network Drives .....	11
GuardIEn Remote Scripts.....	12
Task Dispatcher / Queues .....	12
Code Generation.....	12
Known Restrictions and Problems .....	12

# Multi-CSE Install for SQL Server

This document describes how to configure GuardIEn to connect, view and manage data across multiple SQL Server CSE databases.

## Pre-Requisites

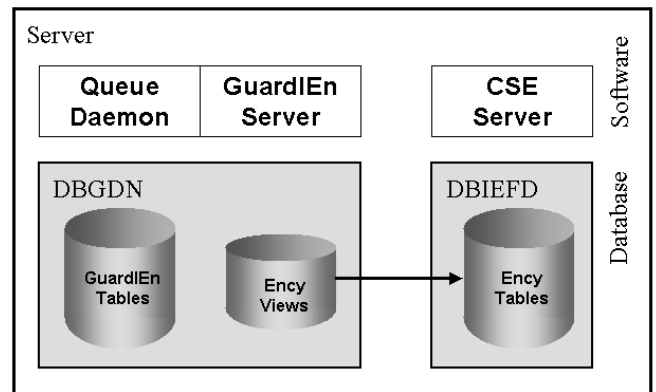
It is assumed that you have already configured a 'main' GuardIEn database server, which is where the physical GuardIEn tables are located.

Before you go any further you need to ensure that you can successfully connect your Client Manager to the TE and access GuardIEn successfully.

## Understanding Connections (Single CSE)

The server executes GuardIEn Distributed Process Servers (servers for the GUI clients) under the control of the AEFAD (TE Daemon). It also executes the Task Dispatcher whose function is to process GuardIEn Tasks.

In a single CSE set-up, GuardIEn Servers and Queue Daemons connect to the DBGDN database using sa.password, which owns the GuardIEn tables. DBGDN contains actual GuardIEn tables plus views to the local CSE's tables.



In this diagram the separate databases are shown as grey boxes with the relevant tables and views shown within the boxes.

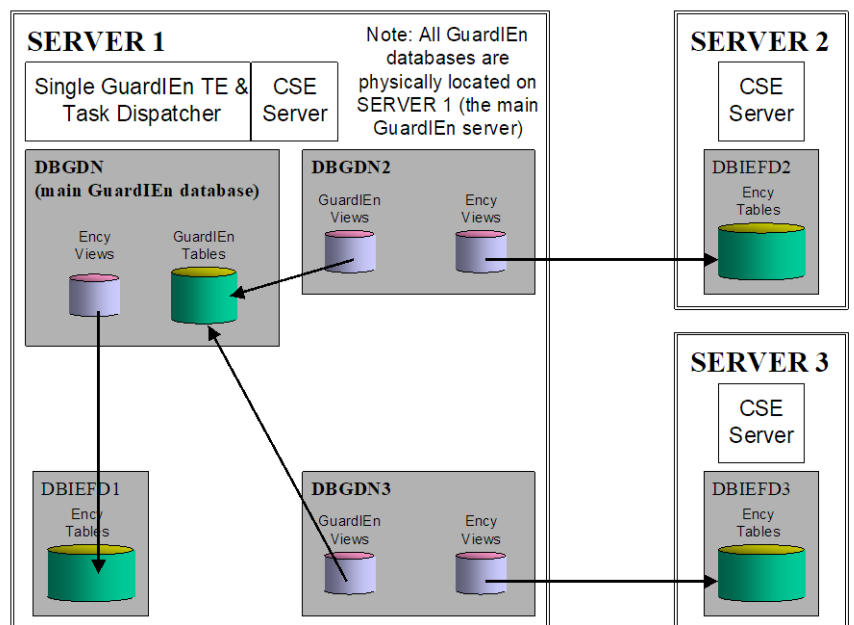
## Understanding Connections (Multiple CSEs)

The recommended approach for multiple CSEs is to install the GuardIEn software (and TE) once on a main GuardIEn server, and also create one GuardIEn database per CSE on that main GuardIEn server, each database viewing one of the CSEs.

Hence there will be only one copy of the GuardIEn software and just one TE to connect to.

When GuardIEn needs to access data from another CSE, it connects to a different database, locally on the main server, for example DBGDN2. The connect details that it should use are defined as properties of the encyclopaedia in the table CDW\_ENCY.

This diagram shows an example architecture for 3 CSEs.



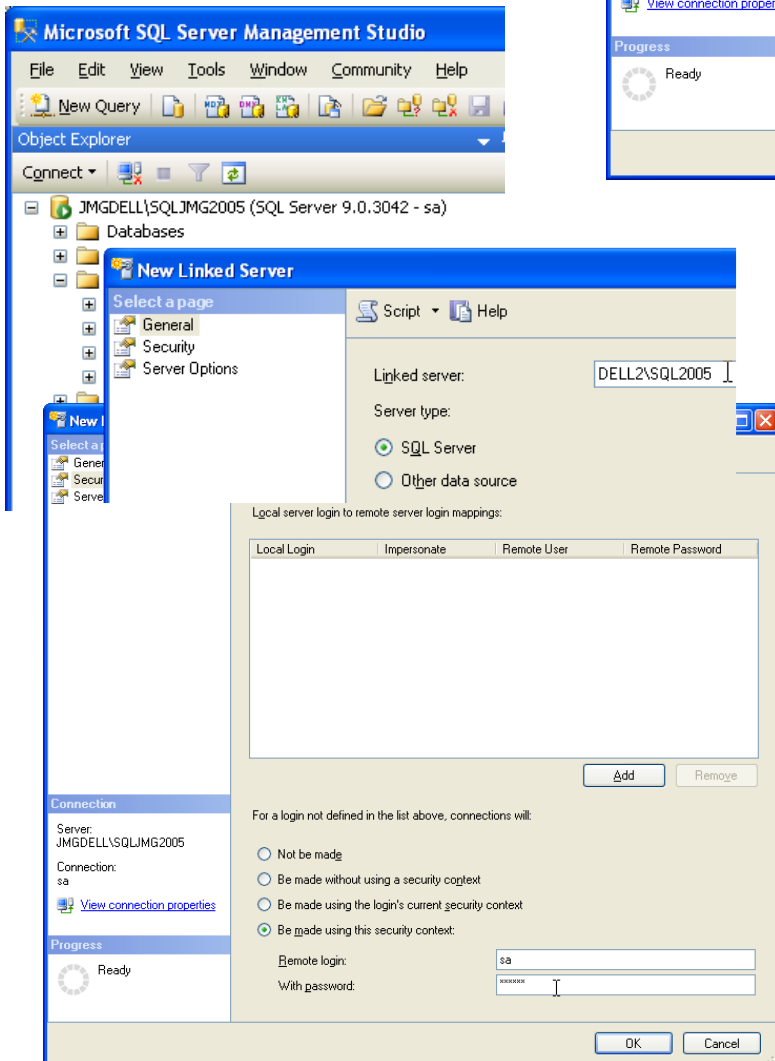
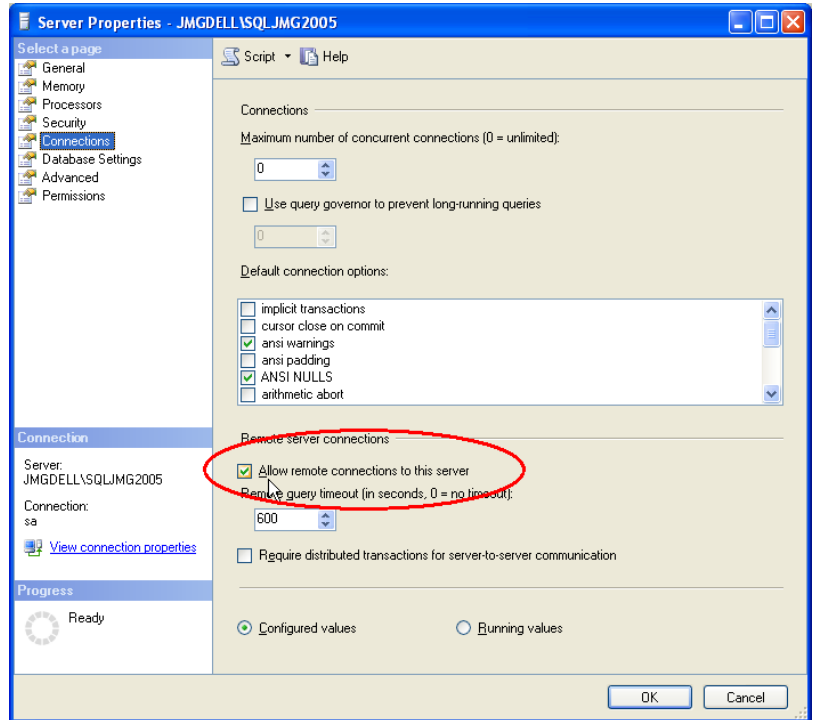
# Preparing SQLServer

Before creating the mini GuardIEn databases for your remote CSEs, you must prepare SQL Server so that the main GuardIEn server can view the all of the other servers. The easiest method is to use the *SQLServer Management Studio* on the main GuardIEn server and follow this process.

Note that you will need both the SQL Server database service started and also the *SQL Server Browser* service started, on both servers.

In both servers, use the Management Studio to set the *Allow remote connections to this server* option. Get to here by right-clicking the Server itself, choosing *Properties*, then opening the *Connections* page.

Next, in the Management Studio on the main GuardIEn server, expand *Server Objects* and right click, then choose *New Linked Server* from the pop-up menu.



In the *General* page, enter the remote CSE's SQLServer name and choose *SQL Server* for the *Server type*.

In the *Security* page, select *Be made using this security context* and enter the remote SQLServer's administrator login details.

Press *OK* when done and this should successfully create and validate that this remote connection works.

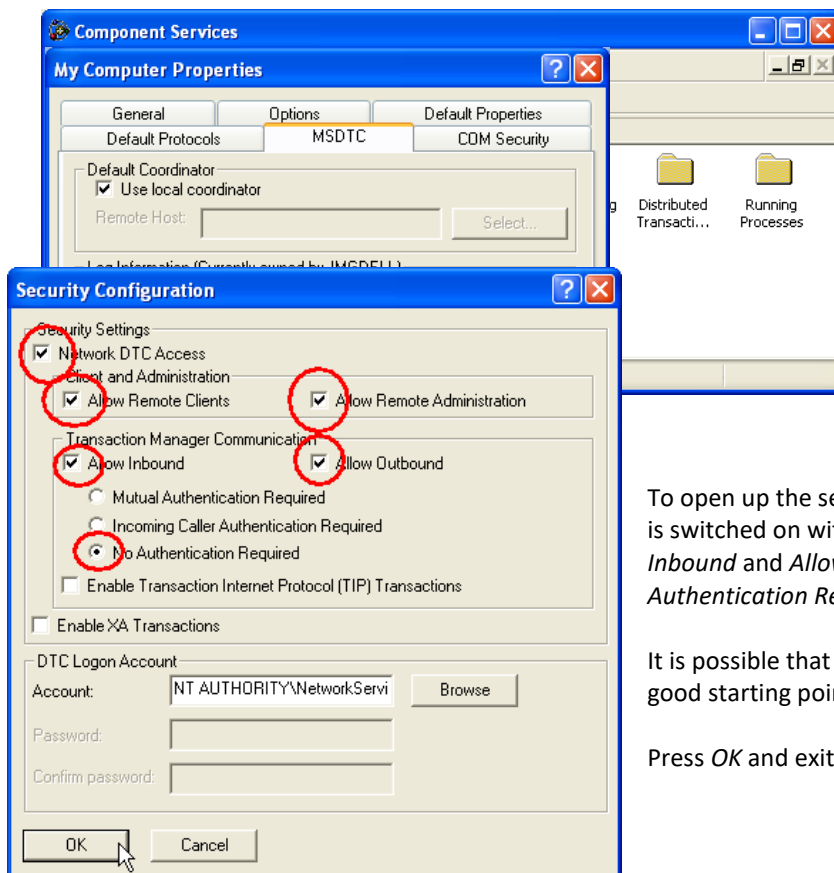
# Considerations for Windows Server

You may need to configure security to allow Microsoft's MSDTC service to communicate between servers.

Windows includes many security features that prevent unwanted access to your machine. Unfortunately, in locking down so many services and ports resulted in certain applications no longer being able to function. In particular, the MSDTC (Distributed Transaction Co-ordinator) will not communicate between servers until you open up its ports and allow its security to work more openly. MSDTC is required by GuardIEn in a Multi-CSE SQL Server configuration.

## Configure MSDTC Component

To open the MSDTC security up enough to allow linked-server communication in SQL Server, open the *Component Services* application (normally found on your *Administrative Tools* start menu). In this console, expand *Component Services*, then *Computers*, then *My Computer*. Right-click *My Computer* and choose *Properties* from the pop-up menu.



In the *Properties* dialog, change to the *MSDTC* tab and press the *Security Configuration* button.

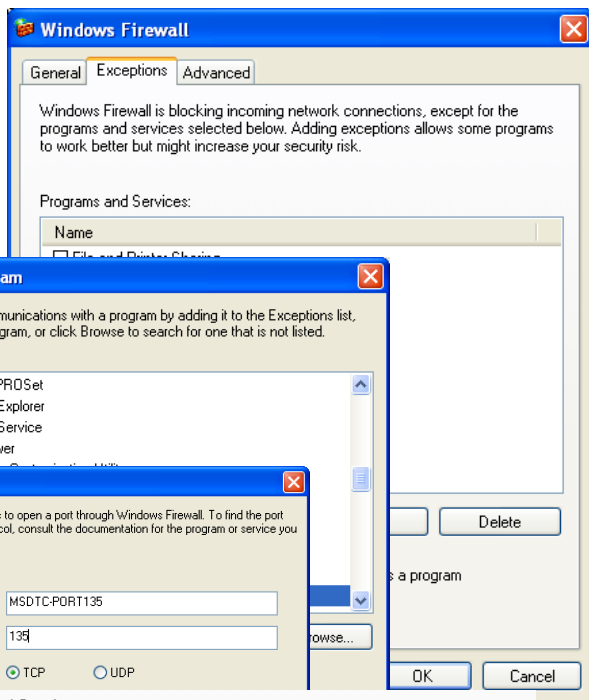
To open up the security for MSDTC, ensure that *Network DTC Access* is switched on with *Allow Remote Clients*. Also ensure that *Allow Inbound* and *Allow Outbound* are switched on, and that *No Authentication Required* is selected.

It is possible that a more secure configure may also work, but this is a good starting point to get the connection running and tested.

Press *OK* and exit from the *Component Services* application.

## Configure Firewall for MSDTC

Windows also provides a Firewall, which secures most ports and applications by default. You will need to allow Port 135 for the MSDTC.EXE application... from the Windows Control Panel, open the *Security Center* and choose *Windows Firewall* to manage...

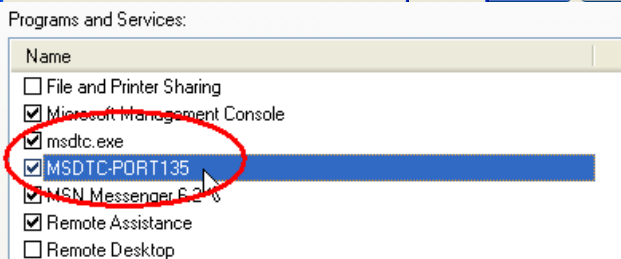


Change to the *Exceptions* tab and click the *Add Program* button.

Use *Browse* to locate the MSDTC.EXE executable (in your Windows\System32 directory) and press *OK*.

Next use the *Add Port* button to open Port 135, which is used by MSDTC. You can give the Port whatever name you like, such as "MSDTC-Port135". Press *OK* and close the Firewall configuration.

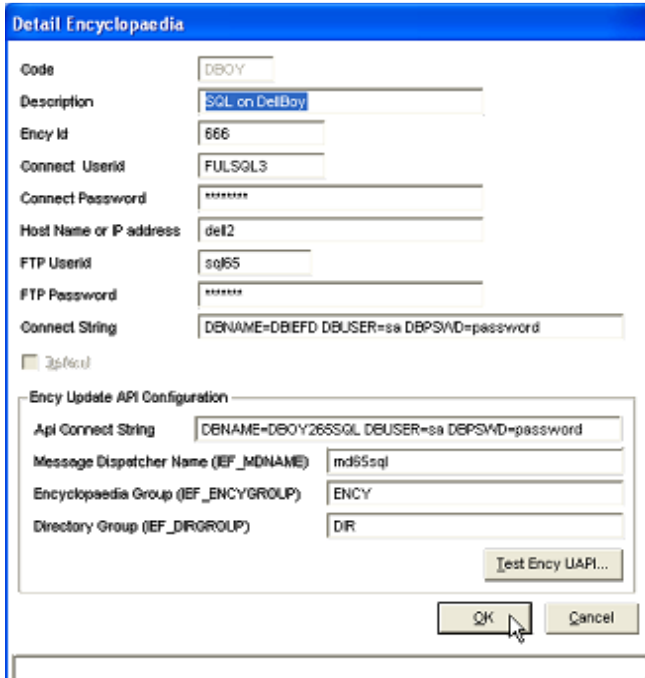
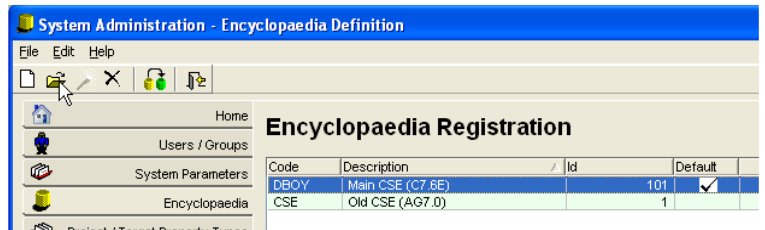
MSDTC should now be enabled and ready to use from that server.



# Performing the Installs

The *mini* GuardIEn databases are created using a function in the *System Administration* client. We shall look at that function now...

Start the *System Administration* client and open the list of Encyclopaedias defined to GuardIEn.



Initially you will only see your default CSE as set-up by the install program.

To begin the process of creating links to a remote CSE, use the **New** function and fill in the fields on the *Detail Encyclopaedia* dialog as follows:

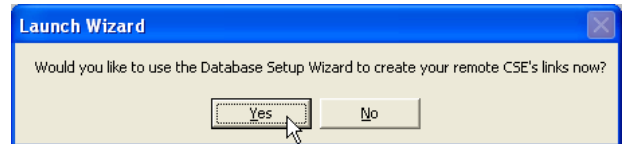
Field	Description	Examples
Code	Choose a unique 4-character code to identify the remote CSE.	SRVn
Description	Enter a long description for the remote CSE. If the Gen version is different on the remote CSE to the local CSE, then you need to include a special string in the description to indicate the version e.g. (CG7.6E)... see the System Admin documentation on Ency definition for more info.	My other CSE (CG7.6E)
Ency Id	Enter the Gen Encyclopaedia Id for that CSE – you can find this in the Co-Ordination client by detailing that Ency.	
Connect Userid	The SQLServer <b>Database Name</b> (not connect userid!) that will be created with views to the main GuardIEn tables as well as views to the remote CSE's tables.	GDNSV2
Connect Password	The SQLServer sa userid and password for the remote CSE.	sa.password
Host Name or IP address	The network name (DNS) or TCP/IP address for the remote server.	10.20.30.40 or myserver
FTP Userid	A userid which can logon to the remote server through FTP.	gdn
FTP Password	The password for the above FTP Userid.	gdn
Connect String	The remote CSE's database connection string.	DBNAME=DBIEFD DBUSER=sa DBPSWD=password
API Connect String	Similar to the above Connect String, but should be relative from the GuardIEn server machine, so the DBNAME must be an ODBC datasource defined on the GuardIEn server that can see the remote CSE's database.	DBNAME=CSE2ODBCSRC DBUSER=sa DBPSWD=password
Message Dispatcher Name	The value of IEF_MDNAME for the remote CSE, as viewed from the GuardIEn server machine.	iefmd2
Encyclopaedia Group	IEF_ENCYGROUP for the remote CSE	ENCY2
Dir Group	IEF_DIRGROUP for the remote CSE	DIR2

If you are not familiar with all of these values yet, you should obtain them now before continuing with the remote CSE's definition.

The API details only need to be filled in if you intend to use genIE Direct's update functions (updates from a PAD Viewer). A *Test* button is available to check that the basic details of the API connection are correct. Note that this does not guarantee that the API will be available, but is a good starting test.

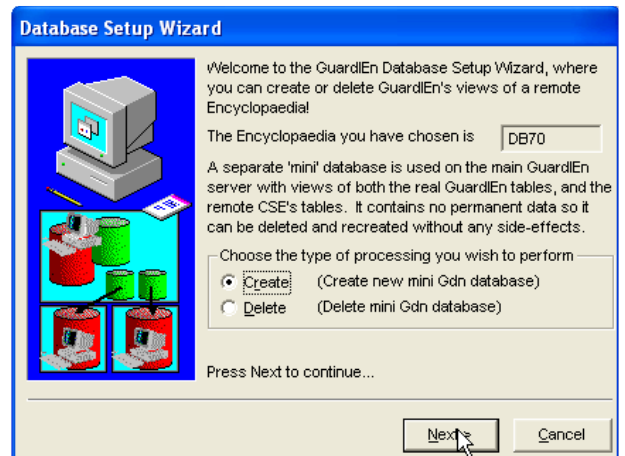


Once you have completed all of these values, press **OK** to create the CSE's definition in the GuardIEn database. This inserts a single row for the new CSE in the CDW\_ENCY table. Whenever you create a new record for a remote CSE, this will be detected automatically and you will be prompted to begin the physical creation of the mini GuardIEn database required to actually connect to that CSE. Hence the following question will be presented:



If you answer **No** then you can always perform the task later by clicking on your new CSE's record in the *Encyclopaedia Definition* list and using the **Wizard** function.

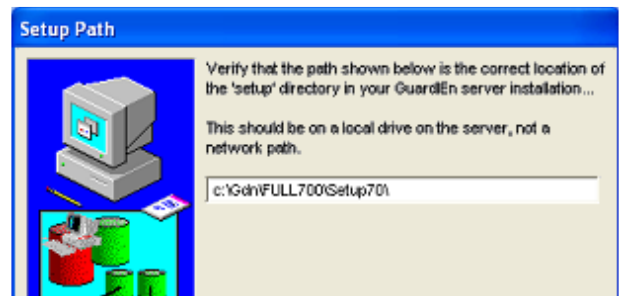
If you choose **Yes** or launch the wizard later, you will be presented with the *Database Set-Up Wizard*...



This wizard can be used to create or delete the GuardIEn links to a remote CSE. If you already have such a link and would like to upgrade it to the later GuardIEn release, performing a *Delete* followed by a *Create* is the correct approach to use.

In our example, we are creating a new link to a remote CSE. Select **Create** on this first dialog and press **Next** to begin the creation portion of the wizard...

You will be asked to enter the *Setup* path for your GuardIEn server installation (relative to the server). The wizard will hopefully have worked this out for you automatically, but you should verify it and press **Next** when you are ready. The wizard will perform some checks itself at this point to ensure that the required set-up files are located beneath the chosen path.

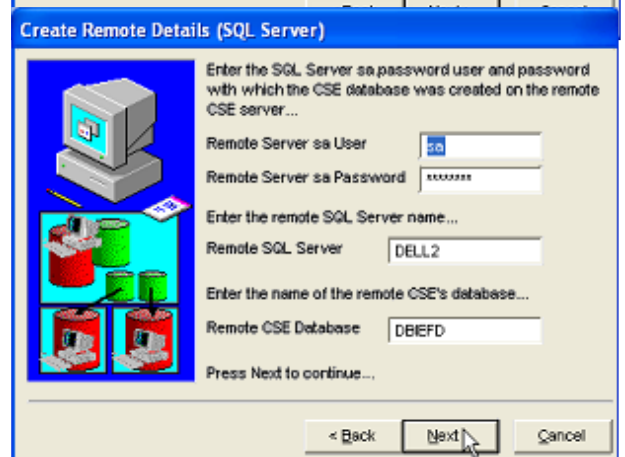
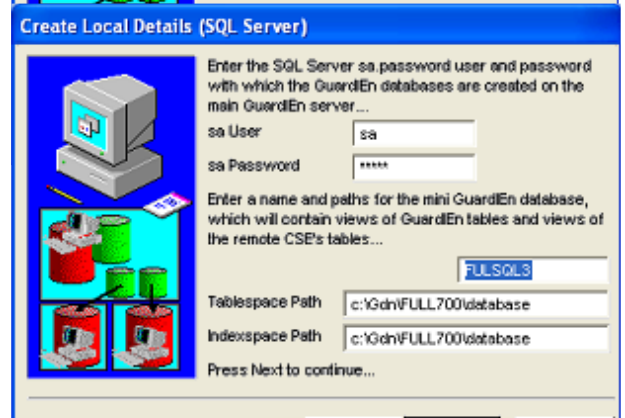


After these verifications are made, the wizard presents the next dialog, to begin gathering more information related to the SQLServer parameters required to create a new mini GuardIEn database...

The wizard will make some suggestions based on your current installation and the values that you provided when you defined the new CSE to GuardIEn:

- Sa User/Password – these are taken from the Connect Password for you *default* CSE and should be the sa connect userid.password for your **main** server where the GuardIEn server is installed.
- Mini DB Name – this is a suggested name which includes the Ency Code that you provided for the Ency definition, and will be the name of the new SQLServer database which contains views of the main GuardIEn tables plus views of the remote CSE's table.
- Indexspace Path/Tablespace Path – these are suggested as the same path that your main GuardIEn tablespace/indexspace files are located

Usually, the suggested values will be ok. The wizard will prevent you from using values which might corrupt your main GuardIEn database. Press **Next** to continue...

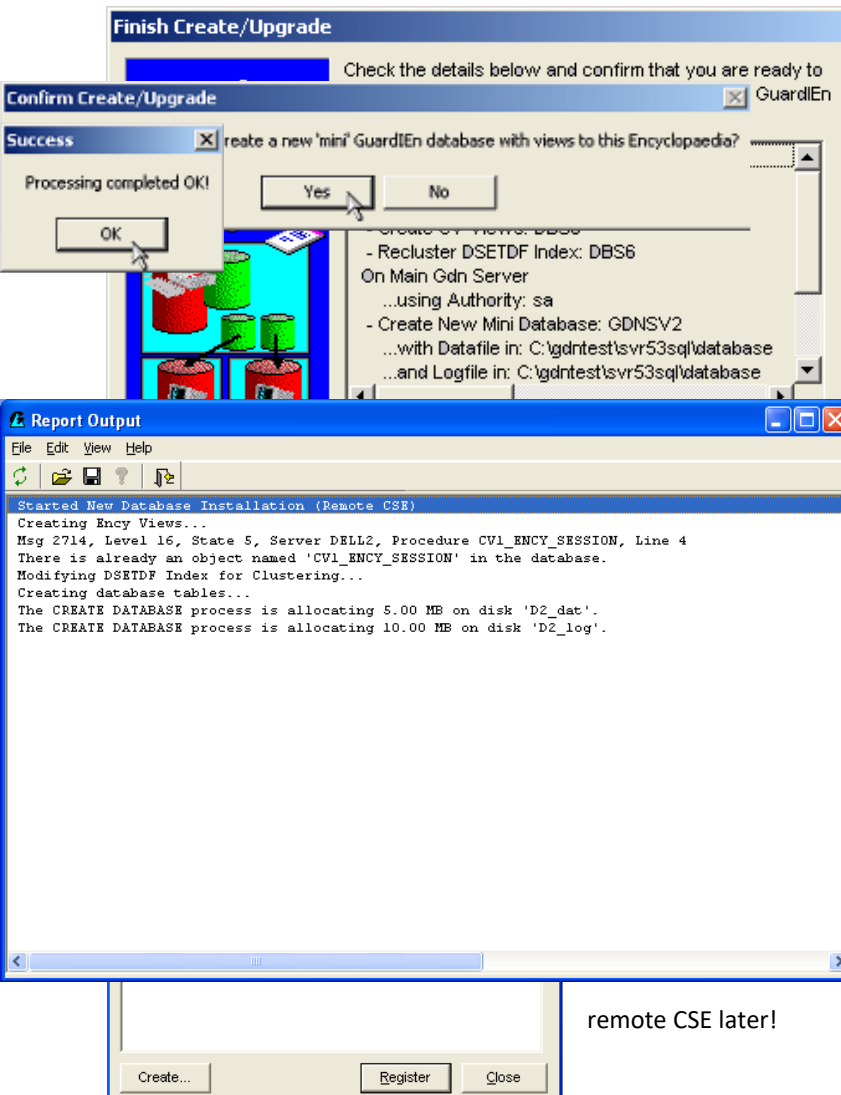


This dialog asks for information about the remote CSE. Again, most of the information is obtained from the values you entered earlier:

- Remote CSE User/Password – these are taken from the Connect Password for you *remote* CSE and should be the sa connect userid.password for your remote server.
- Remote SQLServer – this is suggested, but probably needs amending. It is the SQLServer *NAME* of the Server where the remote CSE resides – as defined when you registered the remote SQLServer using the *Enterprise Manager*. Your DBA should be able to configure/provide this value for you.
- Remote CSE Database – this is also taken from the Connect String that you provided for the Ency definition.

When you are happy with these values, press **Next** to proceed...

The wizard does not need any more information to begin creating the mini GuardIEn database to view the remote CSE, so a confirmation panel is displayed where you can review all of the values you just chose. Press **Create** to commence the database creation...



Confirm **Yes** and the creation will begin...

The wizard will attempt to detect failures as it proceeds, as well as performing a short test at the end to see if the new connection works. If all is ok, then you will see a *Success* message. In addition, you can review the output from the DDL processed...

Don't worry if you see an *error 'cv1\_ency\_session' already exists* messages – this is normal if you have previously configured that remote CSE as a GuardIEn-ready CSE.

Your remote CSE is now available to connect to via GuardIEn!

A good *real* test after this is complete, is to register some *Aggregate Sets* from the remote CSE using the *System Administration* client. This will not only confirm that the connection is ok, but will be required if you intend to perform any migrations which involve the

remote CSE later!

# Cluster Aggregate Set Definition Index

The primary index on DSETDF must be clustered explicitly in order for updates to be made to Aggregate Set contents by GuardIEn migrates to overcome a SQL Server bug in deleting rows across a linked server. A DDL script is supplied to perform this called *GdClusterDSETDF.ins/sql*. Run this script for each of your remote CSEs. Note that *if you used the Database Set-Up Wizard* described above then this will have been done for you automatically already.

## Remote Shell Daemon

GuardIEn will require a remote shell daemon to be present and active on each of the remote servers. This is to handle rsh commands to perform server-specific jobs e.g. run uploads/downloads/generates etc. against the remote server's CSE. IET recommends WRSHDNT from DeniComp, although other versions of daemons that support the same features should work just as well.

If you choose to use the DeniComp software then you need to perform the following configuration (most set-up is done via the Control Panel for WRSHD):

- Attempt Redirection on every command – ON
- Default window type – MINIMIZED
- Environment Variable File - <wrshdnt install path>\%ruser%.var
- Execute all RSH commands as user - <user>/<password> - e.g. gdn – it is recommended that you define a special user just for GuardIEn e.g. gdn, and define this user to NT. You will never actually log onto Windows with this user, but RSH will use this logon.
- All other options in the Control Panel for WRSDH should be blank or off
- Configure WRSHD variable files... create a <userid>.var file containing the environment of the Message Dispatcher (example below). The processes will actually run under gdn (or whatever you choose above). You will need a separate userid.var file per CSE on each remote server, e.g. if you have 3 CSEs on one server, then you should create 3 userid.var files e.g. csea.var, cseb.var and csec.var, each containing the environment for their own CSEs. Note: no SET keywords should be used in the .var files. The value of <userid> should be the same as the value entered for the FTP userid in the GuardIEn encyclopaedia definition. An example .var file for Gen 7.6 might look something like this:

```
AEHOME=c:\CA\GEN\  
COOL-BASE=c:\CA\GEN\  
IEF=c:\CA\GEN\  
IEFCONST=c:\CA\CLIENT\  
IEFCSGEN=c:\CA\CSE\bin  
IEFGEN=c:\CA\GEN\  
IEFGXTP=c:\CA\translat\  
IEFH=c:\CA\GEN\  
IEFLAN=c:\CA\CLIENT\  
IEF_CCP=c:\CA\CLIENT\  
IEF_DIRGROUP=DIR  
IEF_ENCYGROUP=ENCY  
IEF_MDMODE=TCPIP  
IEF_RDSNODE=localhost  
INCLUDE=c:\CA\GEN;c:\CA\CSE\include;%INCLUDE%  
LIB=c:\CA\CSE\lib;c:\CA\GEN;%LIB%  
PATH=c:\CA\GEN;c:\CA\CLIENT;c:\CA\CSE\bin;c:\CA;%PATH%  
IEF_MDNAME=iefmd
```

## Shared Network Drives

Set-up a Shared Network Drive for GuardIEn e.g. drive X: which should point to the GuardIEn root directory on the main GuardIEn server from all servers i.e. server 2 should see X: the same as server 1 etc. Under this drive will be the GuardIEn home directory, queue home directories, PAD Listing path and any code gen directories. Of course, you may specify different directories for queue home or code gen, but use the same Shared Network Drive technique.

For WRSHD configuration, create a file called automap.ini in the WRSHD home that has NET USE commands for the Network Drives you have e.g.:

```
net use x: \\server\c$\gdn
```

Start the WRSHD service and verify from a DOS Prompt with something like the following (replace csea with the userid from the userid.var file you created earlier). You should see the correct environment variables returned for user csea.

```
rsh SERVER2 -l csea set
```

# GuardIEn Remote Scripts

Copy supplied files `gdruncmd.exe`, `gdrmrtrun.bat` and `gdgetvar.bat` (placed in the `setup86\remote` directory where you installed GuardIEn by the set-up program) into each of the remote servers to a directory that is in the PATH of the remote server e.g. `\windows\system32`.

## Task Dispatcher / Queues

The Task Dispatcher will physically run on the main GuardIEn server, submitting remote shell commands (`rsh`) to invoke CSE processes, such as downloads/uploads etc., on remote CSEs.

In the Queue Manager, point all of your Queues to a common Shared Network Path e.g. `X:` for the drive and `\GdnQueues\` for the path. The Task Dispatcher will take care of stopping/starting your queues automatically according to their start/stop times.

## Code Generation

Code Generation must be performed into shared network drives too. Amend any Code Generation paths in your Target Properties to point to a common Shared Network Path e.g. `X:\MYPROJ\DEV\` etc. You *do not need* to do this for generations out of the CSE on the main GuardIEn server.

## Known Restrictions and Problems

1. Do not use spaces in the path names for the shared directories, i.e. queue home, pad listing and code generation. If you do, you should specify them with DOS short names e.g. `progra~1` instead.
2. The system date/time on each of the servers must be less than 1 minute apart, ideally kept in synch with a time synchronisation service.
3. Local install methods will execute on the GuardIEn server. If the installs have to execute on another server, then remote install method should be used.
4. Some queries and updates are particularly slow against a remote CSE. You may wish to contact IET to discuss how performance can be improved. A hidden option allows for better performance (environment variable `GDN_MULTICSE=YES`), but you may wish to discuss what effects this has with IET first.
5. If you are using encyclopaedias that have a mixture of Gen releases, e.g. Gen 7.6 on one CSE and Gen 6.5 on another and a single GuardIEn project will contain models from more than one encyclopaedia, then you must ensure that the schema tables for each encyclopaedia are loaded to contain the highest schema defined for models in the project.