ptc

integrity™ lifecycle manager

Server Failover Configuration

11.1

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

Contents

Integrity Lifecycle Manager Server Failover Configuration 7

- Before You Start 8
- Overview..........8
- Pre-requisites..........9
- Creating a Mount Start Up Script (Windows only) 12

Installation and Configuration 15

- Installing a Failover Configuration 16
- To install a standard failover configuration (Windows and Solaris)..........16
- To install a Sun Failover Configuration (Solaris) 19
- Defining a Cluster Name 22
- Defining Cluster Member Names 22
- Defining Bind Addresses for Members..........22
- Configuring Additional Properties Files 23
- Using an Oracle RAC Configuration 24
- Configuring the Layer 4 Switch 24
- Defining a Load-Balance Using a Cisco Layer 4 Switch 26

Diagnostics and Troubleshooting..........29

- Defining Log and Temp Directories..........30
- Configuring the Primary Server’s Heartbeat..........30
- Reconnecting After a Failover..........31
- Forcing a Failover 31
- Forcing a Shutdown..........31
- Forcing a Restart 32
- mks.logfile Parent Directory Not Created 32
- Troubleshooting..........32
- Getting Help 32

1

Integrity Lifecycle Manager Server Failover Configuration

Before You Start 8
Overview 8
Pre-requisites........ 9
Creating a Mount Start Up Script (Windows only) 12

For global organizations with hundreds or thousands of users working 24x7, it is imperative that key infrastructure systems support high levels of availability and not suffer from single points of failure. In the event of an unforeseen disaster, bringing mission-critical systems back online in the least amount of time is critical.

As a safeguard, PTC offers a failover configuration for the Integrity Lifecycle Manager server. In the event that your main Integrity Lifecycle Manager server fails (the server's operating system goes down, the network to the database is lost, or the network that the server is communicating over is lost), a secondary Integrity Lifecycle Manager server takes over and resumes server operations without manual intervention and minimal impact on users.

Note

As a best practice, PTC highly recommends that you also have a disaster recovery process in place, such as backing up the Integrity Lifecycle Manager server.

Before You Start

If you are upgrading from Integrity Lifecycle Manager 2009, remove the `failover-service.xml` file from the *`installdir`*`/server/mks/deploy` directory. Failure to remove the file prevents the server from starting.

If you remove the `failover-service.xml` file but do not set the `mksis.ha.type` property in the `is.properties` file, multiple servers will point to the same database, unaware that they are running in a failover configuration. This can lead to serious problems. For more information, see Defining a Cluster Name on page 22

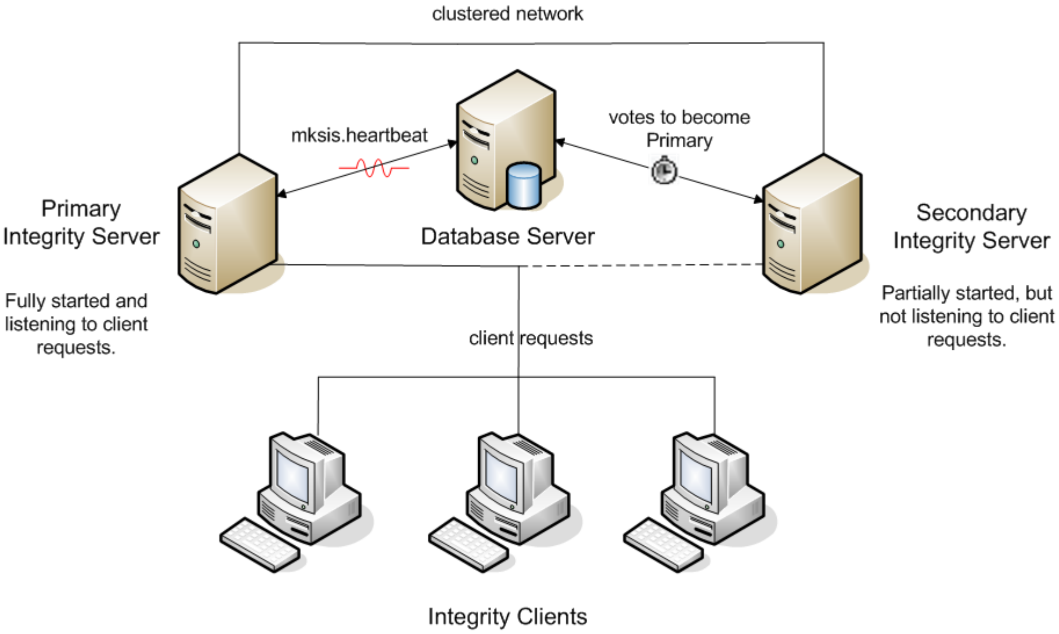
Overview

The Integrity Lifecycle Manager server failover configuration implements a *clustering framework*. Clustering uses a group of servers to transparently run an application as if it were a single entity. However, the failover configuration is actually a *warm failover solution*, not a *full clustering solution*. A future release of Integrity Lifecycle Manager is expected to implement full clustering for a specific set of databases and Integrity Lifecycle Manager functionality, replacing the warm failover solution.

Note

For licensing purposes, each node in a warm failover implementation requires its own Integrity Lifecycle Manager server license. For more information on licensing, see the *Integrity Lifecycle Manager Installation and Upgrading Guide*.

On initial startup, the failover configuration detects the servers that are in the cluster; each server in the cluster is defined as a *node*or *member*. For the purposes of illustration in this document, a main production server and a secondary production server are used. An election service uses the database server to vote on which server becomes the primary server (the active server used for production). The Integrity Lifecycle Manager server that wins the vote becomes the *primary server*, performing a complete start up and begins listening to client requests. The primary server then sends a *heartbeat* (specified by the `mksis.heartbeat`property) to the database server, indicating its status as the active primary server and that it is running incident free. The *secondary*server detects that it lost the vote. It suspends its start up process and does not listen to client requests. As long as the primary server sends a regular heartbeat to the database, the secondary server's start up process is suspended and continues to vote, attempting to become the primary server until the primary server fails.


clustered network
mksis.heartbeat
votes to become Primary
Primary Integrity Server
Fully started and listening to client requests.
Database Server
Secondary Integrity Server
Partially started, but not listening to client requests.
client requests
Integrity Clients

A failover configuration with a primary and secondary Integrity Lifecycle Manager server

When the primary server fails, the secondary server waits two heartbeats to give the primary server a chance to win the election, for example, the database connection may have been temporarily lost. If the secondary server wins the election, it becomes the new primary server and continues its start up process. When the former primary server detects that it lost the election, it shuts down, triggering a process that performs a restart, making it the secondary server.

Connected Integrity Lifecycle Manager client systems GUI clients transparently reconnect after the failover occurs. Intensive and long running commands (create Sandbox and resynchronize) include a **Continue After Recoverable Outages** option that is enabled by default, allowing clients to automatically reconnect and resume the operation after a failover occurs. A similar option is enabled by default from the CLI for `si createsandbox` and `si resync`. To configure the reconnect preferences for the GUI and CLI, see Reconnecting After a Failover on page 31

For clustered staging and production servers, a failover during an admin migration triggers a rollback.

Pre-requisites

To implement the Integrity Lifecycle Manager server failover configuration, the following pre-requisites are required:

- Database

A shared database repository that is Oracle, or Microsoft SQL. For an Oracle database, an Oracle RAC configuration is supported (for more information, see Using an Oracle RAC Configuration on page 24). The RCS repository is not supported.

- Integrity Lifecycle Manager server systems

  At least one secondary server must exist for each primary server. The servers must also reside on separate machines and use a shared database.

  Each Integrity Lifecycle Manager server must be at the same major build version (Integrity Lifecycle Manager server 2009 or greater). Service packs and HotFixes should indicate if they are incompatible. If you need to install a service pack, stop both servers, install the service pack, and then restart the servers.

  Configuration of the two servers must be identical, sharing configuration files on a shared file system.

Tip

As a security measure, you can hide the servers behind a firewall, exposing only the single IP/Port on the switch.

- Network Load Balancer

  A *network load balancer* (balancer) is responsible for making failover transparent to clients by automatically re-directing client requests to the active node. The balancer, placed between the client and server nodes:

  ○ Provides a single access point (IP address) to the multi-node failover configuration, where each node has a different IP address. Without the balancer, clients require manual configuration to connect to a different node in the event of failure.

  ○ Tracks the active node by periodically checking each node to see if the configured port is turned up (in listening state).

  ○ Transparently directs new and existing client connections to the active node.

  PTC recommends using a hardware load balancer for speed and reliability. A dedicated load balancer, such as the Cisco Content Switching Module (CSM) available for the 6500 series of Cisco switches, provides all of the features and flexibility required by failover configurations.

  A modern Layer-4 network switch, such as the Cisco CSS 11501, can also be configured to provide balancing between ports.

- Shared File System

  For ease of installation and maintenance of a failover configuration, a shared file system is used for the application, configuration, and data. The shared file system, such as a *Network Attached Storage* (NAS) device, requires the following:

  - Since the shared network drive that the shared file system resides on contains paths that may contain drive letters, the shared file system must be mounted using the same drive letter on each of the nodes.
  - All Integrity Lifecycle Manager server property files must be available to either Integrity Lifecycle Manager server.
  - The drive setup between the primary and secondary machines must be the same, and the shared file system must have the same name.

    Shared file systems implement one or more network file sharing protocols. The implemented protocol should be robust enough to protect against network outages. The following are the two most commonly used protocols (most NAS devices, such as the NetApp Filer, implement both):
  - *Network File System* (NFS)

    On UNIX, NFS is the default protocol and allows a hard mount. This means that in the case of a network outage to the file system, all file system requests are blocked until the outage is resolved and operations resume. PTC recommends implementing the NFS protocol.

    On Windows, an NFS client is required to access a NAS using the NFS protocol. Windows Vista and later OS‘ include an NFS client; however, you must enable the Services for NFS feature in the Control Panel/ Programs and Features/Windows features.
  - *Server Message Block* (SMB) / *Common Internet File System* (CIFS)

    On Windows, SMB is the default protocol and does not allow a hard mount. This means that in the case of a network outage to the file system, all file system requests become invalid, even if the outage is only temporary.

    Note

    PTC recommends that you only use the SMB protocol if you have high confidence in the network that each Integrity Lifecycle Manager server and the NAS device reside on. In the event of a [temporary] network outage with the SMB protocol, the Integrity Lifecycle Manager server may fail.

To start the primary server from the shared drive on Windows, another service needs to map the drive for the Integrity Lifecycle Manager server. To do this, you need to create a mount startup script. For more information, see Creating a Mount Start Up Script (Windows only) on page 12

On UNIX, there is no concept of a per-session mount. From a command line, use the `mount - o hard` command.

- Redundancy of System Components

  The failover configuration enables redundancy of the Integrity Lifecycle Manager server. To avoid a single point of failure, ensure that you enable redundancy for the following system components: FlexNet license server, database, and shared drive.

Creating a Mount Start Up Script (Windows only)

Note

For ease of installation and maintenance of failover configuration, a shared network drive is used for the application, configuration, and data. Since the shared configuration contains paths that may contain drive letters, the shared file system must be mounted using the same drive letter on each of the nodes.

To start the primary server from the shared drive on Windows, another service must map the drive for the Integrity Lifecycle Manager server. To do this, you create a mount start up script that:

- Mounts the drive with the Integrity Lifecycle Manager server installation to a drive letter.

  The Integrity Lifecycle Manager server must wait until the network redirectors are ready. Services are started in parallel with the network redirectors, so the network may not yet be available.

- Starts the Integrity Lifecycle Manager server Service.

  The Integrity Lifecycle Manager server is configured with a manual startup type, to delay the start up until the mount is complete.

To create a start up script

1. Create a `mountshares.bat` script, updating the mount line in the script as needed for the required permissions, for example:

```
echo %DATE% %TIME%
:loop
mount -o mtype=hard -o anon \\NAS\SHARENAME Z:
if not errorlevel 1 goto :done
echo Exit code: %ERRORLEVEL%
sleep 5
goto :loop
:done
net start "Integrity Server 10"
```

If you are using the Windows NFS Server and do not have an appropriate setup method for NFS UNIX Identity mapping, you can configure the NFS share to be valid for the cluster members only. For example:

```
nfsshare sharename=drive -o anon=yes -o rw=cluster-node1:cluster-node2
```

Only the specified nodes can mount the NFS share; however, with anonymous access permissions.

2. Start the Microsoft Management Console by typing the following from the **Run** prompt:

   ```
   mmc
   ```

3. In the console, select **File ▸ Add/Remove Snap-in**.
4. Click **Add**, and then select **Group Policy Object Editor** from the list.
5. Click **Add**.
6. Click **Finish**.
7. From Windows Explorer, copy `mountshares.bat` to the following directory (the `GroupPolicy` directory may be hidden):

   ```
   c:\windows\system32\GroupPolicy\Machine\Scripts\
   Startup
   ```

8. In the Microsoft Management Console, you should see the **Local Computer Policy** tree. Expand the **Computer Configuration** and the **Windows Settings** node below it.
9. Under **Windows Settings**, select **Scripts (Startup/Shutdown)**, and then double-click **Startup** in the right panel.
10. In the **Startup Properties** dialog box, click **Add**.
11. In the **Script Name** field, browse to the `mountshares.bat` script you created, and then add `>mountshare.log` to the **Script Parameters** field. The output from running the `mountshares.bat` script appears in the `mountshares.log`.
12. Click **OK** and close the Microsoft Management Console.

The next time you start Windows, the `mountshares.bat` script runs before any logins or services start.

2

Installation and Configuration

Installing a Failover Configuration 16
To install a standard failover configuration (Windows and Solaris) 16
To install a Sun Failover Configuration (Solaris) 19
Defining a Cluster Name 22
Defining Cluster Member Names 22
Defining Bind Addresses for Members 22
Configuring Additional Properties Files 23
Using an Oracle RAC Configuration 24
Configuring the Layer 4 Switch 24
Defining a Load-Balance Using a Cisco Layer 4 Switch 26

This chapter covers the installation and configuration of a failover mechanism, as well as the layer 4 switch.

Installing a Failover Configuration

Choose an installation type:

- To install a standard failover configuration (Windows and Solaris) on page 16
- To install a Sun Failover Configuration (Solaris) on page 19

To install a standard failover configuration (Windows and Solaris)

1. Set up the hardware and ensure connectivity.
2. Configure the shared file system and ensure that it is accessible to every node.
3. Configure host names for each node (see Defining Cluster Member Names on page 22), and a cluster host name (see Defining a Cluster Name on page 22) so that every node can resolve each host name.

   Tip

   To avoid confusion, the node's host name should contain the node's member name.

   By default, the failover configuration binds the Integrity Lifecycle Manager server ports to local addresses. If the server machine contains multiple network interfaces, you can optionally choose to bind the ports to a specific address on each server. For more information, see Defining Bind Addresses for Members on page 22
4. On the shared file system, create a base directory for the Integrity Lifecycle Manager server installation, for example, `/shared/mks/servers`.
5. Install the Integrity Lifecycle Manager server in the specified base directory (on one node).
6. Make the necessary changes to security and configuration properties as per a normal standalone Integrity Lifecycle Manager server installation.
7. To ensure a successful installation and configuration, start the Integrity Lifecycle Manager server as a normal standalone server. Verify basic functionality and stop the server.
8. In *`installdir`*`/config/properties/is.properties`, set the following properties:
   - `mksis.ha.type=failover`

Note

Any other value causes the server to run in a normal configuration.

If you are upgrading from Integrity Lifecycle Manager 2009, manually add the property.

Caution

If you remove the `failover-service.xml` file but do not set the `mksis.ha.type` property, multiple servers will point to the same database, unaware that they are running in a failover configuration. This can lead to serious problems.

- `mksis.hostname=`*`host name assigned to the cluster`*

  This is the host name that clients use to connect to the cluster.

- `mksis.clear.port` and/or `mksis.secure.port=`*`chosen port for your service`*

9. If needed, on each node create a directory for storing bulk data on the local file system.
10. Select a member name (Node ID) for each node. Typically, nodes names are numbers (1, 2, 3, …); however, any short string may be used as long as it is unique to other nodes.
11. In *`installdir`*`/config/properties/im.properties`, add the following bulk cache property for each node:

    ```
    im.servercache.ha.member name.bulkrootdir=local bulk data directory
    ```

12. Determine if additional properties need to be configured. For more information, see Defining a Cluster Name on page 22
13. To start and stop the Integrity Lifecycle Manager server on each of the nodes, install the application service. To do this, you create a copy of the Integrity Lifecycle Manager server startup script and customize it for each node by setting a unique member name.

    a. Browse to *`server install dir`*`/bin`.

Note

On Windows, use `mksis.bat` to remove the Integrity Lifecycle Manager Windows service created during the installation. It is replaced with a member specific service

b. Create a copy of the `mksis.bat` (Windows) or `mksis` (Solaris) file for each node, specifying the member name as part of the file name. For example, `mksis-1(.bat)` and `mksis- 2(.bat).`

c. In each copy of the startup script, uncomment and set the `MEMBER_NAME` property to the appropriate member name. For example, in `mksis-1(.bat)` set `MEMBER_NAME=1`.

d. Create log and temp directories for each member. The log and temp directories allow you to distinguish diagnostic messages for each server. For more information, see Defining Log and Temp Directories on page 30

e. On Windows, log in to each node and use the appropriate customized `mksis.bat` file to install the service. For example, on member 1, run `mksis-1.bat` to install the service.

   On Solaris, log in to each node and copy the appropriate customized `mksis` file to the proper `rc` directory and set up any necessary soft links. For example, on member 1, copy the `mksis-1` file.

14. Configure the load balancer. For detailed information, see Configuring the Layer 4 Switch on page 24. In general, the balancer requires the following settings:

    - Hostname set to the cluster host name.
    - Listen ports configured to the selected clear and secure ports, as well as any other ports (HLL adapter port, etc.).
    - Each node's host name and ports configured as a destination. The node's port should match the cluster port.
    - *Sticky sessions.* This means that requests from clients coming from the same IP address should be directed to the same cluster node.
    - Timeouts disabled. Some balancers can timeout connections after inactivity; however, this functionality should be disabled.

- Configure probing to check if the server is up and active. Most balancers can verify this with port status.
- Configure balancing algorithm. For simple deployments, an algorithm that chooses the next node on the list is sufficient.

15. Start the cluster by doing the following:
    - Start the balancer.
    - Log in to each node and run the appropriate customized startup script with the start parameter. For example, on node 1, type the following from a command line:

      ```
      mksis-1 start
      ```

Tip

On Windows, you can use the Windows Service Manager to start each node.

To install a Sun Failover Configuration (Solaris)

1. Set up the hardware, ensure connectivity, and install Sun Cluster software.
2. Make sure the Sun cluster `bin` directory is part of the system `PATH`.
3. Use Sun clustering to configure a global file system that is available to all nodes.
4. Configure host names for each node (see Defining Cluster Member Names on page 22), and a cluster host name (see Defining a Cluster Name on page 22) so that every node can resolve each host name.

Tip

To avoid confusion, the node's host name should contain the node's member name.

By default, the failover configuration binds the Integrity Lifecycle Manager server ports to local addresses. If the server machine contains multiple network interfaces, you can optionally choose to bind the ports to a specific address on each server. For more information, see Defining Bind Addresses for Members on page 22.

5. Create a base directory for the Integrity Lifecycle Manager server installation on the global file system, for example, `/global/mks/servers`.
6. Install the Integrity Lifecycle Manager server by running `mksserver.bin` and following the installation steps. Ensure that you change the installation directory to the one specified on the global file system.
7. Make the necessary changes to security and configuration properties as per a normal standalone Integrity Lifecycle Manager server installation.
8. To ensure a successful installation and configuration, start the Integrity Lifecycle Manager server as a normal standalone server. Verify basic functionality and stop the server.
9. In *`installdir`*`/config/properties/is.properties`, set the following properties

   - `mksis.ha.type=failover`

     Note

     Any other value causes the server to run in a normal configuration.

     If you are upgrading from Integrity Lifecycle Manager 2009, manually add the property.

     Caution

     If you remove the `failover-service.xml` file but do not set the `mksis.ha.type` property, multiple servers will point to the same database, unaware that they are running in a failover configuration. This can lead to serious problems.

   - `mksis.hostname=`*`host name assigned to the cluster`*

     This is the host name that clients use to connect to the cluster.

   - `mksis.clear.port` and/or `mksis.secure.port=`*chosen port for your service*

10. For each node, obtain and record the local node ID by typing the following from a command line:

```
scha_cluster_get -O NODEID_LOCAL
```

11. On each node, create a directory for storing bulk data on the local file system.
12. In *installdir*/config/properties/im.properties, add the following bulk cache property for each node:

    im.servercache.ha.*node ID*.bulkrootdir=*local bulk data directory* (on the specific node)
13. One one of the nodes, change to the Integrity Lifecycle Manager server installation directory on the global file system.
14. Run the scdsbuilder command and set the following information in the Sun Cluster Agent Builder:
    - **Vendor Name:** MKS
    - **Application Name:** Integrity
    - **Working Directory:** *server install dir*
    - **Type:** Failover
    - **Network Aware:** checked
    - **Type of RT:** GDS
15. Click **Create**.
16. After the creation of the resource type, click **Next**.
17. Set the following commands:
    - **Start:** *server install dir*/bin/mksis start
    - **Stop:** *server install dir*/bin/mksis stop
18. Click **Configure**.
19. An MKSIntegrity directory now exists in your server installation directory, containing the configured agent.
20. On each node:
    - Change to the *server install dir*/MKSIntegrity/pkg directory.
    - Add the package to each system: pkgadd -d . MKSIntegrity.
21. Each node should have an /opt/MKSIntegrity package. The package contains commands to start, stop and remove the cluster.
22. Create log and temp directories for each member. The log and temp directories allow you to distinguish diagnostic messages for each server. For more information, see Defining Log and Temp Directories on page 30.
23. On one of the nodes, go to the /opt/MKSIntegrity/util and run the following command to configure and start the cluster:

```
./startIntegrity -h hostname -p port/type list -l Lb_sticky
```

where:

- hostname is the host name that becomes the shared address of the cluster.
- port/type is the selected port (mksis.clear.port) and type. If other ports are in use (secure, HLL) they must be specified, for example, "7001/tcp,7011/tcp,6667/tcp".

Defining a Cluster Name

If you have a large number of servers running in the same network segment, for example, a cluster of staging servers and production servers, PTC recommends specifying a customized partition name to distinguish the different clusters and avoid accidentally joining a logically distinct cluster that has the default cluster name.

To change the default cluster partition name, add the following line at the end of the mks.java.additional section in *installdir*/config/mksservice.conf:

```
mks.java.additional.##=-Djboss.partition.name=cluster name
```

Defining Cluster Member Names

On the primary and secondary servers, the service parameters must be modified to set the MEMBER_NAME variable to the name appropriate for the server. This defines a unique name for each member in the cluster and allows you to differentiate member names in diagnostic messages. For example, the primary server could be named mks-1 and the secondary server could be called mks-2.

Edit *installdir*/config/mksservice.conf, adding %MEMBER_NAME% for each member node.

Defining Bind Addresses for Members

By default, the failover configuration binds the Integrity Lifecycle Manager server ports to local addresses. If the server machine contains multiple network interfaces, you can optionally choose to bind the ports to a specific address on each server.

To define the bind address for a member

1. Add the following line at the end of the mks.java.additional section in *installdir*//config/mksservice.conf:

```
mks.java.additional.##=-Dmksis.bindAddr=%MEMBER_ADDR%
```

2. The service parameters on the primary and secondary machines must be modified to include the `MEMBER_NAME` and `MEMBER_ADDR` variables. For example, the primary server should include the following executable line:

   ```
   ..../mksservice.exe -s .../config/mksservice.conf
   ```

   Modify the executable line to include the member name and bind address, for example:

   ```
   ..../mksservice.exe -s .../config/mksservice.conf
   "set.MEMBER_NAME=secondary"
    "set.MEMBER_ADDR=1.0.0.2"
   ```

3. On the secondary server, modify the service parameters, for example:

   ```
   ..../mksservice.exe -s .../config/mksservice.conf
   "set.MEMBER_NAME=secondary"
    "set.MEMBER_ADDR=1.0.0.2"
   ```

Configuring Additional Properties Files

Additional configuration properties can point to files external to the Integrity Lifecycle Manager server root directory. If you need to configure them, ensure that they are also on the shared drive. For example, point the following property files to the directories:

- *installdir*/config/properties/si.properties:

  ```
  si.Cache.default.bulkRootDir=/shared/server/mks/bulk
  ```

- shared/data/public_html
- In the Integrity Lifecycle Manager administration client, under the **Configuration ▸ Properties** node set the following properties:

  ```
  triggers.environmentVariables=/shared/server/data/triggers/env.properties

  si.triggers.events=/shared/server/data/triggers/events

  triggers.scripts=/shared/server/data/triggers/scripts
  ```

- If you are using SSL, `tls` directory.
- All members of the cluster must use the same host name when identifying themselves to Integrity Lifecycle Manager client systems and each Integrity Lifecycle Manager server they encounter.

  Set the following properties in *installdir*/config/properties/is.properties:

```
mksis.hostname=myname

mksis.hostnameuseip=false
```

Using an Oracle RAC Configuration

To use the failover configuration with an Oracle RAC configuration, you must also modify the JDBC connection string in `installdir/config/properties/is.properties`. For example, if you have two nodes in an Oracle RAC configuration:

```
url=jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on)(FAILOVER=on)(ADDRESS=(P
ROTOCOL=TCP)(HOST=node1)(PORT=port1))(ADDRESS=(PROTOCOL=TCP)(HOST=node2)
(PORT=port2))(CONNECT_DATA=(SERVICE_NAME=service name))),\
```

where

- *node1* and *port1* are the name and port number of the first node in the Oracle RAC configuration.
- *node2* and *port2* are the name and port number of the second node in the Oracle RAC configuration.
- *service name* is the name of the Oracle RAC service.

Note

A known issue (Oracle Metalink note 364855.1) in the Oracle Net Configuration Assistant causes a configuration error when setting up a default Oracle RAC configuration. This known issue causes triggers and commands to fail. To resolve the Oracle issue, refer to the example described at:

http://www.ardentperf.com/2007/04/02/local_listener-and-ora-12545/

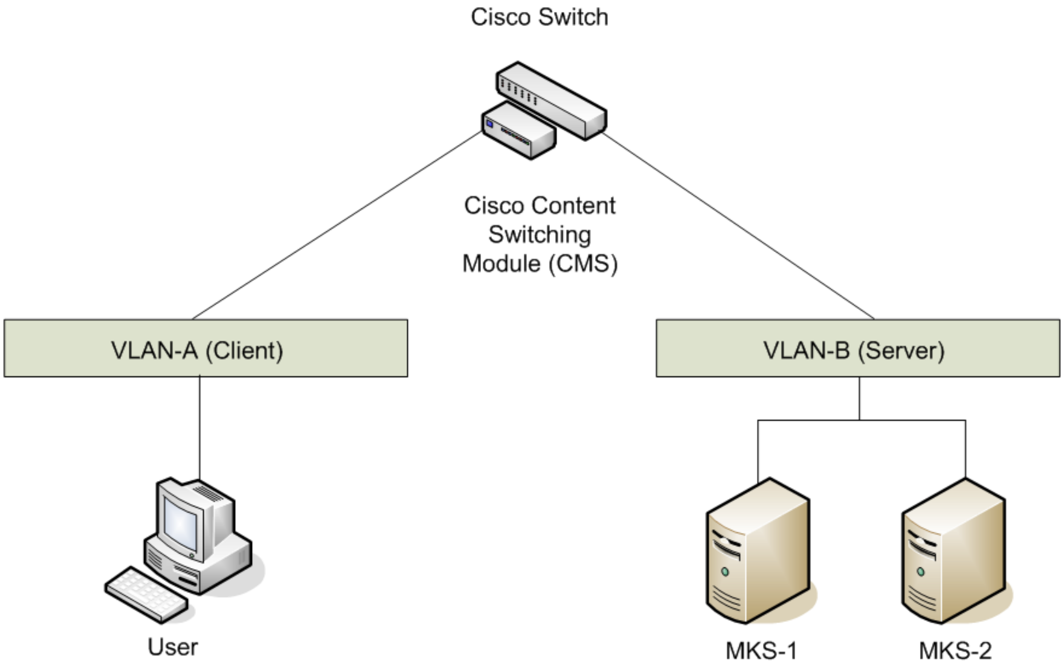
Configuring the Layer 4 Switch

To set up a front-end application switch, you must have hardware that is capable of defining a pool of resources (servers with an application that responds to a specific port). The pool is assigned a single virtual IP address (VIP), which is used by the application users to access the pool. The switch continually probes the pool to determine which servers are available and redirects requests from the VIP address to an available server in the pool. The switch software is also able to load-balance the pool, for example, using the least used connections. In the case of a failover configuration, only one server is recognized by the switch to be available at any one time, directing all traffic to it until a failover occurs.

For a Cisco Layer 4 switch, the load-balance is defined using Virtual Local Area Networks (VLANs). Most load-balancing general purpose switches use VLANs because they allow individual ports of a multi-port switch to be assigned specific subnets, which can be used to pool the appropriate resources.

The diagram below illustrates a Cisco-Bridged VLAN, using a Cisco Layer 4 switch and a Cisco Content Switching (CSM) module. You can implement any switch that has some form of application load-balancing feature; however, the following is mandatory:

- the switch can define an address which is re-directed to a pool of servers (VIP address)
- the switch is able to probe the pool and determine which servers are able to respond so that it will not re-direct traffic to an inactive server



Cisco-Bridged Virtual Local Area Network

PTC assumes that you have your own load-balancing solution and understand how to use it. You can accomplish the desired configuration using switches, servers with load-balancing software, and dedicated load-balancing equipment which combine both or any combination of the above. You should understand if it is necessary to define VLANs in your environment and know the specific IP ranges to use.

For more information on configuring a Cisco CSM module, browse to:

http://www.cisco.com/en/US/products/hw/modules/ps2706/ps780/tsd_products_support_model_home.html

Defining a Load-Balance Using a Cisco Layer 4 Switch

Note

Using the above diagram as a reference, the following is a basic example of defining a load-balance. It does not define a probe (which is possible for some switches). It does not define what, if any, load balancing is to occur (in general, the next node on the list is selected, if left unspecified). The example defines two servers and one pool, marking them active (available).

PTC assumes that some ports of the switch already exist on the client VLAN (`VLAN-A`); therefore, interfaces assigned to the client VLAN are omitted in the diagram.

For example, `VLAN-A` has an address of `10.0.0.0/20`; however, it can be any valid network definition defined on any VLAN

Individual interfaces are assigned to the server VLAN (`VLAN-B`). For example, using a Cisco 11501 switch, ports e1 and e2 are assigned to `VLAN-B`, which has a network range of `10.10.0.0/24`. The range can be anything; however, if it contains only a few addresses, it is likely smaller than a full 255 addresses (`/24`).

The VLANs are assigned addresses to enable the switch to handle the routing. For example, `VLAN-A` is assigned 10.0.5.44, and `VLAN-B` is assigned `10.10.0.1`. VLANs are individual segments within a switch, and the addresses are used as the gateway addresses to each segment.

Next, define each Integrity Lifecycle Manager server in the VLAN. The IP address and the ports used by each Integrity Lifecycle Manager server (`MKS-1` and `MKS-2`) belong to the server VLAN (`VLAN-B`), for example, `10.10.0.x`. The port is the port used by the Integrity Lifecycle Manager process.

The next step is to define the pool. You define the address used by the switch to direct traffic to the pool, the port being redirected, and the resources that belong to the pool. For example, the address is `10.0.3.43`; however, it can be anything visible to the client network, routable by the switch. In the simplest configuration, it is an address on the same VLAN as the client machines. The port is `7001`, the same as the one used by the Integrity Lifecycle Manager server systems.

After the load balance is defined, the client requests a connection to the VIP address of the pool. This is made visible by the switch as soon as the pool is marked active. VIP is defined on `VLAN-A`, the users are on `VLAN-A`, and the switch has an address that belongs to `VLAN-A`, which allows routing to occur without the need for explicit routes.

The packets arrive at the switch and are handled by the load-balancer. The balancer continually polls the address and port of the defined services within the pool to determine if they are active. In a failover configuration, only one server responds to queries on that port so that server is the only one logged as active by the switch. The packets are directed to the available server on `VLAN-B`. The connection is generally handled by network address translation (NAT), so the responses to the user appear to come from the VIP address.

3

Diagnostics and Troubleshooting

Defining Log and Temp Directories 30
Configuring the Primary Server’s Heartbeat...30
Reconnecting After a Failover 31
Forcing a Failover...31
Forcing a Shutdown...31
Forcing a Restart..32
mks.logfile Parent Directory Not Created..32
Troubleshooting 32
Getting Help...32

Defining Log and Temp Directories

To distinguish diagnostic messages for members in the cluster, you must enter variables for the member names. This moves the location of the log and temp directories, storing log files and temporary files in directories with suffixes based on member names, for example, the primary server could contain `log.primary` and `tmp.primary`, and the secondary server could contain `log.secondary` and `tmp.secondary`.

1. Open *`installdir`*`/config/mksservice.conf` and change the following lines:

   ```
   mks.java.additional.##=-Djboss.server.log.dir=.../log

   mks.java.additional.##=-Djboss.server.temp.dir=.../mks/tmp
   ```

   to

   ```
   mks.java.additional.##=-Djboss.server.log.dir=.../log.%MEMBER_NAME%

   mks.java.additional.##=-Djboss.server.temp.dir=.../mks/tmp.%MEMBER_NAME%
   ```

2. To separate the log files for each member in the cluster, modify the `mks.logfile`(controls `startup.log` location) and `mks.java.gcfile` (controls `gc.out` location) properties by adding the following line to the directory used for logging:

   ```
   .../log/... -> .../log.%MEMBER_NAME%/...
   ```

Configuring the Primary Server’s Heartbeat

To configure the primary server’s heartbeat, modify the `mksis.heartbeat` property in the Integrity Lifecycle Manager administration client. By default, the heartbeat is set to 20 seconds, with a minimum value of 5 seconds and a maximum value of 1000 seconds. For example, if you set the property to 20 seconds, the secondary server waits two 20 second intervals to give the primary server a chance to win the election.

Note

Increasing the `mksis.heartbeat` property can prevent unnecessary switch overs; however, a necessary switch over may take longer. Given that the length of garbage collection for servers may vary, PTC recommends modifying this property to suit your environment.

Reconnecting After a Failover

If a failover occurs while the create Sandbox or resynchronize commands are running, the GUI and CLI include options that specify whether the commands fail and automatically reconnect.

In the GUI, the **Continue After Recoverable Outages** option is enabled by default for the `Create Sandbox` and `Resynchronize` commands in the **Preferences Configuration** dialog box, and toggle.

In the CLI, the `--awaitServer` option is enabled by default for `si createsandbox` and `si resynchronize`. To prevent these commands from automatically reconnecting, specify `--noawaitServer`.

Forcing a Failover

To force an immediate failover of the primary server, type the following from the command line:

```
si/im diag --hostname=value --port=value --diag=failover
```

where `--hostname=`*`value`* and `--port=`*`value`* is the server information for the Layer 4 switch

The secondary server wins the election immediately, becoming the primary server and the old primary server restarts.

Forcing a Shutdown

To shutdown the primary server, type the following from the command line:

```
si/im diag --hostname=value --port=value --diag=shutdown
```

where `--hostname=`*`value`* and `--port=`*`value`* is the server information for the primary server.

Unlike a forced failover, the normal failover rule applies: a secondary must win the election after the appropriate time has elapsed.

Note

The failover only works if the machine is clustered, while a shutdown always works.

Both commands do not wait for anything in progress to complete, shutting down immediately.

Forcing a Restart

To restart the primary server, type the following from the command line:

```
si/im diag --hostname=value --port=value --diag=restart
```

where `--hostname=value` and `--port=value` is the server information for the primary server.

Both commands do not wait for anything in progress to complete, restarting immediately.

mks.logfile Parent Directory Not Created

During the initial start up of each server, the `mksservice.conf` does not create the parent `mks.logfile` directory containing the log file. The next time each server starts, the `mks.logfile` directory is automatically created for each member node.

Troubleshooting

If you are unable to diagnose any problems with your failover configuration, collect a support package and contact PTC Technical Support.

Getting Help

PTC Technical Support is focused on delivering the right solutions to issues as they arise. For assistance, you can choose online support or telephone a customer service representative. Online support provides easy access to e-mail, Web request services, automatic product notifications, and the PTC Integrity eSupport portal — a secure database that provides helpful resources such as product documentation, knowledge base articles, product downloads, user forums, presentations, and more. For online support, browse to http://support.ptc.com/support/integrity.htm.

PTC Technical Support professionals comprise a tightly knit team of problem solvers, sharing critical information to help you resolve issues in the shortest possible time with optimal results. Customer Service representatives can provide you with a variety of product related tips and innovative solutions to your unique requirements.