



## Cisco WAN Manager Peer-to-Peer Communication

---

Release 12 of Cisco WAN Manager has been designed to enable multiple CWM workstations to manage a network with improved synchronization and scalability. Due to the size and growth of networks, it is faster to retrieve initial user data from another CWM workstation that is already running and synchronized with the network. An industry standard CORBA architecture is used to implement the communications between two or more CWM workstations through a server-client structure for communications between the CWM server and client processes.

CWM workstations use peer-to-peer communications to synchronize user data with each other. When user data is provisioned or changed, the CWM workstations will propagate the new data to the other CWM workstations. The user is able to continue the provisioning of network data, even when communications between a Primary CWM and Secondary CWM have been interrupted. If for any reason the communications between CWM servers are interrupted, the provisioning of the user data will be suspended on the Secondary CWM, but user data provisioning will continue on the Primary CWM, as user data provisioning will continue on the Primary CWM. During that time, the provisioning of user data and monitoring of the network are not impacted. This is called the Degraded Mode of Operation.

Release 12 of Cisco WAN Manager allows for better handling of peer-to-peer communications failure scenarios. An automatic switchover feature (AutoSwitchOver) has been designed to help a Secondary CWM get out of the degraded mode of operation, without requiring a restart of the CWM core on either the Primary CWM or Secondary CWM. In most cases, a Secondary CWM can get out of the degraded mode of operation automatically within a limited period of time.

In addition to the AutoSwitchOver feature, peer-to-peer communications enhancements include a bidirectional heartbeat feature which allows for better efficiency in the sending and checking of heartbeat messages, and also makes it possible for a Primary CWM to detect and report the connectivity status of Secondary CWMs. This enhanced communication between the Primary CWM and Secondary CWM aids in avoiding a degraded mode of operation. Also, a more reliable mechanism for reporting communication problems through SNMP traps generation for critical CWM-CWM events can expedite users reaction to the problems that could affect peer-to-peer communications.



---

**Note** Peer-to-Peer communications does not affect network sync-up.

---

## CWM Domain

A CWM domain consists of all CWM workstations in a network that are in communication with each other. Each CWM workstation functions as a *CWM gateway*, however one CWM workstation is designated as the *Primary CWM* for keeping the user data, and is the source of all user data that the Secondary CWMs can sync with.

You can define a CWM domain by specifying a list of CWM workstations that will communicate with each other as described in the “[Configuring Peer-to-Peer Communication](#)” section. All CWM workstations in a given domain must be connected to the same set of networks to ensure that their databases remain consistent.




---

**Note** The **network.conf** file does not have to be the same on all CWM stations in the domain as long as the gateway nodes specified in this file are part of the same network. Even though CWMs that manage the same network can talk to each other, managing the same network does not require having identical **network.conf** files on all Secondary and Primary CWMs.

---

## CWMGateway Process

The CWMGateway process provides a communications gateway for CWM workstations. Processes owned by one CWM workstation can communicate with processes owned by another CWM workstation using the CWMGateway process.

CWM workstations within the same CWM domain communicate with other CWM workstations by transferring information between the CWMGateway processes of each CWM workstation. The communication between CWM workstations is set up transparently by the CWMGateway processes in each CWM workstation.

CWM core processes communicate with the CWMGateway process of the CWM workstation to request information from and to send information to another CWM workstation.

## CWMGateway Functions

With the use of the CWMGateway Process, CWM to CWM communications allows CWM to have the following functionality:

- CWM can determine, without the help of the managed network, the presence of other managing CWM servers. This is achieved by having IP connectivity between all CWM workstations.




---

**Note** Loss of IP connectivity means that the Primary and Secondary CWMs are not able to communicate through sending or receiving network information, and are not able to ping each other.

---

- CWM workstations can communicate with other (remote) CWM workstations.
- If a Primary CWM workstation were to be shut down, another CWM workstation would become the new Primary CWM.
- If failure occurs, such as a loss of communication, it can be detected and recovered predictably and reliably.

Apart from the redundancy aspect, one additional benefit of the CWM gateway is the ability for multiple CWM workstations to share User Data, as well as maintain synchronization with the network. The CWM Gateway process maintains consistency of user data across the CWM domain, while the proprietary Robust Trap mechanism and SNMP maintain the CWM database consistent with the network data.

*Network Data* is defined as data that originates in the network and is communicated to the CWM workstation(s) by means of the proprietary Robust Trap mechanism or by SNMP, depending on the network element concerned. An example of network data is a change in alarm status for a user port or access line.

*User Data* is information that is supplied by a CWM user, or by an external OSS, and which cannot be stored in a network element and was therefore not visible to other CWM workstations prior to Release 10.4. Examples of User Data include connection templates, Service Class Templates, SNMP community strings, and the unique node IDs generated by CWM during the network discovery process.

With the CWM gateway function, User Data is propagated between the CWM workstations in a domain, thereby maintaining consistency. The Primary CWM acts as an arbitrator to prevent contention between CWM workstations for the same element of User Data. This means, for example, that if a user on one CWM workstation wants to modify a particular connection template, the Connection Template Manager process on that CWM workstation must request the Primary CWM workstation to lock that resource to prevent concurrent modification by two users.

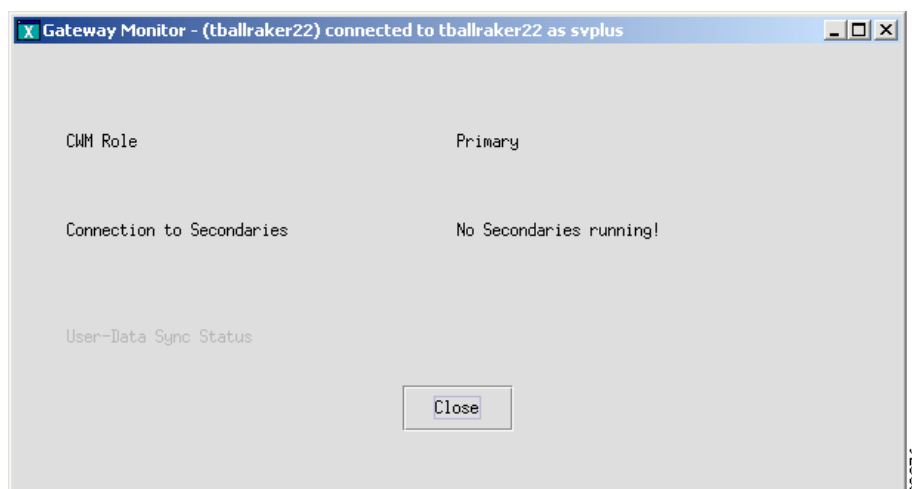
## Displaying the CWMGateway Monitor

You can display all the CWM workstations in the domain, role, and user\_table DSSync Status.

To display the gateway monitor, choose **Apps > Gateway Monitor**.

[Figure A-1](#) displays the CWM role as primary.

**Figure A-1 Gateway Monitor**



## Establishing Primary CWM and Secondary CWM Priority

In a given wide area network managed by Release 12 of CWM, the first CWM workstation to begin operation will assume the role of Primary CWM. As other CWM workstations become active, they will take on Secondary CWM workstation roles. The only difference in function between Primary and Secondary CWM workstations is that the Primary CWM workstation would provide the Secondary CWM workstations with user data when the Secondary CWM workstation launches.

Priority numbers of all Secondaries are assigned by the Primary at the time a Secondary registers with the Primary. It is based on “first-come-first-serve” logic. All the Secondaries have the same privilege except that the Secondary with priority 1 will take over as the Primary if the Primary shuts down.

The priority numbers of Secondaries might change in a Secondary CWMGateway process’s lifecycle.

In the following cases, the priority numbers among Secondary CWMGateways might change randomly:

- Restart the Primary CWMGateway (by watchdog)
- SwitchOver due to the Primary shutdown
- Primary ForceSwitchOver to a Secondary

This is the result of re-registration with the new Primary CWMGateway process. Priority numbers will be re-assigned by the new Primary CWMGateway process when a Secondary registers with the new Primary. In other words, for whatever reason the Primary CWMGateway process changes, the priority numbers of the Secondaries will be subject to re-assignment.

It is possible that an S2 (Secondary with Priority 2) can become an S1 after the Primary CWMGateway is restarted (by watchdog). It is also possible that an S3 can become an S1 after the switchover.

In the following cases, the priority numbers among Secondary CWMGateways will NOT change randomly:

- Restart a Secondary (by watchdog)
- Shutdown Secondary

In these cases, if Sn (Secondary with priority n) is restarted or shutdown, any Sm (m>n) will become S(m-1), while any Sm (m<n) will remain as Sm.

The difference between these scenarios is that in a Shutdown Secondary case, the Primary CWMGateway process did not change.

## Re-establishing Peer-to-Peer Communication Priority

This section describes how to re-establish peer-to-peer communication priority.

### Primary CWM Graceful Shutdown

The first CWM to be launched assumes the role of the primary by default. Subsequently, other CWMs launched will register with the primary CWM and each Secondary CWM is assigned a priority number which identifies the order in which the Secondary CWM was launched with respect to the primary CWM. Before being shutdown gracefully by the user, the CWMGateway on the primary CWM will invoke the IDL interface (used to communicate between two CWMGateway processes), of each of the objects corresponding to a Secondary CWMGateway. This will notify the Secondary CWM that the primary CWM is about to go away and that it will in turn nominate the second CWM in line (priority 1)

to take over as the new primary. The Secondary CWMGateway with priority number 1 (SP1) will now take over as the new Primary, and any remaining CWMGateways will subsequently register with the new Primary CWM. The new CWMGateway priority numbers will be based on “first-come-first-serve” logic.



**Note** Re-sync between the Secondary CWM and Primary CWM does not depend on how fast the Primary CWM can re-sync with the network after restart. If the Primary CWM has previously discovered all of the nodes in the network, and populated its node\_info table with all network node data, it is not necessary for the Secondary CWMs to wait for the Primary CWM to re-sync with the entire network. This capability gives Secondary CWMs in a network the ability to access user data much faster than if they had to wait for the Primary CWM to re-sync with the entire network.

## Secondary CWM Graceful Shutdown

A Secondary CWMGateway will notify its primary counterpart before it’s shutdown gracefully. The primary CWMGateway will assign a new priority number to the remaining Secondary CWMGateway whose priority number is greater than the priority number of the CWMGateway that has just gone away. The new priority number is one less than the previously assigned priority number. For example, if the CWMGateway with priority number 1 has gone away, the CWMGateway with priority number 2 will be changed to 1 so that it becomes next in line to take over the role of the primary CWM when the current primary CWM goes away.



**Note** An *Abnormal Exit* happens when a CWM Gateway process is stopped through a non-graceful shutdown. This includes when a process has been stopped with a **process core dump**, but does not include a power down.

## Configuring Peer-to-Peer Communication

When a CWM workstation is launched, it reads a configuration file called CWMGateway.conf to determine its initial setup and default configurations. Following is an example of a CWMGateway.conf file with a DomainGatewayList of a domain consisting of four CWM workstations named cwmws1, cwmws2, cwmws3, and cwmws4:

```
DomainGatewayList cwmws1 cwmws2 cwmws3 cwmws4
ForcedSwitchOver cwmws2
HeartBeatInterval 20
SP1_AutoSwitchOver ON
PrimaryRestartInterval 15
```

Parameters in this CWMGateway.conf file must be specified to enable inter-CWM communication; it is the only list of other CWM gateways available to this local host. Each host or CWM workstation listed in the DomainGatewayList must be *reachable* from the local host. Confirm this by using the UNIX “ping” command. The host names listed in the DomainGatewayList can be presented in any order.

The CWMGateway.conf file in /usr/users/svplus/config/ specifies the following parameters:

- **DomainGatewayList**—a list of remote gateways or the other CWM workstations that are part of this CWM domain. These workstations are either running Release 12 of CWM or will be at some point in the future. There must be IP connectivity between all of the CWM workstations listed. The DomainGatewayList parameter lists the CWM workstation hostnames that are part of a given domain and enables communication between these workstations. It is required that all CWM workstations in a domain have the same DomainGatewayList in their CWMGateway.conf file. If a given CWM workstation is the only one managing a network, you do not have to specify this parameter.



**Note** The **network.conf** file does not have to be the same on all CWM stations in the domain as long as the gateway nodes specified in this file are part of the same network. Even though CWMs that manage the same network can talk to each other, managing the same network does not require having identical **network.conf** files on all Secondary and Primary CWMs.

- **Forced Switchover**—tells a CWM workstation (if it is Primary) to hand over that role to another CWM workstation and become a Secondary CWM workstation. The ForcedSwitchOver parameter indicates a Secondary CWM gateway that assumes the role of Primary CWM gateway when the Primary releases that role. This parameter is only used by the Primary CWM gateway. The Forced Switchover hostname field is empty by default.

Three conditions are required for a forced switchover to take effect:

- The local CWM gateway must be running as the Primary
- The remote CWM gateway designated by the ForcedSwitchOver parameter must be up and running
- The remote CWM gateway designated by the ForcedSwitchOver parameter must have IP connectivity between the Primary CWM gateway and the Secondary CWM gateway

This option is not read at startup or in response to an HUP or USR1 signal, but processed only in response to a USR2 signal.

To initiate a forced switchover follow these steps:

- Verify that the nominated Secondary host is in the same domain as the Primary host and that CWM is up and running
  - Edit the CWMGateway.conf file manually (with vi or another editor) using the **ForcedSwitchOver cwmwsx** command, and set the host name of the nominated Secondary to the ForcedSwitchOver on the Primary
  - Retrieve the process id (pid) of the CWMGateway on the Primary and send a USR2 signal to the CWM gateway on the Primary
- **Heartbeat Interval**—tells the CWM gateway how often to send the heartbeat signal. Values for the Heartbeat Interval must be the same among all CWM workstations in the same domain. The Heartbeat Interval indicates the interval at which the Primary CWM will send the heartbeat signal to the Secondary CWM. If a Secondary CWM fails to detect two consecutive heartbeat signals, it assumes a loss of connectivity with the Primary CWM. There will be no switchover in this situation. The Secondary CWM will log an L1 message, and print an error message on the console, indicating that it has lost connectivity to the Primary CWM. This will be repeated every 60 seconds. It will go back to the normal mode of operation once the heartbeat message has been received. The Secondary CWM will work in a degraded mode of operation until, and unless, the heartbeat is restored with the Primary CWM.

- **SP1\_AutoSwitchOver ON**—the "ON" is a default value which indicates that the Automatic Switch Over feature is enabled. Reconfiguring this value to "OFF" will disable the Automatic Switch Over feature. Release 12 of Cisco WAN Manager allows for better handling of peer-to-peer communications failure scenarios through automatic switchover. An automatic switchover feature (AutoSwitchOver) has been designed to help a Secondary CWM get out of the degraded mode of operation, without requiring a restart of the CWM core on either the Primary CWM or Secondary CWM. In most cases, a Secondary CWM can get out of the degraded mode of operation automatically within a limited period of time.
- **PrimaryRestartInterval 15**—the Primary CWM restart interval value is set to 15 seconds by default.

## Degrade Mode

The user is able to continue the provisioning of network data, even when communications between a Primary CWM and Secondary CWM have been interrupted. If for any reason the communications between CWM servers are interrupted, user data provisioning will be suspended on the Secondary CWM, but user data provisioning will continue on the Primary CWM. During that time, the provisioning of user data and monitoring of the network are not impacted. This is called the Degraded Mode of Operation.

In order to provide peer-to-peer communications, a CWM workstation must be able to determine, transparently to the network, if another CWM workstation is currently running in the network. This requires IP connectivity between all CWM workstations. If all Secondary CWMs have lost IP connectivity with the Primary CWM, then all Secondary CWMs will function in the degraded mode of operation.

Degrade mode is defined by a loss of connectivity between the Primary CWM and any Secondary CWMs, in which all Secondary CWMs are unable to provision user data, including the adding, deleting, or modifying of user data, while waiting for a connection with the Primary CWM to be restored. In the degrade mode a Secondary CWM can still manage the network data (not user data), and provisioning of network data can still proceed.

Release 12 of Cisco WAN Manager allows for better handling of peer-to-peer communications failure scenarios. An automatic switchover feature (AutoSwitchOver) has been designed to help a Secondary CWM get out of the degraded mode of operation, without requiring a restart of the CWM core on either the Primary CWM or Secondary CWM. In most cases, a Secondary CWM can get out of the degraded mode of operation automatically within a limited period of time.

## Recovering From Degrade Mode

There is one case in which a CWM cannot get out of the degraded mode of operation with the AutoSwitchOver feature, and instead requires a forced switchover by the user. This happens when there is an interruption in communications between CWMs that may be due to a Primary CWM that has lost IP or physical connectivity with the only Secondary CWM in a domain.

If there is a problem with the Primary CWM, then the Secondary CWM will take over as the new Primary after the user does a forced switch over. The user will then need to fix the downed Primary CWM and re-establish IP connectivity.

## Primary CWM Has Lost IP Connectivity to the Only Secondary CWM in a Domain



### Note

In a situation where there is a Primary CWM and only one Secondary CWM in a given domain, and connectivity between the two CWMs is lost, the AutoSwitchOver feature will not work. In this case a manual forced switch over by the user is necessary.

- To recover from this degraded mode of operation, re-establish IP connectivity on the Primary CWM and the Secondary CWM will automatically re-sync user data tables with the Primary CWM.
- **If you can't re-establish IP connectivity on the Primary CWM**, then do a manual forced switch over on to a Secondary CWM. In this case the user must determine which CWM has the problem, and then from the working CWM initiate a manual forced switchover.

## Forced Switchover

Prior to Release 12 of CWM, a user could only do a forced switchover on a Primary CWM. If any Secondary CWMs went into the degraded mode of operation due to a disconnect with the Primary CWM, the Secondary CWM had to wait for the problem that caused the disconnect to be resolved first before getting out of the degraded mode of operation. If the connection problem couldn't be resolved promptly, the Secondary CWM had to be restarted in order to isolate the current Primary CWM.

Release 12 of CWM has improved the CWMGateway by providing the user a method to force a Secondary CWM's switchover to a Primary CWM without restarting the running CWM through this expanded forced switchover capability.



### Note

A forced switchover on a Secondary CWM requires that the user has a clear understanding about the architecture of peer-to-peer communication. Without following proper procedures, a user could break the basic rule of this type of network communication and introduce multiple Primary CWMs into a domain.

## Bidirectional Heartbeat Messages

A bidirectional heartbeat feature is part of the peer-to-peer communications enhancements for Release 12 of CWM which allows for better efficiency in the sending and checking of heartbeat messages. This feature also makes it possible for a Primary CWM to detect and report the connectivity status of Secondary CWMs. This enhanced communication between the Primary CWM and Secondary CWM aids in avoiding a degraded mode of operation. Also, a more reliable mechanism for reporting communication problems through SNMP traps generation for critical CWM-CWM events can expedite users reaction to the problems that could affect peer-to-peer communications.

Prior to Release 12 of CWM, the heartbeat messages were only sent from the Primary CWM to the Secondary CWM through a *oneway* CORBA interface. Release 12 of CWM provides for a *non-oneway* CORBA interface so that messages are now sent bidirectionally.

## Limitations for CWM to CWM Communications

- The Secondary CWMs have to wait for the Primary CWM to finish syncing up with the network. Trap 28075 (svDatabaseInSync) is sent when the Primary CWM has finished syncing up with the network.

**Note**

---

The **network.conf** file does not have to be the same on all CWM stations in the domain as long as the gateway nodes specified in this file are part of the same network. Even though CWMs that manage the same network can talk to each other, managing the same network does not require having identical **network.conf** files on all Secondary and Primary CWMs.

---

- The Configurator can only be run on the Primary CWM.

## Enabling CWM to CWM Communications

In Release 12, CWM provides a script to sync up a CWM database with another remote CWM workstation without running the two CWMs in primary-secondary mode.

**Note**

---

If you want to configure Statistics Collection Manager (SCM) in the gateway, refer to the *Cisco WAN Manager Installation Guide for Solaris 8, Release 12*.

---

The script synchronizes the following user-related tables in the Stratacom database:

- node\_info
- user\_info
- sec\_profile
- xpvc\_preferred
- xpvc
- xpvc\_segment
- sct
- sct\_cosb
- sct\_vc
- sct\_usage
- conn\_template
- conn\_tmpl\_param
- user\_conn\_desc

**Note**

---

The script will delete whatever existing data is in the tables on the local workstation. Do not expect to retain any existing data in the tables after running the script.

---

## Steps for Executing the `usertblDBsync` and `usertblDBcmp` Scripts

Execute the following steps to *copy* the remote table containing user data information to the database on the local machine by running the `usertblDBsync` script.

**Step 1** Execute the `usertblDBsync` script

```
% usertblDBsync <remote_CWM_workstation_name>
```

Example:

```
mмен% usertblDBsync mмен10
```



**Note** This will destroy all the data in the following tables from the local CWM Database and load the data from CWM on host mмен10.

- + node\_info
- + user\_info
- + sec\_profile
- + xpvc\_preferred
- + xpvc
- + xpvc\_segment
- + sct
- + sct\_cosb
- + sct\_vc
- + sct\_usage
- + conn\_template
- + conn\_tmpl\_param
- + user\_conn\_desc

```
Continue to sync? [No]y
```

```
*****Syncing user tables with CWM on host [cwmtopo62]*****
```

```
Syncing Table node_info@mменs12 with
node_info@cwmtopo62.....[DONE]
```

```
Syncing Table user_info@mменs12 with
user_info@cwmtopo62.....[DONE]
```

```
Syncing Table sec_profile@mменs12 with
sec_profile@cwmtopo62.....[DONE]
```

```
Syncing Table xpvc_preferred@mменs12 with
xpvc_preferred@cwmtopo62.....[DONE]
```

```
Syncing Table xpvc@mменs12 with
xpvc@cwmtopo62.....[DONE]
```

```
Syncing Table xpvc_segment@mменs12 with
```

```
xpvc_segment@cwmtopo62.....[DONE]
Syncing Table sct@mmendsl2 with
sct@cwmtopo62.....[DONE]
Syncing Table sct_cosb@mmendsl2 with
sct_cosb@cwmtopo62.....[DONE]
Syncing Table sct_vc@mmendsl2 with
sct_vc@cwmtopo62.....[DONE]
Syncing Table sct_usage@mmendsl2 with
sct_usage@cwmtopo62.....[DONE]
Syncing Table conn_template@mmendsl2 with
conn_template@cwmtopo62.....[DONE]
Syncing Table conn_tmpl_param@mmendsl2 with
conn_tmpl_param@cwmtopo62.....[DONE]
Syncing Table user_conn_desc@mmendsl2 with
user_conn_desc@cwmtopo62.....[DONE]
mmendsl2-11->
```

Executing this script copies the following user-related tables in the stratacom database from the remote CWM workstation specified by *<remote\_CWM\_workstation\_name>* to the local machine:

- node\_info
- user\_info
- sec\_profile
- xpvc\_preferred
- xpvc
- xpvc\_segment
- sct
- sct\_cosb
- sct\_vc
- sct\_usage
- conn\_template
- conn\_tmpl\_param
- user\_conn\_desc



---

**Note** Ensure that the networks in the network.conf file on the local machine are the same as those specified in the network.conf file on the remote CWM station, where CWM is already synced up. You will see a warning to this effect displayed on the screen after the tables have been successfully loaded.

---

Execute the following steps to *compare* the remote table containing user data information to the database on the local machine by running the **usertblDBcmp** script.

**Step 1** Execute the **usertblDBcmp** script

```
% usertblDBcmp <remote_CWM_workstation_name>
```

Example:

```
mnen% usertblDBcmp mmenu10
```

- + node\_info
- + user\_info
- + sec\_profile
- + xpvc\_preferred
- + xpvc
- + xpvc\_segment
- + sct
- + sct\_cosb
- + sct\_vc
- + sct\_usage
- + conn\_template
- + conn\_tmpl\_param
- + user\_conn\_desc

```
cwmtopo62-10-> usertblDBcmp mmen
```

```
*****Comparing user tables with CWM on host [mmen]*****
```

```
Comparing Table node_info@cwmtopo62 with  
node_info@mmen.....[SAME]
```

```
Comparing Table user_info@cwmtopo62 with  
user_info@mmen.....[SAME]
```

```
Comparing Table sec_profile@cwmtopo62 with  
sec_profile@mmen.....[SAME]
```

```
Comparing Table xpvc_preferred@cwmtopo62 with  
xpvc_preferred@mmen.....[SAME]
```

```
Comparing Table xpvc@cwmtopo62 with  
xpvc@mmen.....[SAME]
```

```
Comparing Table xpvc_segment@cwmtopo62 with  
xpvc_segment@mmen.....[SAME]
```

```
Comparing Table sct@cwmtopo62 with  
sct@mmen.....[SAME]
```

```
Comparing Table sct_cosb@cwmtopo62 with  
sct_cosb@mmen.....[SAME]
```

```
Comparing Table sct_vc@cwmtopo62 with
sct_vc@mmen.....[SAME]
Comparing Table sct_usage@cwmtopo62 with
sct_usage@mmen.....[SAME]
Comparing Table conn_template@cwmtopo62 with
conn_template@mmen.....[SAME]
Comparing Table conn_tmpl_param@cwmtopo62 with
conn_tmpl_param@mmen.....[SAME]
Comparing Table user_conn_desc@cwmtopo62 with
user_conn_desc@mmen.....[SAME]
cwmtopo62-10->
```

