IBM Tivoli Monitoring
6.3 Fix Pack 2

High Availability Guide for Distributed Systems

SC22-5455-01

IBM Tivoli Monitoring
6.3 Fix Pack 2

High Availability Guide for Distributed Systems

SC22-5455-01

Note
Before using this information and the product it supports, read the information in "Notices" on page 159.

This edition applies to version 6, release 3, fix pack 2 of IBM Tivoli Monitoring (product number 5724-C04) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

Chapter 1. Overview

Users rely on IBM Tivoli Monitoring products to monitor the performance and availability of their systems and applications. Because many of these systems run mission-critical applications, it is important that the operation and availability of these systems and applications are monitored continuously, so that prompt remedial action can be taken to fix problems, and potential or impending problems can be pretreated.

Therefore, it is necessary that the monitoring products are always available so that system administrators and other stakeholders can effectively monitor and manage their systems without interruption.

This document presents the options available to Tivoli® Monitoring customers today for ensuring high availability of the Tivoli monitoring components in their environments.

Chapter 2, "Monitoring functions and architecture," on page 3 introduces the concepts and terminology necessary to understand the overall monitoring architecture, and the monitoring features that are relevant to high availability of the monitoring functions.

Chapter 3, "Configuring for high availability and disaster recovery," on page 15 provides a high-level description of the high availability options that can be used with IBM® Tivoli Monitoring components.

Chapter 4, "The hot standby option," on page 21 describes the use of the IBM Tivoli Monitoring hot standby feature as another option for addressing high availability requirements.

Chapter 5, "The clustering of IBM Tivoli Monitoring components," on page 37 describes, in detail, the use of clustering as a technique for addressing IBM Tivoli Monitoring high-availability requirements. It also describes the supported IBM Tivoli Monitoring cluster configurations, provides an overview of the setup steps, and describes the expected behavior of the components that run in a clustered environment.

Chapter 6, "Creating clusters with Tivoli Monitoring components in an HACMP environment," on page 49 provides information on installing and configuring IBM Tivoli Monitoring components in High Availability Cluster Multi-Processing (HACMP™) environments under the IBM AIX® operating system.

Chapter 7, "Creating clusters with monitoring components in a System Automation for Multiplatforms environment," on page 69 provides information on the implementation and design of high-availability IBM Tivoli Monitoring environments installed on Linux and AIX working with the IBM Tivoli System Automation for Multiplatforms product.

Chapter 8, "Creating clusters with Tivoli Monitoring components in a Microsoft Cluster Server environment," on page 93 contains information on designing and implementing highly availably IBM Tivoli Monitoring environments by using Microsoft Windows Cluster Server (MSCS).

Chapter 2. Monitoring functions and architecture

There are two primary technological approaches to configuring resiliency (also known as high availability) for the Tivoli monitoring platform components. One approach exploits common, commercially available, high-availability cluster manager software. Examples include:

- High-Availability Cluster Multiprocessing for pSeries on AIX systems (HACMP)
- IBM Tivoli System Automation for Multiplatforms (SA-MP)
- Microsoft Cluster Server, from Microsoft (MSCS)

In the second approach, the hub Tivoli Enterprise Monitoring Server is resilient to specific failure scenarios. This alternative approach is also referred to as *hot standby* in Tivoli publications. These two approaches provide different resiliency and failover characteristics. *Failover* is the process of taking resource groups offline on one node and bringing them back on another node; the resource dependencies are respected.

The first approach requires the use of a high-availability cluster manager such as HACMP, IBM's SA-MP, or Microsoft's MSCS. Using this approach, you can configure all of the components of the monitoring platform for resiliency in the case of component failure. See the following chapters for a detailed description of how to create clusters with your Tivoli Monitoring components by using each of the cluster managers:

- Chapter 6, “Creating clusters with Tivoli Monitoring components in an HACMP environment,” on page 49
- Chapter 7, “Creating clusters with monitoring components in a System Automation for Multiplatforms environment,” on page 69
- Chapter 8, “Creating clusters with Tivoli Monitoring components in a Microsoft Cluster Server environment,” on page 93

If you are primarily concerned with the availability of the hub Tivoli Enterprise Monitoring Server, the IBM Tivoli Monitoring platform provides the hot standby option. The hot standby option replicates selected state information between the hub monitoring server and a secondary hub monitoring server running in a listening standby mode; the secondary hub monitoring server monitors the active hub's heartbeat so that it can remain up-to-date with the hub’s environment information. In an appropriately configured environment, the secondary (that is, the backup) hub monitoring server takes over as the active hub monitoring server whenever the primary hub monitoring server fails. Hot standby operates without shared or replicated persistent storage between the two monitoring servers and does not require cluster manager software. However, hot standby addresses only the hub monitoring server component of the monitoring platform, and is therefore suited to users without stringent resiliency requirements on the other components of the monitoring platform. High availability and disaster recovery configuration are also possible when using hot standby. Additional strategies must be used to ensure high availability for other IBM Tivoli Monitoring components, including the Tivoli Enterprise Portal Server, the Warehouse Proxy Agent, and the Summarization and Pruning Agent.

The following three features are requirements for implementing high availability on z/OS® systems:

- Sysplex environment
- Shared DASD
- Dynamic Virtual IP Address (DVIPA)

For more information about implementing high availability on z/OS systems, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS*.

Monitoring functions

IBM Tivoli Monitoring products consist of a set of products and components that provide monitoring performance and availability actions and functions, which can be integrated to establish a service-management process. The IBM Tivoli Monitoring components provide you with multiple capabilities under three main areas: visibility, control, and automation. These key functions are defined in the following sections.

Data visualization

Users of the monitoring products use the Tivoli Enterprise Portal, a graphical user interface, or monitoring dashboards in Dashboard Application Services Hub to view various reports about the operations of the monitored environment. The reports include the status of the managed systems, various events, and performance data that is specific to various managed systems, such as the CPU utilization of a particular process or the disk usage of a particular computer system. This type of performance data is also referred to as *performance metrics*.

Situations, events, and alerts

Using the Tivoli Enterprise Portal, users can create monitoring specifications called *situations* to detect when specific conditions or events in their environment occur, thus raising an *alert*. Each situation is assigned (or *distributed*) to one or more managed systems that is to be monitored for a specific condition of a set of conditions.

There are two types of events that might be triggered by a situation: pure or sampled. When the determination of the event must be made based on observations made at specific intervals, the event is known as a *sampled event*. When the event is based on a spontaneous occurrence, the event is known as a *pure event*. Therefore, situations for sampled events have an interval associated with them, while those for pure events do not. Another characteristic of sampled events is that the condition that caused the event can change, thus causing it to be no longer true. Pure events cannot change. Therefore, alerts raised for sampled events can transition from true to not true, while a pure event stays true when it occurs.

An example of a sampled event is `number of processes > 100`. An event becomes true when the number of processes exceeds 100 and later becomes false again when this count drops to 100 or lower. A situation that monitors for an `invalid logon attempt by user` is a pure event: the event occurs when an invalid logon attempt is detected, and does not become a False event.

System managers can also specify actions that must be taken when an event occurs. For example, when a disk becomes full, a command can automatically be run to delete temporary files, thereby reclaiming additional storage.

Event-management integration

In many environments, multiple components might trigger events that must be sent to the Tivoli Enterprise Console® or IBM Tivoli NetCool®/OMNIbus, which can be used for event correlation and management. Tivoli monitoring can be configured to forward events to the Tivoli Enterprise Console or Tivoli NetCool/OMNIbus for further correlation and management.

Workflow policies

In addition to defining situations, the Tivoli Enterprise Portal interface can be used to define workflow policies. A workflow policy can specify complex rules that monitor multiple events and take various actions.

Historical data collection

You can use the Tivoli Enterprise Portal to configure historical data collection for one or more sets of performance metrics. When configured, metrics for the last 24 hours are available for viewing at any time

as short-term history. Optionally, the collected metrics can be sent to the Tivoli Data Warehouse for storage as long-term history. Metrics sent to the warehouse are also available for viewing from the Tivoli Enterprise Portal and from monitoring dashboards that support historical data in Dashboard Application Services Hub.

To prevent unbounded growth of the warehouse database, the summarization and pruning features of the warehouse database can be configured from the Tivoli Enterprise Portal to manage the data in the database.

Monitoring architecture

Tivoli Monitoring products use a set of service components (known collectively as Tivoli Management Services) that are shared by a number of other product suites, including IBM Tivoli XE monitoring products, IBM Tivoli Composite Application Manager products, System Automation for z/OS, Web Access for Information Management, and others. The information in this section is also relevant to these products.

Tivoli Monitoring products, and other products that share Tivoli Management Services, participate in a server-client-agent architecture. Monitoring agents for various operating systems, subsystems, databases, and applications (known collectively as Tivoli Enterprise Monitoring Agents) collect data and send it to a Tivoli Enterprise Monitoring Server. Data is accessed from the monitoring server by Tivoli Enterprise Portal clients and by dashboard users of the Dashboard Application Services Hub. A Tivoli Enterprise Portal Server provides presentation and communication services for these clients. Several optional components such as an historical data warehouse extend the functionality of the framework.

IBM Tivoli Monitoring also includes Jazz™ for Service Management which brings together the Open Services for Lifecycle Collaboration (OSLC) community's open specifications for linking data and other shared integration services, including administrative, dashboard, reporting, and security services. You can use IBM Tivoli Monitoring components to extend the Jazz for Service Management functionality for your monitoring environment.

Before deciding where to deploy the components of the Tivoli Monitoring product in your environment, you should understand the components of the product, the roles that they play, and what affects the load on these components.

Typical configuration

A typical configuration, depicted in “Typical configuration,” incorporates the following components:

- One or more Tivoli Enterprise Monitoring Servers, which act as a collection and control point for alerts received from the agents, and collect their performance and availability data. The monitoring server also manages the connection status of the agents. One server in each environment must be designated as the *hub*.
- A Tivoli Enterprise Portal Server, which provides the core presentation layer for retrieval, manipulation, analysis, and pre-formatting of data. The portal server retrieves data from the hub monitoring server in response to user actions at the portal client, and sends the data back to the portal client for presentation. The portal server also provides presentation information to the portal client so that it can render the user interface views suitably.
- One or more Tivoli Enterprise Portal clients, with a Java-based user interface for viewing and monitoring your enterprise. Tivoli Enterprise Portal offers two modes of operation: desktop and browser.
- Tivoli Enterprise Monitoring Agents, installed on the systems or subsystems you want to monitor. These agents collect data from monitored, or managed, systems and distribute this information either to a monitoring server or to an EIF or SNMP event server such as Netcool/OMNIbus.
- One or more instances of the tacmd Command Line Interface (CLI). This CLI is used to manage your monitoring environment and can also be used to automate many of the administrative functions

performed using the Tivoli Enterprise Portal. The CLI commands either send requests to the Hub monitoring server or to the Tivoli Enterprise Portal Server.

- z/OS only: Tivoli Management Services: Engine (TMS:Engine) provides common functions, such as communications, multithreaded runtime services, diagnosis (dumps), and logging (RKLVLOG), for the Tivoli Enterprise Monitoring Server, monitoring agents, and components of XE products running on z/OS.
- An Eclipse Help Server for presenting help for the portal and all monitoring agents for which support has been installed. The help server is installed with Tivoli Enterprise Portal Server.

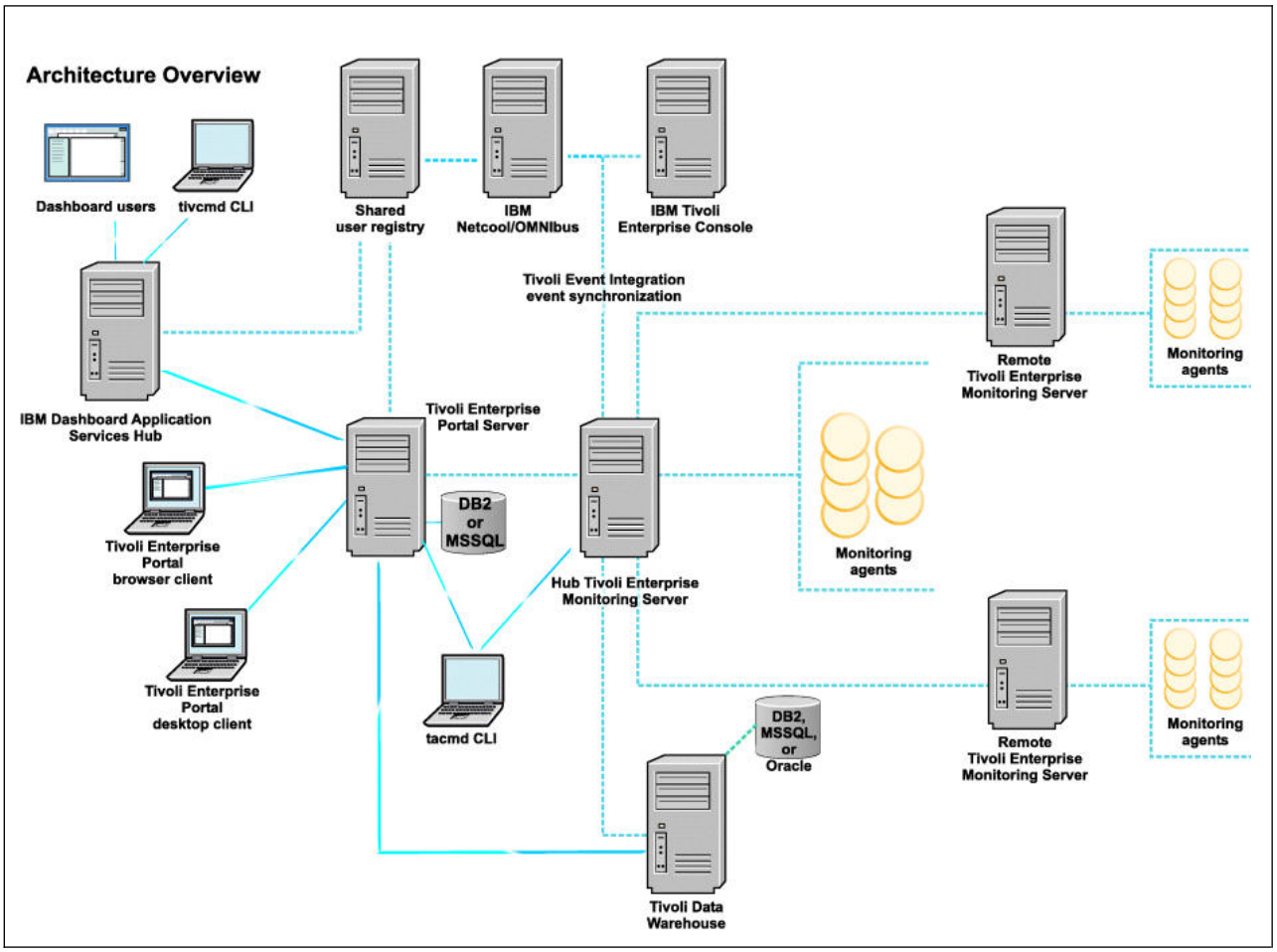

Figure 1. Typical IBM Tivoli Monitoring configuration

Optional components

A configuration optionally includes the following components:

- Tivoli Data Warehouse for storing historical data collected from agents in your environment. The data warehouse is located on an IBM DB2® for Linux, UNIX, and Windows, DB2 on z/OS, Oracle, or Microsoft SQL database. To store data in this database, you must install the Warehouse Proxy Agent. To perform aggregation and pruning functions on the data, you must also install the Summarization and Pruning Agent.
- Event synchronization component, the Event Integration Facility, that sends updates to situation events that have been forwarded to a Tivoli Enterprise Console event server or a Netcool/OMNIbus ObjectServer back to the monitoring server.

- IBM Dashboard Application Services Hub is a Jazz for Service Management component that provides dashboard visualization and reporting services. Operators of the dashboard access it through a web browser interface. You can install the following types of applications into the Dashboard Application Services Hub:
  - The IBM Infrastructure Management Dashboards for Servers application displays situation event information, managed system groups and key health metrics for Windows OS agents, Linux OS agents, and UNIX OS agents. Situation events and monitoring data are retrieved from the Tivoli Enterprise Portal Server using its dashboard data provider.

    Notes:

    1. Other monitoring products, such as IBM Tivoli Monitoring for Virtual Environments and IBM Smart Cloud Monitoring, may provide their own management dashboard applications that use the dashboard data provider. You can also create custom dashboard views that display monitoring data using the Dashboard Application Services Hub user interface.
    2. To use monitoring dashboards, you must enable the dashboard data provider component of the Tivoli Enterprise Portal Server. IBM Dashboard Application Services Hub sends requests for monitoring data to the dashboard data provider which uses the portal server services to retrieve agent data through the monitoring servers.
  - A shared user registry is an LDAP server such as Tivoli Directory Server or Microsoft Active Directory that can be used to authenticate portal server users, IBM Dashboard Application Services Hub users, and optionally, Netcool/OMNIbus Web GUI users. When a shared user registry is used, users are authenticated by the first server that they access and authentication tokens are passed to the other servers so that the users are not required to re-enter their credentials. A shared user registry is strongly recommended if you plan to use IBM Dashboard Application Services Hub with monitoring dashboards. The registry takes advantage of the authorization features supported by IBM Tivoli Monitoring and enables single signon when the portal client is launched from IBM Dashboard Application Services Hub.
  - The Tivoli Authorization Policy Server application is used to create authorization policies that control which managed system groups and managed systems can be viewed by a dashboard operator. The authorization policies are created using the tivcmd Command Line Interface for Authorization Policy and stored at the Authorization Policy Server. The policies are enforced in the dashboard data provider component of the Tivoli Enterprise Portal Server. The dashboard data provider retrieves the policies from the Authorization Policy Server.
  - Tivoli Common Reporting can be used to gather, analyze, and report important trends in your managed environment using historical data from the Tivoli Data Warehouse. The Tivoli Common Reporting user interface is installed with Dashboard Application Services Hub and can be used to display predefined reports provided by monitoring agents and to create custom reports. Tivoli Common Reporting accesses the Tivoli Data Warehouse directly (this interaction is not depicted in "Typical configuration" on page 5.

  IBM Dashboard Application Services Hub Server supports load balancing for high availability. Refer to the *Jazz for Service Management Configuration Guide* for more details on setting up this support.

  However, the Authorization Policy Server does not support load balancing in this release. Therefore, if you set up multiple Dashboard Application Services Hub Severs for load balancing, you can only install the Authorization Policy Server with one of the Dashboard Application Services Hub Servers. Also, when enabling authorization policies in the portal server in your IBM Tivoli Monitoring environment, you must configure the location of the single Dashboard Application Services Hub server where the authorization policy server package is installed and configured. In addition, users of the tivcmd CLI must specify the hostname and port number of the Dashboard Application Services Hub with the Authorization Policy Server instead of the hostname and port number of the load-balancing HTTP server.
- Tivoli Performance Analyzer for predictive capability with Tivoli Monitoring so you can monitor resource consumption trends, anticipate future performance issues, and avoid or resolve problems more quickly.

Open Services Lifecycle Collaboration

When IBM Tivoli Monitoring uses Open Services Lifecycle Collaboration and linked data principles for product integration, depicted in "Open Services Lifecycle Collaboration," the IBM Tivoli Monitoring environment is extended by adding the following components:

- The Tivoli Enterprise Monitoring Automation Server is installed with the Hub monitoring server. It extends the Hub monitoring server by providing the Open Services Lifecycle Collaboration Performance Monitoring (OSLC-PM) service provider. The service provider registers monitoring resources such as computer systems, software servers, and databases with the Registry Services and also responds to HTTP GET requests for resource health metrics from OSLC clients.
- Other products, such as Tivoli Application Dependency Discover Manager, can also provide OSLC service providers that register shared resources such as computer systems, software servers, and databases with Registry Services and that respond to HTTP GET requests from OSLC clients.
- Registry Services is a Jazz for Service Management integration service that provides a shared data repository for products in an integrated service management environment. It reconciles resources registered by multiple service providers. OSLC client applications can retrieve a single record for a shared resource such as a computer system from Registry Services. The record contains URLs that the OSLC client application can use to retrieve additional details about the resource directly from each service provider using HTTP GET requests. Some OSLC clients can display the information retrieved from the service providers in a hover preview on the user interface so that the operator does not have to launch a separate application to see the details. For example, Tivoli Business Service Manager v6.1.1 can display hover preview in its service tree user interface to show health metrics for resources registered by the Performance Monitoring service provider and configuration and change history information registered for the same resources by Tivoli Application Dependency Discovery Manager version 7.2.1 FP4 or later.
- Security Services is an optional Jazz for Service Management service that enables non-WebSphere based servers such as the Tivoli Enterprise Monitoring Automation Server to participate in Lightweight Third Party Authentication (LTPA) based single sign-on with OSLC clients installed on WebSphere® servers. For the V6.3 release of IBM Tivoli Monitoring, the Performance Monitoring service provider assumes that Registry Services and Security Services are installed into the same WebSphere® Application Server.

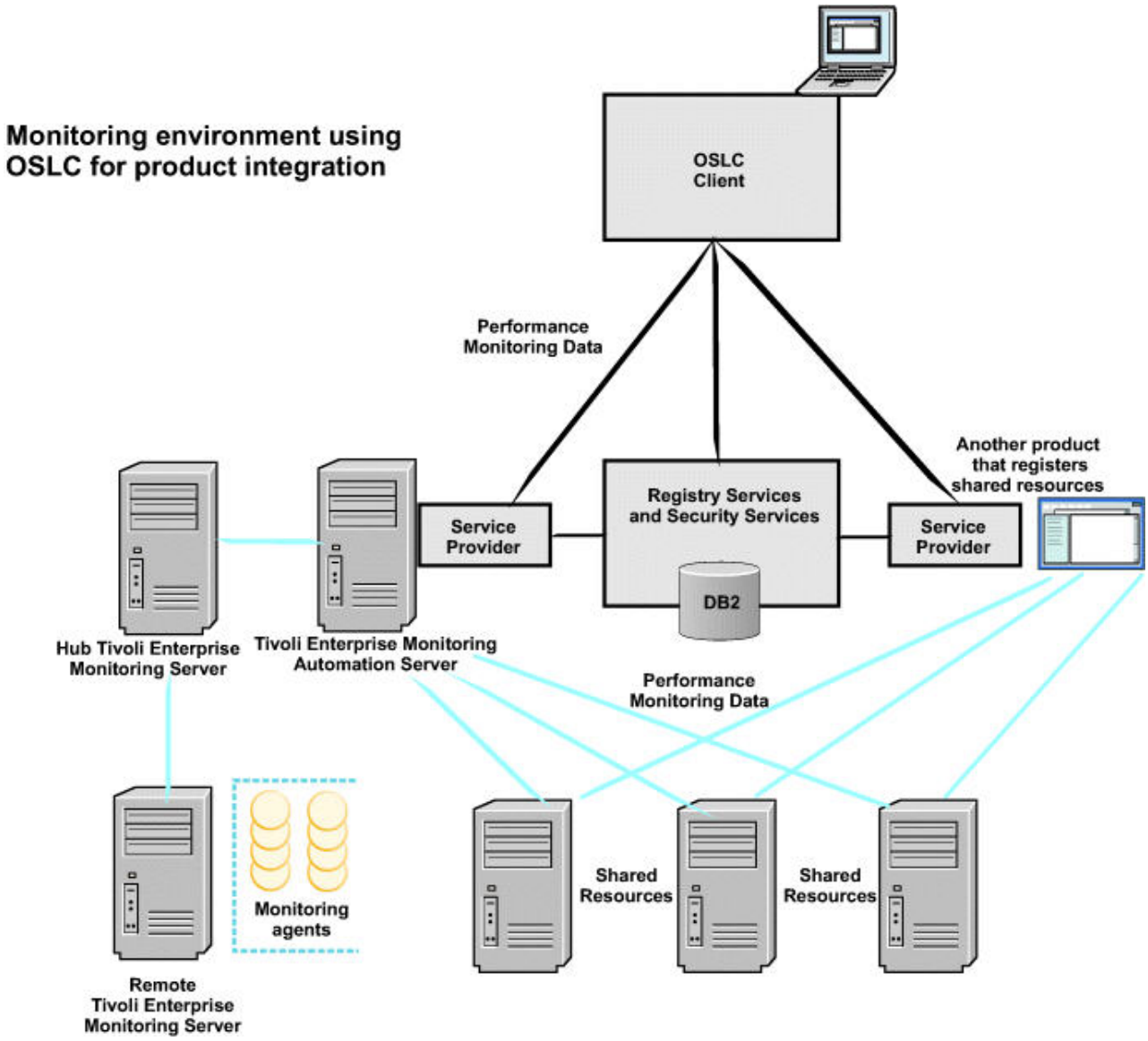

Figure 2. IBM Tivoli Monitoring environment using Open Services Lifecycle Collaboration for product integration

Agent resiliency

The Agent Management Services feature in IBM Tivoli Monitoring V6.2.1 or higher provides resiliency at the agent level. Specifically, the IBM Tivoli Monitoring OS Monitoring Agent for Windows, Linux, or Unix agents are monitored for availability. These agents are automatically restarted according to default policy settings expressed as an XML file. You can create or modify this XML file.

In addition, the OS agents will monitor and automatically restart other agents running adjacent to them according to the terms of the policy files. Policy files for the Warehouse Proxy Agent, Summarization and Pruning Agent, and Tivoli Universal Agent are installed along with the OS Monitoring Agents. The files can be activated by using a set of actions associated with the new Agent Management Services workspace, which is part of the Windows and Linux OS Monitoring Agent navigation tree nodes in the Tivoli Enterprise Portal. By monitoring and responding to abnormal downtime or behavior exhibited by an agent, IBM Tivoli Monitoring adds a layer of fault tolerance to endpoint applications and increases their availability rating.

For further information on Agent Management Services, see Chapter 11 of the *IBM Tivoli Monitoring: Administrator's Guide*.

High Availability considerations for the Tivoli Enterprise Monitoring Server

In general, the Tivoli monitoring components are highly resilient. The components are tolerant of network and communications failures, attempting to reconnect to other components and retry communication until they succeed. The functions described in "Monitoring functions" on page 4 have the following requirements for the various components:

- Tivoli Enterprise Monitoring Agents must be available at all times. If a monitoring agent fails to communicate with a monitoring server, it must be able to connect to another monitoring server and continue operation uninterrupted.
- The hub Tivoli Enterprise Monitoring Server must be available at all times. If the hub fails, another instance of the hub, along with all persistent data files and the failed hub's internal state, must be available to take over for the failed hub.
- Remote Tivoli Enterprise Monitoring Servers must be able to sense a hub failure and then reconnect to the hub as soon as it (or another instance) becomes available, while maintaining its internal state.
- The Warehouse Proxy Agent and Summarization and Pruning Agents must be available at all times. If they fail, another instance of these processes must be available to take over where the failed agent left off.
- Tivoli Enterprise Monitoring Automation Server must be able to sense a hub failure and reconnect to the co-located hub when it becomes available. When Hot Standby is configured, there is an automation server for each hub monitoring server. The automation server that is co-located with the acting hub is responsible for registering OSLC resources and responding to requests from OSLC client applications.

See Table 1 for the failover options available for each monitoring component.

Note: Other options are available to achieve high availability, such as installing multiple Tivoli Enterprise Portal Servers and using the migrate-export and migrate-import commands to synchronize their customization.

Table 1. Options for Tivoli Monitoring component resiliency

Component	Potential single point of failure?	Cluster failover available?	Hot standby failover available?
Hub monitoring server	Yes	Yes	Yes
Portal server	Yes	Yes	No
Tivoli Data Warehouse database	Yes	Yes	No
Warehouse Proxy Agent	Yes, if a single Warehouse Proxy Agent is in the environment.	Yes	No
Summarization and Pruning Agent	Yes	Yes	No
Remote monitoring server	No. Another monitoring server can assume the role of a remote monitoring server for connected agents. This is known as "agent failover."	N/A	N/A
Agent	Not a single point of failure for the whole monitoring solution, but a specific point of failure for the specific resource being monitored.	Yes	No

Table 1. Options for Tivoli Monitoring component resiliency (continued)

Component	Potential single point of failure?	Cluster failover available?	Hot standby failover available?
Tivoli Enterprise Monitoring Automation Server	No, the Tivoli Enterprise Monitoring Automation Server at the peer Hot Standby Hub would take over the role of publishing OSLC resource registrations and responding to metric requests.	No	Yes[1]

Note: [1] Hot standby failover is only available for this component when the co-located hub monitoring server is configured for hot standby and the Tivoli Enterprise Automation Server is installed at each of the hubs in a hot standby environment. This component does not support hot standby failover independent of the hub monitoring server. When the automation server is configured in a Hot Standby environment, the Registry Services component of Jazz for Service Management must be at version 1.1.0.1 (or later). For additional information on configuring the automation server for Hot Standby support, see the *IBM Tivoli Monitoring Installation and Setup Guide*.

For resiliency characteristics for each option, see Table 2.

Table 2. Resiliency characteristics of IBM Tivoli Monitoring components and features

Component	Characteristics of a hub cluster failover	Characteristics of a hub hot standby failover
Hub monitoring server	The hub monitoring server is restarted as soon as the cluster manager detects failure.	Communication failure between hubs causes the standby hub to start processing to establish itself as master, or primary hub server.
Portal server	The portal server reconnects to the hub monitoring server as soon as it is restarted.	The portal server needs to be reconfigured to point to the new hub.
Tivoli Data Warehouse database	No relationship to hub	No relationship to hub
Warehouse Proxy Agent	As an agent, the Warehouse Proxy Agent reconnects to its hub and continues to export data from agents to the Tivoli Data Warehouse.	As an agent configured with a secondary connection to the hub server, the Warehouse Proxy Agent connects to its secondary hub and continues to export data from agents to the Tivoli Data Warehouse.
Summarization and Pruning Agent	As an agent, the Summarization and Pruning Agent reconnects to its hub and continues to summarize and prune data from the Tivoli Data Warehouse.	As an agent configured with a secondary connection to the hub server, the Summarization and Pruning Agent connects to its secondary hub and continues to summarize and prune data from the Tivoli Data Warehouse.
Remote monitoring server	The remote monitoring server detects the hub restart and tries to reconnect, synchronizing with the hub.	When configured with a secondary connection to the hub server, the remote monitoring server retries the connection with the primary hub and if unsuccessful tries to connect to the secondary hub. When the new hub has been promoted to master, the remote monitoring server detects the hub restart and tries to reconnect, synchronizing with the hub.

Table 2. Resiliency characteristics of IBM Tivoli Monitoring components and features (continued)

Component	Characteristics of a hub cluster failover	Characteristics of a hub hot standby failover
Agent	All agents directly connected to the hub reconnect to the hub after restart and begin synchronization.	When configured with a secondary connection to the hub server, agents directly connected to the hub perceive the loss of connection and retry. With the first hub down, the agent tries to connect to the second hub, and begin synchronization that includes restarting all situations.
Event data	Agents resample all polled situation conditions and reassert all that are still true. Situation history is preserved.	Agents resample all polled situation conditions and reassert all that are still true. Previous situation history is not replicated to the failover hub server and thus lost. To persist historical event data, use the Tivoli NetCool/OMNIbus or Tivoli Enterprise Console.
Hub failback (Failback is the process of moving resources back to their original node after the failed node comes back online.)	Available through cluster manager administration and configuration.	The secondary hub must be stopped so that the primary hub can become master again.
Time for failover	The detection of a failed hub and subsequent hub restart is quick and can be configured through the cluster manager. The synchronization process continues until all situations are restarted and the whole environment is operational. The amount of time depends on the size of the environment, including the number of agents and distributed situations.	The detection of a failed hub is quick. There is no restart of the hub, but the connection of remote monitoring server and agents to the standby hub require at least one more heartbeat interval because they try the primary before trying the secondary. The synchronization process continues until all situations are restarted and the whole environment is operational. The amount of time depends on the size of the environment, including the number of agents and distributed situations.
z/OS environments	The clustered solution on a z/OS hub has not yet been tested and therefore is not a supported configuration. Remote monitoring servers on z/OS systems are supported.	Hot standby is fully supported on z/OS systems, for both remote and local hubs.
Data available on failover hub	All data is shared through disk or replication.	All Enterprise Information Base data, except data for the following components, is replicated through the mirror synchronization process: • Situation status history • Publishing of any Tivoli Universal Agent metadata and versioning • Remote deployment Depot

Table 2. Resiliency characteristics of IBM Tivoli Monitoring components and features (continued)

Component	Characteristics of a hub cluster failover	Characteristics of a hub hot standby failover
Manageability of failover	Failover can be automatic or directed through cluster administration. You control which hub is currently the master hub server and the current state of the cluster.	Failover can be directed by stopping the hub. Note that the starting order controls which hub is the master hub server.

Note: When using a clustered hub monitoring server, you must completely shut down for maintenance. However, in a hot standby environment, you can apply a patch one node at a time.
For further information on the primary hub monitoring server and its configuration, see Chapter 5, "The clustering of IBM Tivoli Monitoring components," on page 37.

IBM Tivoli Monitoring portal navigation

The configuration of the Tivoli Enterprise Portal Server includes a new KFW_TOPOLOGY_CLUSTER_LIST environmental variable. Agents with affinities that are included in the KFW_TOPOLOGY_CLUSTER_LIST variable are displayed in the physical view of the navigation tree below the name specified by the CTIRA_HOSTNAME agent variable. By using this enhancement, you can group agents by a unique name in the physical view of the navigation tree. You must include the agent type (the affinity symbolic name) in the list for every agent you plan to use in the cluster.

Note that the default behavior for the CTIRA_HOSTNAME variable is to take on the value of the system host name and is displayed under the system host name in the portal Navigator pane. Setting the CTIRA_HOSTNAME variable for those agents that also appear in KFW_TOPOLOGY_CLUSTER_LIST causes the agents to appear in the physical view of the Navigator tree under the value specified in CTIRA_HOSTNAME. In this way, you can group all the agents from one cluster by setting all the CTIRA_HOSTNAME names to the cluster name. The clustered agents appear in the Navigator pane under the cluster name (SQLCLUSTER) while the Windows OS agents appear under the cluster node names (TIVVM13 and TIVVM14).

The following agent variables are used to modify the agents:

CTIRA_HOSTNAME
<Cluster host name>

CTIRA_SYSTEM_NAME
<Cluster host name>

CTIRA_HIST_DIR
Location for data collection in the Tivoli Data Warehouse (optional).

CTIRA_SIT_PATH
Path to situation files. Can point to shared disk.

CTIRA_LOG_PATH
Path to common log files.

If you have a subnode Agent, you will also want to add:

CTIRA_SUBSYSTEM_ID
<cluster name>

The KFW_TOPOLOGY_CLUSTER_LIST variable includes a number of agents, such as:

AFF_MS_CLUSTER,
// Cluster agent

AFF_MS_SQL_SERVER,
// SQL Server

AFF_NT_EXCHANGE,
// Exchange Server

AFF_ALL_ORACLE,
// Oracle

AFF_ALL_SYBASE,
// Sybase

AFF_SIEBEL
// Siebel

A complete list of the affinity names can be found in the following file:

- For the Windows Tivoli Enterprise Portal Server:
  `C:\ibm\ITM\CNPS\affinity.properties`
- For the Unix/Linux Tivoli Enterprise Portal Server:
  `/opt/IBM/ITM/`*`platform`*`/cq/data/affinity.properties`

You can add entries or remove entries. For example,

```
KFW_TOPOLOGY_CLUSTER_LIST=AFF_UDB_AGENT -AFF_MS_SQL_SERVER
```

adds the affinity UDB agent (AFF_UDB_AGENT) to the list and removes the SQL Server (-AFF_MS_SQL_SERVER) from the list.

You can disable this new behavior (using CTIRA_HOSTNAME at the agent, not grouping by the IP address, and not using the operating system–assigned host name) by using the following setting:

```
KFW_TOPOLOGY_CLUSTER_ENABLE=N
```

To enable this Navigator behavior in the portal on Windows, complete the following steps:

1. Stop the Tivoli Enterprise Portal Server (TEPS).
2. Right-click on the **TEPS** icon in the portal, and select **Advanced->Edit Variables** from the pop-up menu.

   The Tivoli Enterprise Portal Server Override Local Variable Settings dialog box is displayed.
3. Click on the **Variable** pull-down menu and locate the KFW_TOPOLOGY_CLUSTER_LIST variable.
   - If the variable exists, ensure that it is set to the agent affinity (for example, for DB2, AFF_UNIVERSAL_DATABASE).
   - If the variable does not exist, type the variable name, and set the value to the agent affinity (for example, for DB2, AFF_UNIVERSAL_DATABASE).
4. To initiate the changes, start the Tivoli Enterprise Portal Server.

Note: This portal behavior affects all agents with that affinity.
To enable this Navigator behavior in the portal on Linux or AIX, you need to manually edit the environment file (cq.ini).

Chapter 3. Configuring for high availability and disaster recovery

Among the most important considerations in setting up your Tivoli Monitoring environment is ensuring high availability of the product components and being able to recover quickly from failures.

There are multiple components to consider when discussing high availability. Ensuring high availability involves achieving redundancy for every Monitoring component. Disaster recovery means being able to recover from a major outage such as a data center going offline or losing its WAN link.

Configuring the hub monitoring server for high availability and disaster recovery

The hub monitoring server is a highly reliable component, and many users choose to run with a single hub monitoring server and use backup and restore operations to ensure that they have minimum downtime in case of a hardware failure. Other users require higher availability and less downtime and employ multiple hub monitoring servers to achieve either a high availability (HA) environment, disaster recovery (DR) environment, or a combination (high availability and disaster recovery) environment. The following section describes some of the strategies used to achieve the desired level of availability and downtime.

If you have a smaller environment and do not want to invest in additional hardware, you can set up a single hub monitoring server. Because the hub monitoring server is very reliable, there is no need to purchase any additional hardware. You can expect some downtime when patching the hub monitoring server. There are times when the monitoring server must be recycled. If you use one hub, it is important that you use a good backup and restore strategy. You can install the hub in a virtualized environment such as VMWare so that you can quickly bring up an identical virtual operating system to replace the original. In addition, there is a VMWare HA option with release 3.0.x that automates the start of a failing image on a different node.

If you want to achieve high availability, you have two options. The first option is to implement the Hot Standby feature that is built into the monitoring server. Extensive large scale testing has taken place to ensure that Hot Standby is a robust solution. For more information, see Chapter 4, “The hot standby option,” on page 21.The second option is to implement an operating system cluster. Extensive testing has been performed with some operating system clusters. Supported clustering options include Windows Cluster, High Availability Cluster Multi-Processing (HACMP), and IBM Tivoli System Automation for Multiplatforms (SA-MP).

The main difference between the clustering options is the scripts to control the automated control of resources within the cluster. In this sense, Tivoli Monitoring is cluster-ready for other clustering solutions.

For more information about using a high availability z/OS hub monitoring server, see *Configuring the Tivoli Enterprise Monitoring Server on z/OS* (http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.omegamon_share.doc_6.3.0.1/ztemsconfig/ztemsconfig.htm).

Configuring for portal server high availability and disaster recovery

It is important to have multiple portal servers available in case of a hardware failure. While the hub monitoring server tracks the state of your Tivoli Monitoring environment it is not as critical to ensure that data is synchronized in real-time between multiple portal servers. The primary data that you want to protect is the customization that is stored in the portal server database, such as user-defined workspaces. Because this data does not change frequently, a good backup and restore strategy ensures a highly available environment.

You can employ a variety of configurations to achieve both high availability and disaster recovery with the portal server, depending on the amount of hardware you are willing to dedicate to your solution:

OS Cluster

Many users set up an OS Cluster for their portal server. Depending on the clustering software used, the cluster can be set up across a WAN to achieve disaster recovery. For detailed information on setting up the monitoring server and portal server in an OS Cluster, see the sections that follow.

Cold backup

Some smaller users do not want to dedicate CPU cycles and memory to a live backup portal server. If that is the case in your environment, install a second portal server on another computer that serves as a production server. The backup portal server is typically shut down so that it does not use any CPU or memory. If the primary portal server goes down, the cold backup can be brought online. The key for a cold backup portal server is to periodically export the portal server database content and import it into the cold backup. In addition, ensure that the cold backup portal server is patched with the same software levels as the primary portal server and the same application support.

Select one of several methods for exporting and importing the portal server database:

- Using RDBMS backup utilities such as DB2's db2 backup and db2 restore commands
- Using the migrate-export and migrate-import command provided by the Tivoli Monitoring product
- Using a tool like the Tivoli System Automation for Multiplatforms to automate the process of backing up the resources.

If the various portal server databases are not running on the same OS version, then the RDBMS backup and restore utilities will probably not work. In those cases, use the Tivoli Monitoring migrate-export and migrate-import commands as described in the "Replicating the Tivoli Enterprise Portal Server database" chapter of the *IBM Tivoli Monitoring Administrator's Guide*.

Multiple active portal servers

As discussed previously, some users choose to implement a master read-write portal server and one or more read-only portal servers. The strategy for backup and restore is to have one master Tivoli Enterprise Portal Server database where all customization is done. Then, periodically export the content from the "master" portal server and import the content into any other portal server. The import replaces the Tivoli Monitoring content in the portal server database, so be aware that any customization made in the secondary portal server environments will be overwritten during the import.

The export and import of the portal server database can be done in two ways:

- Using RDBMS backup utilities such as DB2's db2 backup and db2 restore commands
- Using the migrate-export and migrate-import command provided by the Tivoli Monitoring product

If the various portal server databases are not running on the same OS version, then the RDBMS backup and restore utilities will probably not work. In those cases, use the Tivoli Monitoring migrate-export and migrate-import commands as described in the "Replicating the Tivoli Enterprise Portal Server database" chapter of the *IBM Tivoli Monitoring Administrator's Guide*. Any Tivoli Enterprise Portal client users and

tacmd CLI users that perform customizations must connect to the "master" portal server. All other portal client users can connect to any of the other portal servers.

You can use a HTTP load balancing server between IBM Dashboard Application Services Hub and multiple active portal servers for a high availability dashboard environment. See "Configuring load balancing for a high availability dashboard environment" in the *IBM Tivoli Monitoring Administrator's Guide*.

Configuring for IBM Dashboard Application Services Hub high availability and disaster recovery

IBM Dashboard Application Services Hub Server supports load balancing for high availability.

See the Jazz for Service Management Configuration Guide for more details on setting up this support, Jazz for Service Management Information Center (http://pic.dhe.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.psc.doc_1.1.0/psc_ic-homepage.html).

However, the Tivoli Authorization Policy Server does not support load balancing in this release. Therefore, if you setup multiple Dashboard Application Services Hub servers for load balancing, you can only install the Authorization Policy Server with one of the Dashboard Application Services Hub servers. Also when enabling authorization policies in the portal server in your IBM Tivoli Monitoring environment, you must configure the location of the single Dashboard Application Services Hub server where the authorization policy server package is installed and configured. The tivcmd CLI users also must specify the hostname and port number of the Dashboard Application Services Hub with the Authorization Policy Server instead of the hostname and port number of the load balancing HTTP server.

Configuring for agent and remote monitoring server high availability and disaster recovery

All agents can be defined with a primary and secondary monitoring server, which allows the agent to connect to the secondary monitoring server if the primary is unavailable. Failover to the secondary monitoring server occurs automatically if the agent fails to communicate with the primary monitoring server.

If no other communication occurs between the agent and the monitoring server, the longest interval it should take for the failover to occur is the heartbeat interval, which defaults to 10 minutes.

The primary concern when building a high availability and disaster recovery configuration for the agents and remote monitoring servers is to determine how many agents to connect to each remote monitoring server. For Tivoli Monitoring V6.3, no more than 1500 monitoring agents should connect to each remote monitoring server.

The following information is important when planning your agents and remote monitoring servers:

- Ensure that failover does not result in many more than 1500 monitoring agents reporting to a single remote monitoring server. There are two strategies users typically take to avoid this situation.
  - The *first* and preferred strategy involves having a spare remote monitoring server. By default, the spare remote monitoring server has no agents connected. When the monitoring agents that report to the primary monitoring server are configured, they are configured to use the spare remote monitoring server for their secondary monitoring server. Over time, network and server anomalies cause the agents to migrate.

    To manage this environment, write a situation to monitor how many agents are connect to the spare remote monitoring server. You can then use the situation to trigger a Take Action command that forces the agents back to their primary remote monitoring server by restarting them. Restarting the

agents cause them to connect to their primary monitoring server. Ideally, migrate the agents back to their primary remote monitoring server when the number of agents connect to the spare monitoring server is greater than 20.

The disadvantage to using a spare remote monitoring server is that you must dedicate a spare server to be the spare remote monitoring server. Some users choose to co-locate this server with the Warehouse Proxy Agent or run in a virtualized environment to minimize the extra hardware required.

– The *second* strategy is to evenly distribute the agents so that they failover to different remote monitoring servers to ensure that no remote monitoring server becomes overloaded. In the example below, there are four remote monitoring servers. In this example, configure one-third of the agents on each remote monitoring server to failover to a different remote monitoring server. Review the following scenario:

  RTEMS_1 has 1125 agents, RTEMS_2 has 1125 agents, RTEMS_3 and RTEMS_4 have 1125 agents.

  A third of RTEMS_1's agents failover to RTEMS_2, a third failover to RTEMS_3, and a third failover to RTEMS_4.

  This strategy ensures that none of the remote monitoring servers become overloaded. The problem with this strategy is that it requires a lot of planning and tracking to ensure that all of the remote monitoring servers are well-balanced.

- If you want your agent to failover to a remote monitoring server in another data center, ensure that you have good network throughput and low latency between the data centers.

Note: Connect a very small number of agents to the hub monitoring server. Typically, only the Warehouse Proxy Agent, Summarization and Pruning Agent, and any OS agents that are monitoring the monitoring server are connected to the hub monitoring server.

Use the Tivoli Monitoring heartbeat capabilities to ensure that agents are running and accessible. The default heartbeat interval is 10 minutes. If an agent does not contact the monitoring server, a status of `MS_Offline` is seen at the monitoring server. An event can be generated when an agent goes offline. An administrator can evaluate whether the agent is having problems or whether there is another root cause. In addition, there is a solution posted on the Tivoli Integrated Service Management Library Web site that leverages the `MS_Offline` status and attempts to ping the server to determine if the server is down or whether the agent is offline. You can find more information by searching for "Perl Ping Monitoring Solution" or navigation code "1TW10TM0F" in the IBM Integrated Service Management Library (http://www.ibm.com/software/brandcatalog/ismlibrary).

Configuring for warehouse high availability and disaster recovery

When setting up the Warehouse for high availability and disaster recovery, the primary concern is backing up the data.

The warehouse database can grow rapidly and has significant change, with many gigabytes of new data inserted per day plus summarization and pruning. Use the native database replication tools to achieve a high availability solution. All of the major database vendors provide data replication tools.

Configuring for Warehouse Proxy Agent high availability and disaster recovery

You need to achieve redundancy with the Warehouse Proxy Agent. Only one Warehouse Proxy Agent can be receiving historical data from a specific agent. You can encounter problems if two Warehouse Proxy Agents are configured to receive historical data from the same agent. To avoid problems, ensure that only one Warehouse Proxy Agent is responsible for collecting the historical data from a remote monitoring server.

To ensure that your Warehouse server performs optimally, ensure that the *WAREHOUSELOG* and *WAREHOUSEAGGREGLOG* tables are pruned on a regular basis.

Note: Beginning with Tivoli Monitoring V6.2.3 the tables *WAREHOUSELOG* and *WAREHOUSEAGGREGLOG* are disabled by default.
Pruning for these tables can be configured by specifying retention intervals in the configuration dialog for the Summarization and Pruning Agent or in the configuration file (KSYENV on Windows, sy.ini on UNIX or Linux).

Configuring for Summarization and Pruning Agent high availability and disaster recovery

Connect the Summarization and Pruning Agent to the hub monitoring servers. When the Hot Standby option is used, the Summarization and Pruning Agent must be configured with the standby hub as the secondary monitoring server. However, there are some additional considerations for achieving high availability with the Summarization and Pruning Agent.

Only one Summarization and Pruning Agent may be running against a warehouse database. Thus, it is important to ensure that there is data integrity within the database and that there is no database deadlock between two competing agents. So, by default, only one Summarization and Pruning Agent must be installed and running.

As in the Warehouse Proxy Agent set up, you want to install a second Summarization and Pruning Agent that serves as a cold backup to the primary Summarization and Pruning Agent. By default, the backup Summarization and Pruning Agent is disabled. Write a situation that detects when the primary Summarization and Pruning Agent is down and automatically starts up the backup Summarization and Pruning Agent through a Take Action command.

Care must be taken in writing the Take Action command to ensure that only one Summarization and Pruning Agent is running at any given time. To ensure the two Summarization and Pruning Agents are not running at the same time, perform the following steps:

1. Have the situation trigger only after the second or third missed heartbeat. Occasionally, there are temporary outages triggered by network problems or routine maintenance. You do not want the automated Take Action to occur during a temporary outage.
2. When starting up the backup Summarization and Pruning Agent using a Take Action command, the primary Summarization and Pruning Agent must be disabled so that it does not accidentally restart until an administrator manually corrects the problem.
3. Write up a documented procedure to ensure that only one of the Summarization and Pruning Agents is brought back online following a failover.

Configuring for Tivoli Performance Analyzer high availability and disaster recovery

The Tivoli Performance Analyzer must always connect to the hub monitoring server. When you use the Hot Standby option, you can configure Tivoli Performance Analyzer Agent with the standby hub as the secondary monitoring server.

Since there can only be one Tivoli Performance Analyzer Agent in your Tivoli Monitoring environment (that is one Tivoli Performance Analyzer Agent per hub monitoring server) it is not possible to setup a secondary agent in Hot Standby mode. However, you can setup a second Tivoli Performance Analyzer Agent and keep it stopped, as long as the primary server is running. The secondary agent can only be started when the primary agent is stopped and disabled. The Tivoli Monitoring Administrator can perform the switch manually, or use Take Action commands. In both cases it is very important to ensure

that only one agent is running at the same time. See "Configuring for Summarization and Pruning Agent high availability and disaster recovery" on page 19 for information on writing the Take Action command.

Chapter 4. The hot standby option

This chapter provides an overview of the IBM Tivoli Monitoring hot standby option. It gives a brief overview of hot standby and explains requirements for preparing hot standby for the IBM Tivoli Monitoring components.

Instructions on how to enable the hot standby feature, and an overview of how each component in the IBM Tivoli Monitoring environment is configured to enable the hot standby feature, are provided in this chapter.

Using hot standby

Figure 3 depicts an environment with the hot standby feature configured.

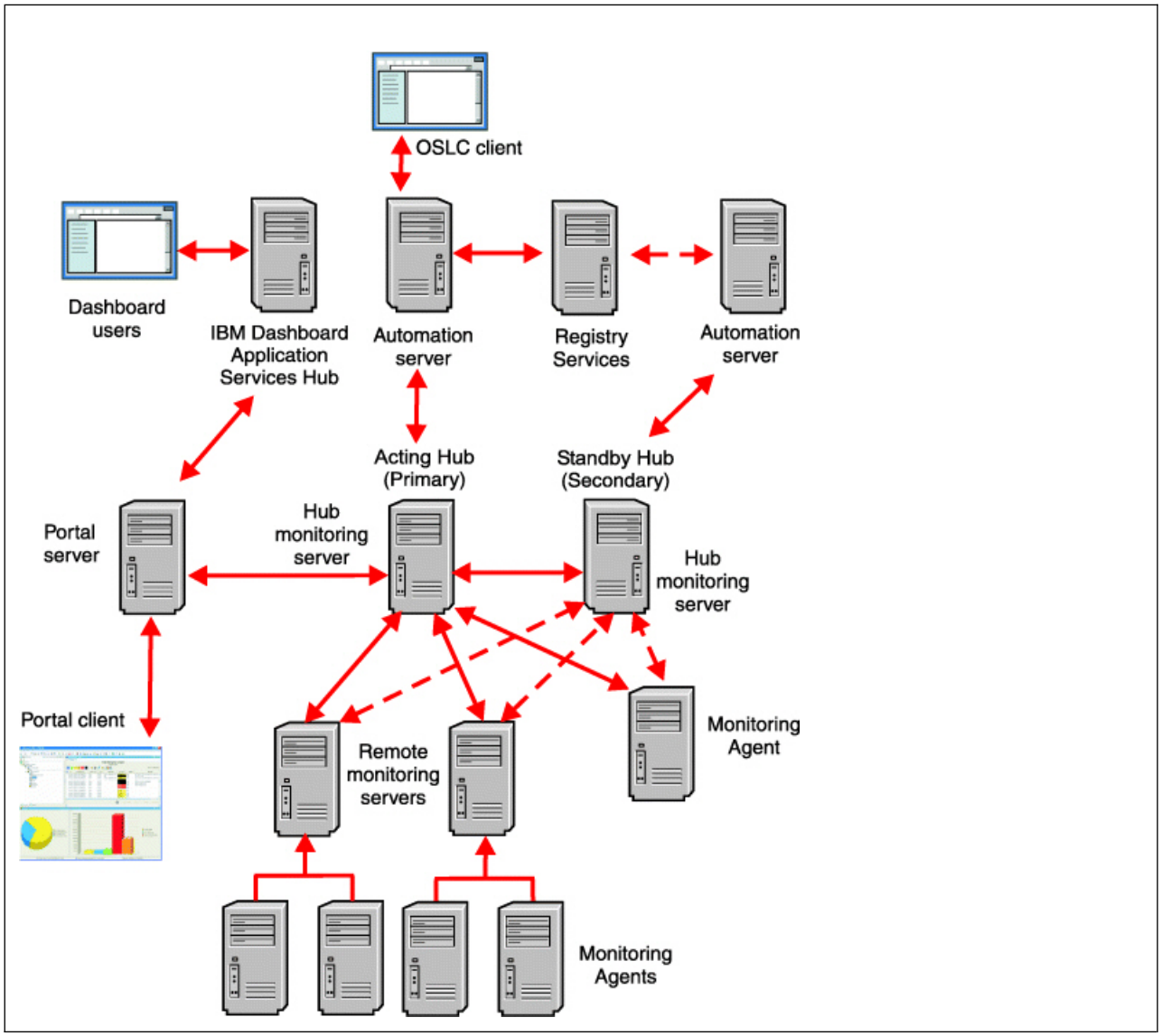

Figure 3. IBM Tivoli Monitoring configuration with hot standby

The following sections provide an overview of how each component in the IBM Tivoli Monitoring environment is configured to enable the hot standby feature.

Hub Tivoli Enterprise Monitoring Servers

In a hot standby environment, there are two hub Tivoli Enterprise Monitoring Servers, which both must be at the same IBM Tivoli Monitoring release levels. The configuration of each hub designates the other hub as the hot standby hub. At any given time, one of the two hub monitoring servers is operating as the hub. This server is referred to as the *acting hub*. The other hub monitoring server is in standby mode and is referred to as the *standby hub*. (See Figure 3 on page 21.)

When the two hub monitoring servers are running, they continuously synchronize the data within their Enterprise Information Base. The Enterprise Information Base contains definition objects such as situations and policies, information about managed systems, and information about the distribution or assignment of situations and policies to managed systems. The hub Tivoli Enterprise Monitoring Servers synchronize data within their Enterprise Information Base to enable the standby hub to take over the role of the acting hub whenever the acting hub becomes unavailable.

The two hub monitoring servers are symmetrical, but for reasons that are explained later, one hub monitoring server is designated as the *primary hub* and the other is designated as the *secondary hub*. While it is not necessary, you can designate as the primary hub the server that you expect to be the acting hub most of the time.

Note that the terms *acting* and *standby* refer to operational states, which can change over a period of time. The terms *primary* and *secondary* refer to configuration, which is relatively permanent.

Remote monitoring servers

All remote monitoring servers must be configured to operate in the hot standby environment. When you configure each remote monitoring server, you specify the primary and secondary hub monitoring servers to which the remote monitoring server reports.

You must specify the same primary and secondary hub monitoring servers for each remote monitoring server. In Figure 3 on page 21, the connections from the remote monitoring servers to the primary hub are depicted with solid arrows. The connections to the standby hub are depicted with dashed arrows.

Tivoli Enterprise Monitoring agents

Monitoring agents that report directly to the hub monitoring server, the Warehouse Proxy Agent, and the Summarization and Pruning Agent must be configured to operate in the hot standby environment. When you configure each of these agents, you specify the primary and secondary hub monitoring servers to which the agents report.

In Figure 3 on page 21, the connection between these monitoring agents and the primary hub is depicted with a solid arrow. The connection to the standby hub is depicted with a dashed arrow.

Tivoli Enterprise Portal server

The Tivoli Enterprise Portal Server cannot be configured to fail over to a standby hub. If the Tivoli Enterprise Portal Server is connected to the standby hub before it takes over as the acting hub, the portal server will need to be recycled in order to reflect data changes. Portal clients do not need to be reconfigured. Portal clients automatically reconnect to the portal server when the portal server is reconfigured and restarted.

Note: You can configure an alias name for the portal server on your DNS server. If you do so, the portal server connects over an alias to the hub instead of by system name or IP address. By using an alias name,

you can switch the Tivoli Enterprise Portal Server by changing the destination for the alias, rather than reconfiguring the Tivoli Enterprise Portal Server. After a recycle, the Tivoli Enterprise Portal Server switch is in effect.

The alias must be set on your DNS Server. The following steps show how the alias can be set on the DNS server:

```
Physical System Name IP-Address ALIAS-Name
physicalsystem1.zurich.com 192.168.1.1 HUB-prod.zurich.com
physicalsystem2.zurich.com 192.168.1.2
```

These steps will allow the ALIAS Name to be moved on either System 1 or 2.

The dashboard data provider component of the portal server has a default provider ID (itm.<*Hub TEMS name*>). The provider ID is sent to the Dashboard Application Services Hub to uniquely identify the data provider connection. This ID also serves as the domain name to define domain-specific authorization policies. If you use the dashboard data provider component to provide monitoring data for display in the Dashboard Application Services Hub Server, then you must configure a domain override for the dashboard data provider when hot standby is used. This configuration ensures the connection between the dashboard server and the portal server and any domain-specific authorization policies are not required to change when the acting Hub changes. The domain override changes the provider ID and domain name for authorization policies to itm.<*domain override value*>. Specify the domain override when you configure your portal server and enable the dashboard data provider. If you modify the domain override after you configure a connection in your Dashboard Application Services Hub Server to the dashboard data provider, then you must delete the connection and add it again. See the *ITM Administrator's Guide* for details on how to configure a dashboard data provider connection.

Also, if you create any domain-specific authorization policies, then you must delete the permissions that specify the previous domain name and create new permissions that specify the new domain name when you change the domain override value. See the *ITM Command Reference* for details on the tivcmd CLI commands that are used to create and work with authorization policies.

Note: Until the portal server is reconfigured to connect to the acting hub, the dashboard data provider component of the portal server cannot retrieve monitoring data for display in the Dashboard Application Services Hub.

Tivoli Enterprise Automation Server

The Tivoli Enterprise Automation Server supports hot standby when it is co-located with a Hub monitoring server that is configured for hot standby. The Tivoli Enterprise Automation Server must be co-located with each Hub monitoring server that is configured for hot standby. When the automation server is configured in a Hot Standby environment, the Registry Services component of Jazz for Service Management must be at version 1.1.0.1 (or later). The Tivoli Enterprise Automation Server does not support hot standby failover independent of the hub monitoring server.

The Tivoli Enterprise Automation Server queries the hot standby role of the Hub that it is co-located with, and assumes a corresponding operational role. When the co-located hub is the acting hub, the Tivoli Enterprise Automation Server is fully operational: it can register resources, and respond to client requests for resource metrics. When the co-located hub is the standby hub, the Tivoli Enterprise Automation Server also assumes a standby role: it does not register resources, and it redirects client requests for resource metrics to the Tivoli Enterprise Automation Server that is co-located with the acting hub. Client requests are redirected with an HTTP 301 status code, and include the URL of the resource at the Tivoli Enterprise Automation Server that is co-located with the acting hub. If client redirection is not possible, an HTTP 404 status code is returned. After a hub failover, the automation server co-located with the acting hub sends a request to Registry Services to update resource URLs to point to the acting automation server. This resource URL remapping minimizes client request redirects and resolves the condition that the automation server for the standby hub is not available to perform a redirect.

Failover scenario

The hot standby operation is best illustrated by describing a scenario. In this example, all monitoring components are started in order, a scenario that might take place only when the product is initially installed. After installation, components can be started and stopped independently.

Starting up the components

The following list describes the order of startup and what happens as each component is started:

1. The primary hub is started first.
   a. When the primary hub starts, it adds a message to its operations log to indicate that it is configured for hot standby:

      ```
      04/17/07  09:38:54 KQM0001  FTO started at  04/17/07  09:38:54.
      ```

   b. The primary hub then attempts to connect to the standby hub (the secondary hub). Because the standby hub is not yet available, the primary hub assumes the role of the acting hub.
   c. When the co-located automation server is started, it determines that the hub has the acting role, registers resources, and responds to OSLC client requests.
2. The secondary hub is started.
   a. When the secondary hub starts, it adds a message to its operations log to indicate that it is enabled for hot standby:

      ```
      04/17/07  09:47:04 KQM0001  FTO started at  04/17/07  09:47:04.
      ```

   b. The secondary hub attempts to connect to the primary hub. The connection succeeds. Sensing that the primary hub started earlier (and is therefore the acting hub), the secondary hub assumes the role of the standby hub, indicated by the following messages:

      ```
      04/17/07  09:47:18 KQM0003  FTO connected to IP.PIPE:#9.52.104.155
      at 04/17/07 09:47:18.
      04/17/07  09:47:33 KQM0009  FTO promoted HUB_PRIMARY as the acting HUB.
      ```

   c. The primary hub also succeeds in connecting with the secondary hub, and issues the following messages:

      ```
      04/17/07  09:45:50 KQM0003  FTO connected to IP.PIPE:#9.52.104.155
      at 04/17/07 09:45:50.
      04/17/07  09:45:58 KQM0009  FTO promoted HUB_PRIMARY as the acting HUB.
      ```

   d. The standby hub queries the acting hub for any updates to the Enterprise Information Base data since it last communicated with the acting hub. The standby hub replicates all updates.
   e. After the initial startup and connections, the two hubs monitor connections with each other periodically to ensure that the other hub is running and that there is no change in status.
   f. The standby hub also monitors the acting hub periodically for further updates to the Enterprise Information Base, and replicates the updates in its own Enterprise Information Base. By default, this monitoring takes place every 5 seconds. As a result, the standby hub is ready to take over the role of the acting hub when required.
   g. When the co-located automation server is started, it determines that the hub has the standby role and, therefore, does not perform any OSLC resource registration.

   Note: If the secondary hub is recycled after initial startup, the node status at the secondary hub is not necessarily the same as the node status at the primary hub. This does not cause an operational problem because the node status at the secondary hub is corrected when a failover occurs from the primary hub to the secondary hub.
3. The remote monitoring servers and monitoring agents are started.

   When the remote monitoring servers and monitoring agents start, they attempt to connect to the primary hub in their configuration. In this scenario, the primary hub is also the current acting hub. Therefore, the connection attempt is successful, and these components start reporting to the primary hub.
4. The Tivoli Enterprise Portal Server connects to the primary hub.

The portal server is configured to connect to the primary hub. One or more portal clients are connected to the portal server for monitoring purposes.

Failing over

The acting hub might become unavailable for a number of reasons. It might need to be shut down for scheduled maintenance, the computer on which it is running might need to be shut down or might have stopped, or it can be experiencing networking problems.

When the standby hub discovers that the acting hub is unavailable, it takes over the role of the acting hub and issues the following messages:

```
04/17/07  10:46:40 KQM0004  FTO detected lost parent connection
at 04/17/07 10:46:40.
04/17/07  10:46:40 KQM0009  FTO promoted HUB_SECONDARY as the acting HUB.
```

The primary hub is now the standby hub and the secondary hub is the acting hub, as depicted in Figure 4:

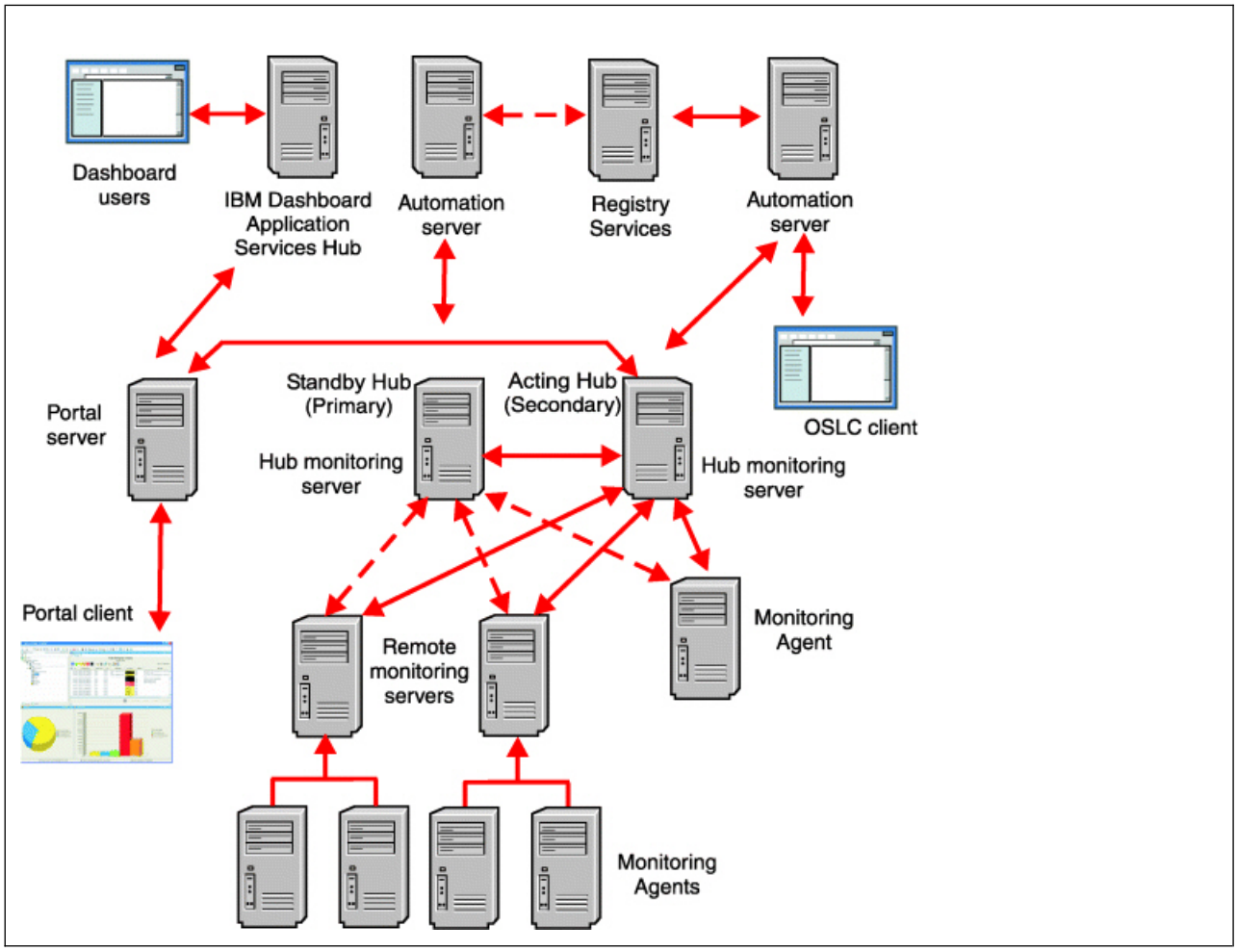

Figure 4. Configuration after failover

The automation server that is co-located with the secondary hub determines that the secondary hub is now the acting hub. Therefore, this automation server assumes the acting role for OSLC resource registration. As part of the failover processing, it sends a request to Registry Services to update resource URLs to point to the new acting automation server and begins registering new and changed resources. Because the resource URLs were updated in Registry Services, the new acting automation server receives

and responds to resource requests from OSLC clients. If any OSLC clients cached the old resource URLs, the automation server co-located with the standby hub redirects the client requests to the new acting automation server.

As the remote monitoring servers and agents connected to the previous acting hub discover that the primary hub is no longer available, they switch and reconnect to the new acting hub. Because these components are in various states of processing and communication with the hub monitoring server, the discovery and reconnection with the new hub is not synchronized.

All remote monitoring servers and agents now report to the new acting hub. There is no mechanism available to switch them back to the standby hub while the acting hub is still running. The only way to switch them to the standby hub is to shut down the acting hub.

The processing that takes place after reconnection is similar to the processing that takes place after reconnection in an environment without a hot standby server. The following processing applies with regard to situations and policies:

1. Pure events that occurred before the failover are not visible. Subsequent pure events are reported when they occur.
2. Sampled situations are reevaluated and are reported again if they are still true.
3. A Master Reset Event is sent to the Tivoli Enterprise Console when the failover occurs. Events that result from situations being reevaluated are resent to the Tivoli Enterprise Console if the monitoring server has been configured to send events to the Tivoli Enterprise Console.
4. Policies are restarted.

The Tivoli Enterprise Portal Server must be reconfigured to point to the new acting hub and then restarted. All portal clients reconnect to the portal server after its restart.

When reconfiguring the portal server on Windows systems for a different monitoring server, a window is displayed asking if a snapshot of the portal server data should be taken. **No** is the correct response when reconfiguring the portal server for a hot standby monitoring server because the same portal server data is relevant to both the primary and hot standby monitoring server.

When **Yes** is selected as the response to the dialog, a snapshot of the portal server data is taken through the "migrate-export" process. The data is saved in a file called `saveexport.sql` and is placed in the `%CANDLE_HOME%\CNPS\CMS\`*`HOSTNAME:Port`* directory, where *HOSTNAME:Port* is the current monitoring server hostname and connection port number.

Then, if no existing snapshot exists for the monitoring server that is being switched to, a new set of portal server data is used and all the customizations are not included. In order to get these restored for use on the new monitoring server, a "migrate-import" needs to be run using the `saveexport.sql` created from the snapshot.

When reconfiguring the portal server to switch back to previous monitoring server, answering **Yes** causes the previous snapshot to be automatically loaded thus restoring the customization. Responding **No** should be done when switching between the primary hub monitoring server and the hot standby monitoring server since the same portal server data should be relevant to both.

The new acting hub, which is the secondary hub, retains its role even after the primary hub Tivoli Enterprise Monitoring Server becomes operational again. The primary hub monitoring server now becomes the standby hub. When the new standby hub starts, it checks the Enterprise Information Base of the new acting hub for updates and replicates updates to its own Enterprise Information Base if necessary. The two hub Tivoli Enterprise Monitoring Servers also start monitoring connections with each other to ensure that the other hub is running.

If a remote monitoring server or agent experiences a transient communication problem with the acting hub and switches over to the standby hub, the standby hub instructs it to retry the connection with the acting hub because the standby hub knows that the acting hub is still available.

The environment continues to operate with the configuration shown in Figure 4 on page 25 until the acting hub is shut down or until the computer on which the acting hub is running becomes unavailable. Each time the acting hub becomes unavailable, the failover scenario described in this section is repeated.

Configuring failover support

The optional hot standby feature enables you to maintain high availability by defining a standby monitoring server to provide failover support for your hub monitoring server. If the hub monitoring server fails, hub functions automatically switch to the backup monitoring server. IBM Tivoli Monitoring automatically connects all remote monitoring servers and agents to the backup monitoring server.

Configuring the hot standby feature involves the following steps:

1. "Configuring the hot standby feature for hub monitoring servers"
2. "Configuring the hot standby feature for remote monitoring servers" on page 31
3. "Tivoli Enterprise Monitoring agents" on page 22
4. "Verifying that failover support is working" on page 34

Configuring the hot standby feature for hub monitoring servers

The two hub Tivoli Enterprise Monitoring Servers that you configure for hot standby (the primary hub and the secondary hub) must be exact copies or *mirrors* of each other:

- The IBM Tivoli Monitoring software on both hub monitoring servers must be at the same release and maintenance level.
- Both hub Tivoli Enterprise Monitoring Servers must be on the same IBM Tivoli Monitoring level of code and maintenance.
- Application support for all required applications must be installed on both monitoring servers.
- The settings for the KMS_SDA parameter on the primary and secondary Tivoli Enterprise Monitoring Servers must match. A mismatch is reported in the monitoring server operational log (MSG2) and in audit logging with message ID KQMSD100.

Perform the following procedures to configure the primary hub and secondary hub monitoring servers for the hot standby feature. These procedures and the examples are ordered as if both hub monitoring servers are installed on the same operating system. However, it is possible to have one hub on a Windows system and the other hub on a UNIX system. Complete the procedure that applies to each system.

On Windows systems: Configuring the hot standby feature for hub Tivoli Enterprise Monitoring Server

Complete the following steps to configure the primary hub and secondary hub monitoring servers for the hot standby feature on Windows. On the configuration windows shown as examples in this procedure, the primary hub is named HUB_PRIMARY and the secondary hub is named HUB_SECONDARY. Actual names can be different.

Configuring the primary hub:
About this task

Install the two hub Tivoli Enterprise Monitoring Servers and then complete the following steps to configure the primary hub to point to the secondary hub as its backup or standby server:

Procedure

1. On the Manage Tivoli Monitoring Services window on the server that will contain the primary hub, right-click the name of the monitoring server that you are going to use as the primary hub, and click **Reconfigure**.

   The monitoring server stops automatically.
2. Complete the following steps on the Tivoli Enterprise Monitoring Server Configuration window:
   a. Enter the name of the primary hub in the **TEMS Name** field.
   b. Select the **Configure Hot Standby TEMS** check box to configure the secondary hub as a standby server for the primary hub.
   c. Specify the protocols used by the secondary hub. These protocols must be the same for both monitoring servers (the primary hub and secondary hub).

      Figure 5 shows an example of a completed configuration window:

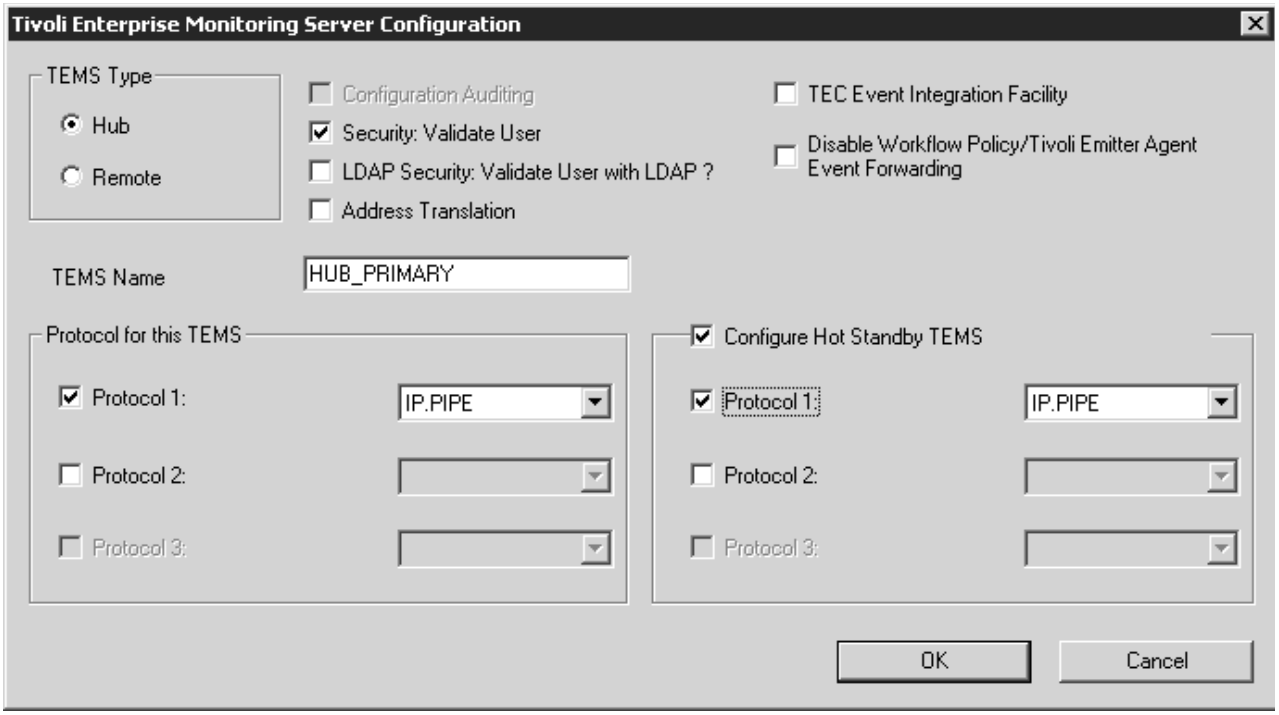

Figure 5. Tivoli Enterprise Monitoring Server Configuration window: primary hub configuration

   d. Click OK.
3. On the Hub Tivoli Enterprise Monitoring Server Configuration window, enter the host name or IP address of the primary hub and verify the communication settings for this server. Click **OK**.
4. On the Hub TEMS Configuration for Hot Standby window, specify the secondary hub as the standby server for the primary hub. Enter the host name or IP address of the secondary hub in the **Hostname or IP Address** field, as shown in the following example.

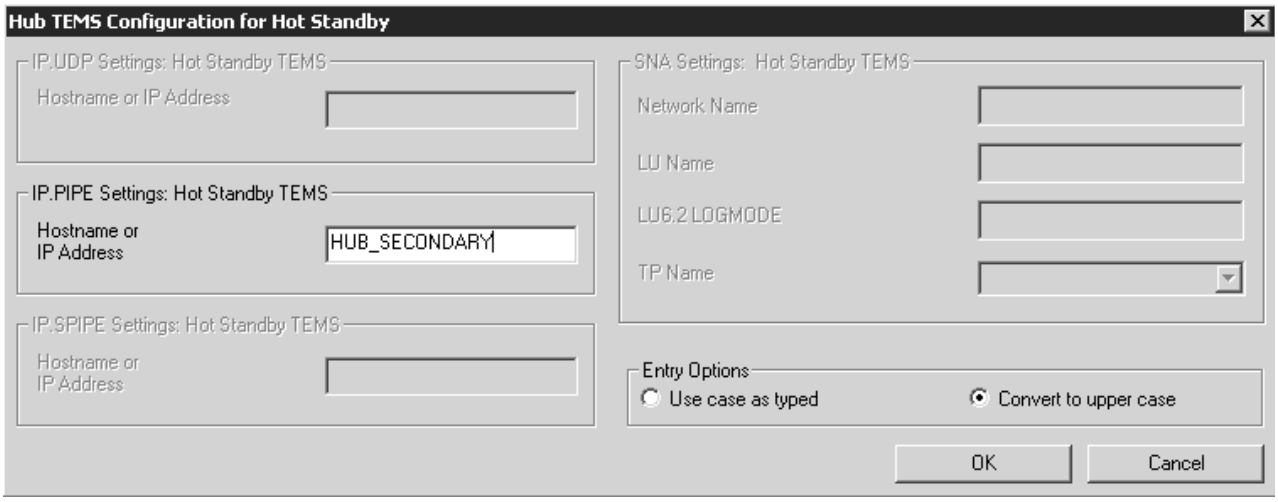

Figure 6. Specifying the secondary hub as a backup or standby server for the primary hub

5. Click **OK**.
6. Restart the monitoring server.

Configuring the secondary hub:
About this task

Complete the following steps to configure the secondary hub to point to the primary hub as its backup or standby server:

Procedure

1. On the Manage Tivoli Monitoring Services window on the Server that will contain the primary hub, right-click the name of the monitoring server that you are going to use as the secondary hub, and click **Reconfigure**.

   The monitoring server stops automatically.
2. Complete the following steps on the Tivoli Enterprise Monitoring Server Configuration window:
   a. Enter the name of the secondary hub in the **TEMS Name** field.
   b. Select the **Configure Hot Standby TEMS** check box to configure the primary hub as a standby server for the secondary hub.
   c. Specify the protocols used by the primary hub. These protocols must be the same for both monitoring servers (the primary hub and secondary hub).

      Figure 7 on page 30 shows an example of a completed configuration window:

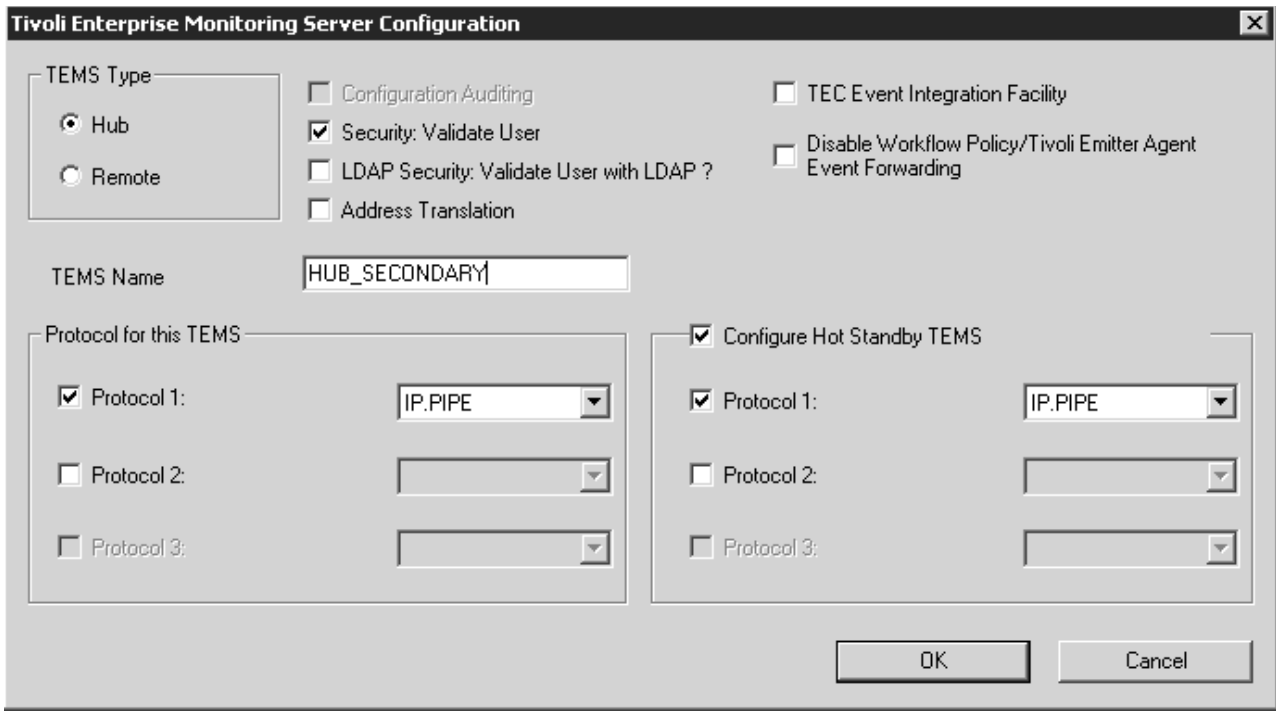

Figure 7. Tivoli Enterprise Monitoring Server Configuration window: secondary hub configuration

d. Click **OK**.

3. On the Hub TEMS Configuration window, enter the host name or IP address of the secondary hub, and verify the communication settings for this server. Click **OK**.
4. Select **Configure Hot Standby**.
5. Click **OK**.
6. On the Hub TEMS Configuration for hot standby window, specify the primary hub as the standby server for the secondary hub. Enter the host name or IP address of the primary hub in the **Hostname or IP Address** field, as shown in Figure 8.

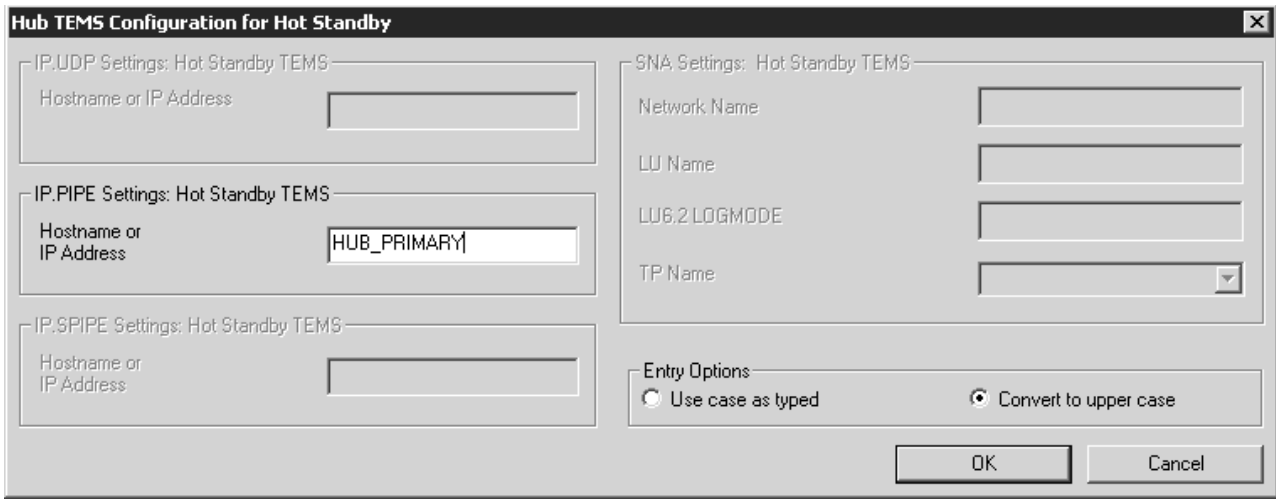

Figure 8. Specifying the primary hub as a backup or standby server for the secondary hub

7. Click **OK**.
8. Restart the monitoring server on the hub Primary server.

On Linux or UNIX systems: Configuring the hot standby feature for hub monitoring servers

Complete the following steps to configure the primary hub and secondary hub monitoring servers for the hot standby feature on Linux or UNIX computers.

Configuring the primary hub:
About this task

Install two hub Tivoli Enterprise Monitoring Servers and then complete the following steps to configure the primary hub to point to the secondary hub as its backup or standby server:

Procedure

1. On the Manage Tivoli Monitoring Services window on the server which contains the Hub Primary, right-click the name of the monitoring server that you are going to use as the primary hub, and click **Configure**.
2. Click the **Advanced Settings** tab.
3. Select **Specify Hot Standby**.
4. Type the host name of the secondary hub in the **Standby TEMS Site** field.
5. Select the type of protocol to use for hot standby. This should be the same protocol on both the primary hub and the secondary hub.
6. If you specified any backup protocols for the primary hub, specify identical protocols for the secondary hub.
7. Click **Save**.
8. Stop and restart the monitoring server.

Configuring the secondary hub:
About this task

Complete the following steps to configure the secondary hub to point to the primary hub as its backup or standby server:

Procedure

1. On the Manage Tivoli Monitoring Services window on the server to contain Hub Secondary, right-click the name of the monitoring server that you are going to use as the secondary hub, and click **Configure**.
2. Click the **Advanced Settings** tab.
3. Select **Specify Hot Standby**.
4. Type the host name of the primary hub in the **Standby TEMS Site** field.
5. Select the type of protocol to use for hot standby. This should be the same protocol on both the primary hub and secondary hub.
6. If you specified any backup protocols for the secondary hub, specify identical protocols for the primary hub.
7. Click **Save**.
8. Stop and restart the monitoring server.

Configuring the hot standby feature for remote monitoring servers

Configure remote monitoring servers to switch to a standby hub monitoring server when the acting hub monitoring server becomes unavailable. Configure all remote monitoring servers consistently by specifying the primary hub as the hub to which they connect and the secondary hub as the standby hub.

On Windows systems: Configuring the hot standby feature for remote monitoring servers

About this task

Complete the following steps on Windows to configure a remote monitoring server to switch to a standby hub when the acting hub monitoring server becomes unavailable:

Procedure

1. On the Manage Tivoli Monitoring Services window, right-click the name of a remote monitoring server, and click **Reconfigure**.

   The monitoring server stops automatically.
2. Complete the following steps on the Tivoli Enterprise Monitoring Server Configuration window:
   a. In the **TEMS Type** field, select **Remote** to indicate that this is a remote monitoring server.
   b. Enter the name of the remote monitoring server in the **TEMS Name** field.
   c. Select the **Configure Hot Standby TEMS** check box to configure a Standby server.
   d. Specify the protocols used by the Standby server. The protocols are the same for the primary hub and secondary hub.

      Figure 9 shows an example of a completed configuration window:

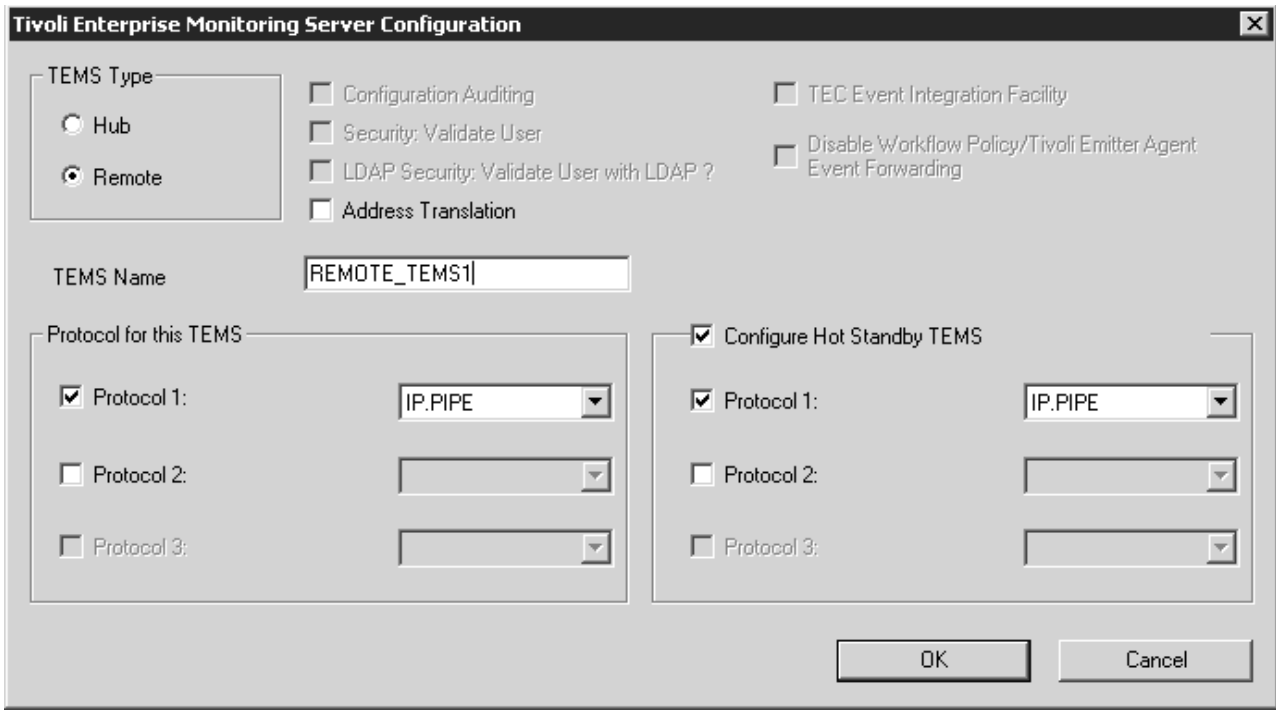

Figure 9. Configuring hot standby for a remote monitoring server

   e. Click **OK**.
3. On the Hub TEMS Configuration window, enter the host name or IP address of the primary hub, and verify the communication settings for this server. Click **OK**.
4. On the Hub TEMS Configuration for Hot Standby window, specify the secondary hub as the standby server. Enter the host name or IP address of the secondary hub in the **Hostname or IP Address** field.
5. Click **OK**.
6. Restart the remote monitoring server.

On Linux or UNIX systems: Configuring the hot standby feature for remote monitoring servers

About this task

Complete the following steps on Linux or UNIX systems to configure a remote monitoring server to switch to a standby hub when the active hub monitoring server becomes unavailable:

Procedure

1. On the Manage Tivoli Monitoring Services window, right-click the name of a remote monitoring server, and click **Configure**.
2. Click the **Advanced Settings** tab.
3. Select **Specify Hot Standby**.
4. Type the host name of the secondary hub in the **Standby TEMS Site** field.
5. Select the type of protocol to use for hot standby. This is the same protocol on both the primary hub and secondary hub.
6. If you specified any backup protocols for the hub monitoring servers, specify identical protocols for the remote monitoring server.
7. Click **Save**.
8. Stop and restart the monitoring server.

Configuring the hot standby feature for monitoring agents

About this task

Configure any monitoring agents that report directly to the hub monitoring server, the Warehouse Proxy Agent, and the Summarization and Pruning Agent to switch to a standby hub monitoring server when the Active hub monitoring server becomes unavailable. Configure all agents consistently by specifying the primary hub as the hub to which they connect and the secondary hub as the standby hub. Complete the following steps:

Procedure

1. On the Manage Tivoli Monitoring Services window, right-click a monitoring agent, and click either **Reconfigure** (on Windows systems) or **Configure** (on UNIX systems).
2. Select **Optional: Secondary TEMS Connection**, and specify the protocol for the standby monitoring server.

   On UNIX agents, click **Protocols** to display a separate window where you can configure the standby server (secondary hub).

   Figure 10 on page 34 shows an example of a completed configuration window on Windows systems:

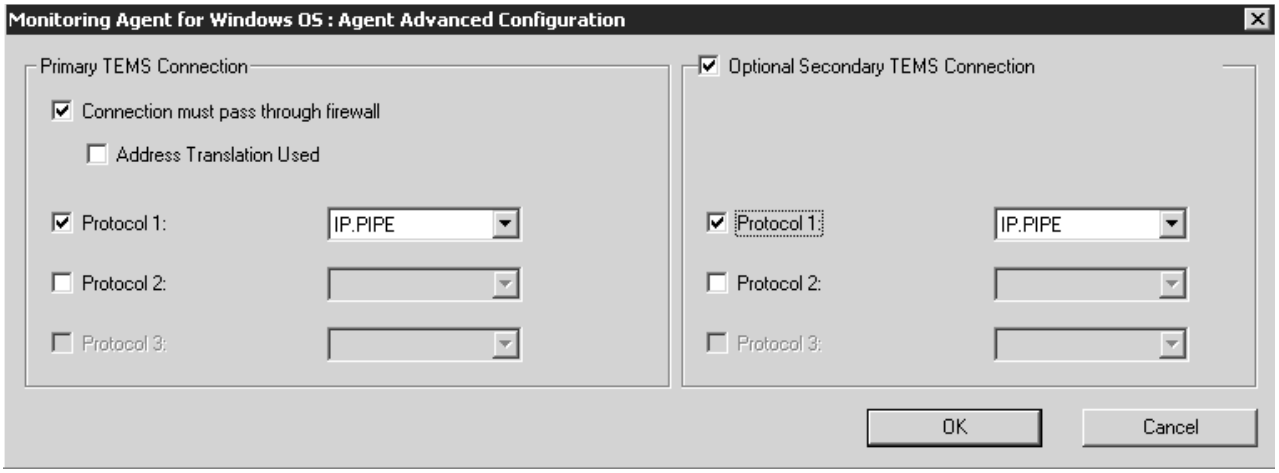

Figure 10. Configuring a Windows monitoring agent to connect to a standby hub monitoring server

3. Click **OK**.
4. Enter the host name or IP address of the primary hub and the port number, and click **OK**.
5. Enter the host name or IP address of the secondary hub, and click **OK**.
6. Restart the monitoring agent.

Configuring the hot standby feature for automation servers

The Tivoli Enterprise Automation Server supports hot standby when it is co-located with a Hub monitoring server that is configured for hot standby.

For instructions on configuring the hot standby feature for the Tivoli Enterprise Automation Server, see "Installing and configuring Tivoli Enterprise Monitoring Automation Server" in the *IBM Tivoli Monitoring Installation and Setup Guide*.

Verifying that failover support is working

To verify that the failover support provided by the hot standby feature is working, take your hub monitoring server offline by stopping it in Manage Tivoli Monitoring Services. When the hub monitoring server has stopped, reconfigure the Tivoli Enterprise Portal Server to point to the backup monitoring server, and restart the portal server. Open the Tivoli Enterprise Portal. If everything is configured correctly, you can open the portal and view data.

Self describing feature in a failover environment

In a Hot Standby failover environment, files required for the self describing feature are replicated to the standby monitoring server and audit records clearly indicate whether self describing installations at the standby hub are initiated by failover.

If the self describing feature is enabled in a Hot Standby failover environment, the following behaviors take place:

- Tivoli Enterprise Monitoring Server monitoring definitions, such as situations, distributions, and calendars, are replicated directly, as in the past, to the standby hub monitoring server.
- Application installation records are replicated and allow the standby hub to collect self describing support files from the acting hub monitoring server so that application support is installed on the standby hub and Tivoli Enterprise Portal Server and browser support files are available for a portal server that might connect to the standby hub.
- Self describing agent configuration option settings are replicated.

- Pending self describing agent requests are cancelled when a hub switch takes place to avoid connection error. These requests are redriven when the first agent of that product connects to the new acting hub.

Chapter 5. The clustering of IBM Tivoli Monitoring components

This chapter provides an overview of the clustering of IBM Tivoli Monitoring components, some supported clustering configurations, and instructions on the setup of IBM Tivoli Monitoring components in a clustered environment. This chapter also includes information on the operations of the Tivoli Monitoring infrastructure in a clustered environment when encountering a failover or failback. *Failback* is the process of moving resources back to their original node after the failed node comes back online.

Clustering overview

This section uses clustering techniques to provide an overview of IBM Tivoli Monitoring high availability. In addition to a brief overview of clustering, this section includes requirements for preparing to create clusters with Tivoli Monitoring components; it also describes the supported IBM Tivoli Monitoring cluster configurations, provides an overview of the setup steps, and describes the expected behavior of the components running in a clustered environment.

Detailed instructions on how to set up Tivoli Monitoring components on different cluster managers are provided in the following chapters.

Review the following concepts to enhance your understanding of clustering technology:

Cluster
A cluster is a group of individual computer systems working together to provide increased application availability.

Failback
Failback is the process of moving resources back to their original node after the failed node comes back online.

Failover
Failover is the process of taking resource groups offline on one node and bringing them back on another node. Resource dependencies are respected.

Node A node is a computer system that is a member of a cluster.

Resource
A resource is a physical or logical entity that can be managed by a cluster (that is, brought online, taken offline, or moved between nodes).

Resource group
Resource groups are collections of resources that are managed as a single unit and hosted on one node at any point in time.

Although clusters can be used in configurations other than basic failover (for example, load sharing and balancing), the current IBM Tivoli Monitoring design does not support multiple, concurrent instances of the monitoring components. For this reason, this document addresses only IBM Tivoli Monitoring component failover.

Supported configurations

The following clustering configurations are presented:

Configuration A
The hub monitoring server, portal server, and data warehouse (with the Summarization and Pruning Agent and Warehouse Proxy Agent) each have their own cluster.

Configuration B
Same as Configuration A, but without the Summarization and Pruning Agent and the Warehouse Proxy Agent in the data warehouse cluster.

Configuration C
The hub monitoring server, portal server, and data warehouse (with the Summarization and Pruning Agent and Warehouse Proxy Agent) are clustered on the same cluster.

The IBM Universal Database (DB2) is used as the database in all of the configuration tests. Other IBM Tivoli Monitoring-supported databases can also be clustered by following the specific database cluster setup procedures.

Note: All the clustering configurations include at least one agent that directly reports to the hub. For simplicity, this configuration is not shown in the configuration diagrams that follow.

Configuration A

This configuration contains the following components:

- Hub cluster
- Portal server cluster
- Data warehouse cluster (including the Summarization and Pruning Agent and the Warehouse Proxy Agent)
- Multiple remote monitoring servers (RT1, RT2, RT3) and agents
- Tivoli Enterprise Console integration

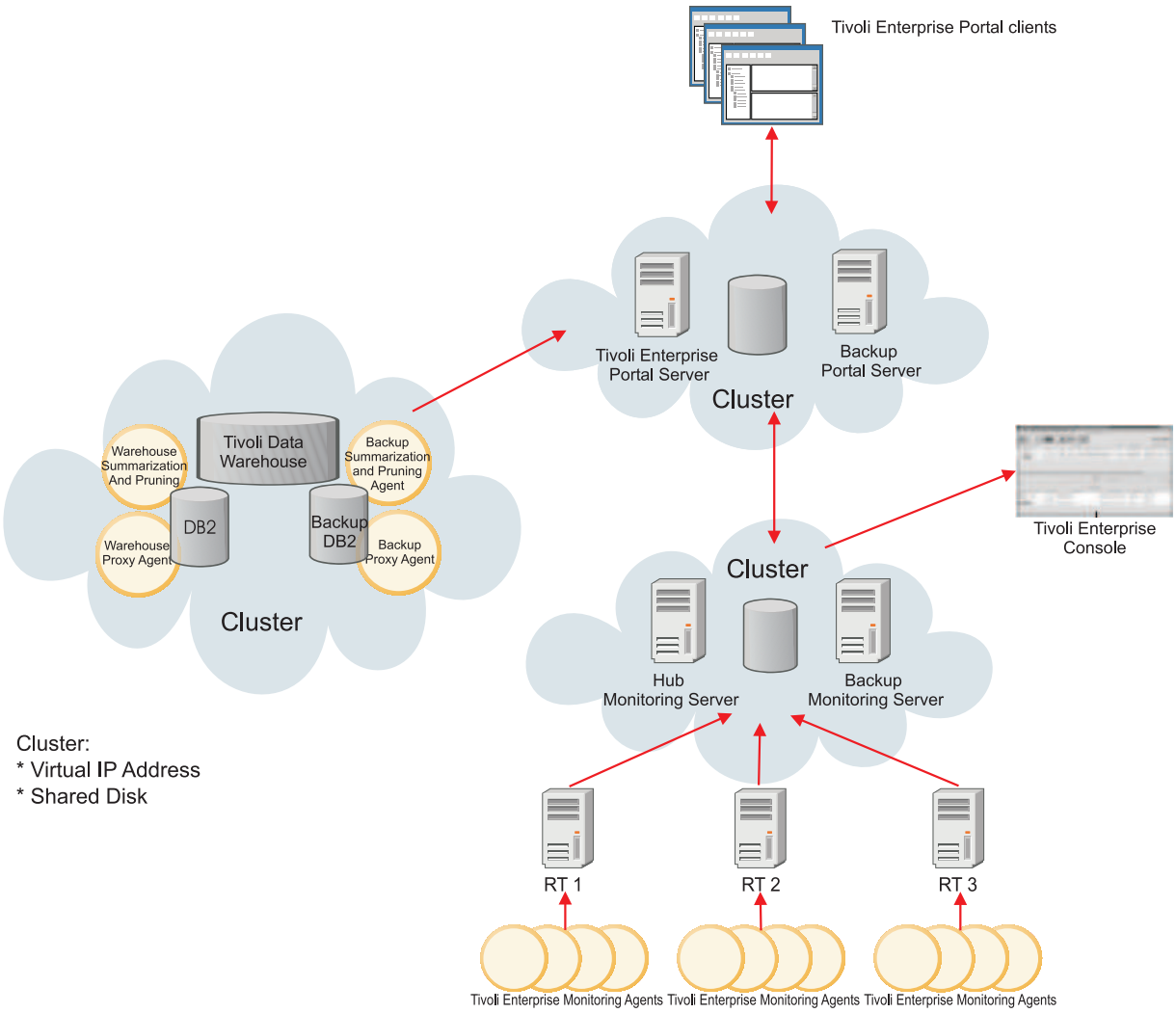

Figure 11. Separate component clusters

Configuration B

In this configuration, the Warehouse Proxy Agent and Summarization and Pruning Agent run outside of the data warehouse cluster.

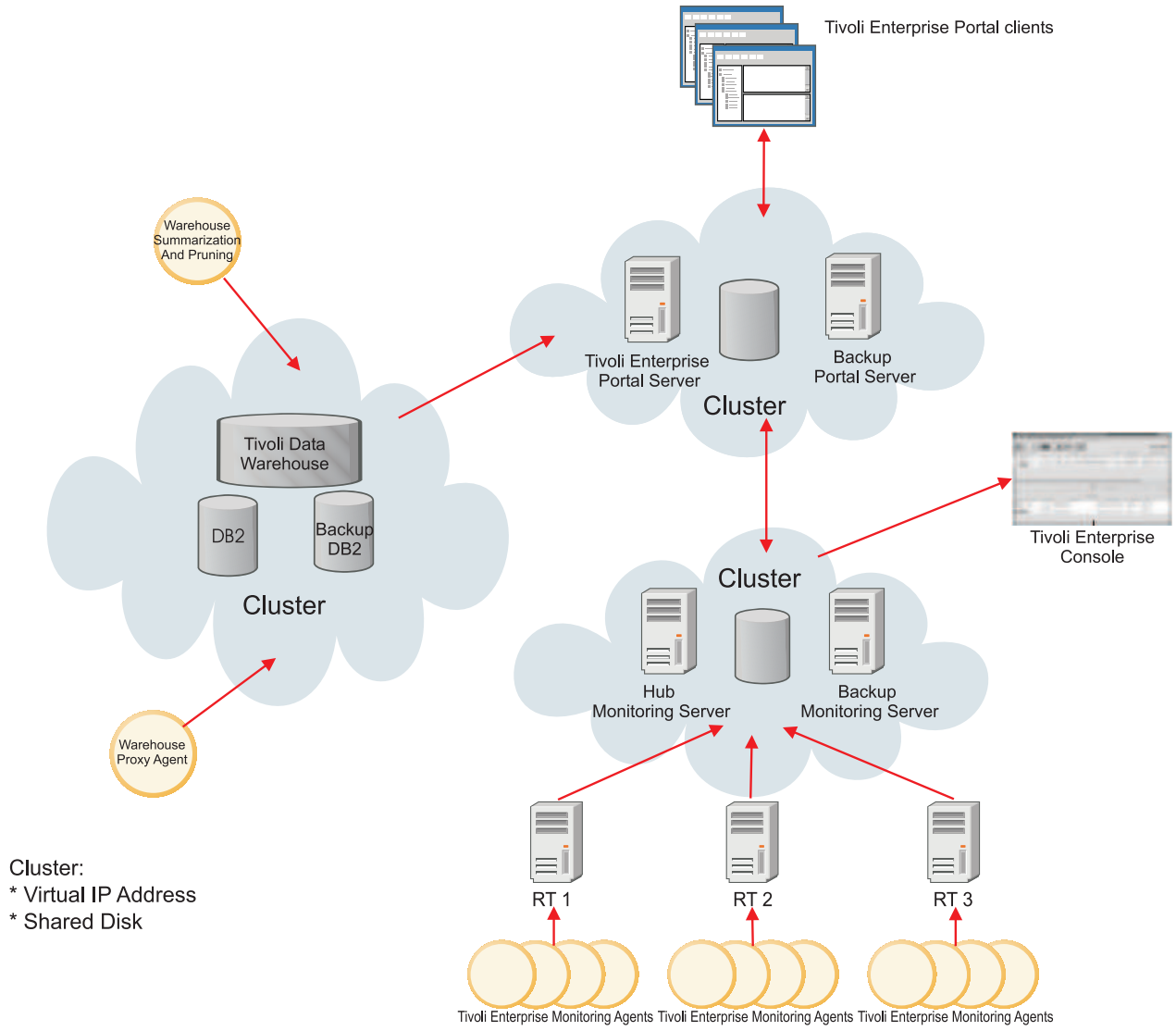

Figure 12. Separate component clusters, with the warehouse proxy agent and Summarization and Pruning Agents outside the data warehouse cluster

Configuration C

In this configuration, all of the main components are clustered in one single cluster. There is also one agent running on the clustered environment that directly reports to the hub; this agent has not been included in the picture for simplification.

This configuration represents a smaller scale deployment that is similar to Configuration A. You must ensure that the computers used for such an environment have enough power to handle all the components. Although the behavior of the clustered components do not change, the setup of IBM Tivoli Monitoring on such an environment has some differences.

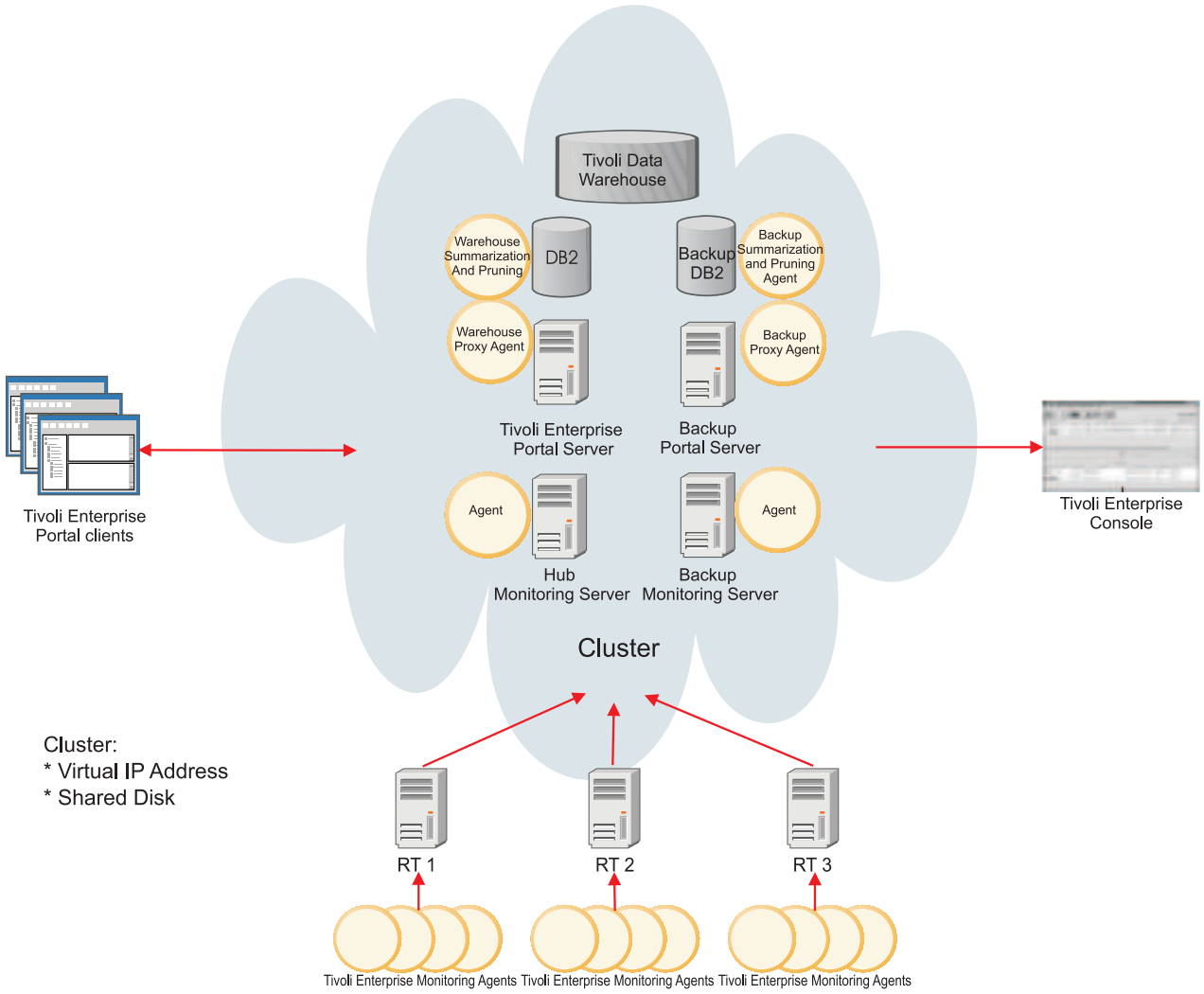

Figure 13. One cluster for the hub, portal server, and data warehouse

Setting up Tivoli Monitoring components in a clustered environment

This section describes the overall procedure for setting up IBM Tivoli Monitoring components on clustered environments.

IBM Tivoli Monitoring cluster setup

A basic cluster setup includes two computers that participate as nodes in the cluster. Each computer must have two Network Interface Cards (NICs): one for the heartbeat function between the two computers, the other for public access.

The Tivoli Monitoring requirements for the cluster server are:

- A shared persistent data storage for the IBM Tivoli Monitoring installation directory and for any component-specific data.
- A virtual IP address to be assigned to the component that runs on the cluster.

Some important characteristics of a clustered IBM Tivoli Monitoring setup include:

- The shared persistent storage and virtual IP Address must be available before you can install IBM Tivoli Monitoring; also, the node where Tivoli Monitoring is installed must own these resources.
- Install an IBM Tivoli Monitoring component (such as the hub monitoring server) only once on the first node of the cluster (on the shared directory); then, configure it on the second node. In the Windows environment, this means that the IBM Tivoli Monitoring registry keys must be replicated to the second node.
- The IBM Tivoli Monitoring component must be bound to the virtual IP address such that the infrastructure's other IBM Tivoli Monitoring components can reach it no matter which node it is running on. To do this, set the network interface where the component listens on.
- Only one instance of the component can be active at a time.
- When IBM Tivoli Monitoring is running under the cluster, its services cannot be automatically started. Likewise, its services cannot be directly started or stopped through the command line or through the Manage Tivoli Enterprise Monitoring Services GUI. Instead, the services must be controlled through the cluster manager.

The following table shows the high-level steps required to set up the main Tivoli Monitoring components running in a cluster.

Table 3. Basic steps to set up Tivoli Monitoring on a cluster

Step	Hub monitoring server	Portal server	Data warehouse (Summarization and Pruning Agent and Warehouse Proxy Agent optional)
1		Install the database software. Configure the database software for the cluster.	Install the database software. Configure the database software for the cluster.
2	Create the cluster. Define the resource group with resources: • Virtual IP • Shared persistent storage	Create the cluster. Define the resource group with resources: • Virtual IP • Shared persistent storage • Database	Create the cluster. Define the resource group with resources: • Virtual IP • Shared persistent storage • Database
3	Install and set up the monitoring server on the first node of the cluster. Set up the monitoring server on the second node of cluster.	Install and set up the portal server on the first node of the cluster. Set up the monitoring server on the second node of cluster.	Install and set up the Summarization and Pruning Agent and Warehouse Proxy Agent on the first node of the cluster. Set up the Summarization and Pruning Agent and Warehouse Proxy Agent on the second node of the cluster.
4	Add the monitoring server as a resource to the resource group.	Add the portal server as a resource to the resource group.	Add the Summarization and Pruning Agent and Warehouse Proxy Agent as a resource to the resource group.

Monitoring server setup

About this task

The monitoring server cluster resources include:

- Shared persistent storage

- Virtual IP address
- Monitoring server service

The generic setup of the hub monitoring server on a cluster includes the following steps:

Procedure

1. Set up the basic cluster resource group with a shared persistent storage and virtual IP address.
2. Install the monitoring server on the first node of the cluster (shared persistent storage).
3. Remove automatic startup of the monitoring server service (if applicable to your platform).
4. Bind the monitoring server to the virtual IP address.
5. Set up the monitoring server on the second node of the cluster (depending on the platform, this might involve copying registry keys, environment variables, or both to the second node).
6. Add the monitoring server as a resource to the resource group.

   Note: The specific setup of the monitoring server on different platforms and cluster managers has variations.

Portal server setup

About this task

The portal server cluster includes the following resources:

- Shared persistent storage
- Virtual IP address
- Database middleware (such as DB2)
- Portal server service

The generic setup of the portal server on a cluster includes the following steps:

Procedure

1. Install the database middleware locally on both nodes.
2. Set up the database users and groups to be exactly the same on both nodes.
3. Remove automatic startup of the database middleware, so it can run under cluster control.
4. Set up the basic cluster resource group with shared persistent storage, virtual IP address, and the database middleware.
5. Create the portal server database on the first node of the cluster (shared persistent storage).
6. Catalog the portal server database on the second node of the cluster.
7. Install the portal server on the first node of the cluster (shared persistent storage).
8. Disable autostart of the portal server service (if applicable to your platform).
9. Bind the portal server to the virtual IP address.
10. Set up the monitoring server on the second node of the cluster (depending on the platform, this might involve copying registry keys, environment variables, or both to the second node).
11. Add the portal server as a resource to the resource group.

    Note: The setup of the portal server on different platforms and cluster managers has variations; in this document, the setup is described for the specific cluster managers mentioned in this document.

Data warehouse setup

The data warehouse cluster includes the following resources:

- Shared persistent storage

- Virtual IP address
- Database middleware (such as DB2)
- Warehouse Proxy Agent and Summarization and Pruning Agent processes (optional)

The generic setup of the data warehouse on a cluster involves the following steps:

1. Install the database middleware locally on both nodes.
2. Set up the database users and groups to be exactly the same on both nodes.
3. Remove automatic startup of the database middleware, so it can run under cluster control.
4. Set up the basic cluster resource group with shared persistent storage, virtual IP address, and the database middleware.
5. Create the data warehouse database on the first node of the cluster (shared persistent storage).
6. Catalog the data warehouse database on the second node of the cluster.

The following steps are necessary only if the Summarization and Pruning Agent and the Warehouse Proxy Agent are included in the cluster. Otherwise, these steps are optional:

1. Install the Summarization and Pruning Agent and the Warehouse Proxy Agent on the first node of the cluster (shared persistent storage).
2. Disable automatic startup of the Summarization and Pruning Agent and Warehouse Proxy Agent services (if applicable to this platform).
3. Bind the Summarization and Pruning Agent and the Warehouse Proxy Agent to the virtual IP address.
4. Set up the Summarization and Pruning Agent and Warehouse Proxy Agent on the second node of the cluster (depending on your platform, this might involve copying the registry keys or environment variables or both to the second node).
5. Add the Summarization and Pruning Agent and Warehouse Proxy Agent as resources to the resource group.

What to expect from the IBM Tivoli Monitoring infrastructure in a clustered environment

In general, when a failover or failback of a clustered component occurs, the Tivoli Monitoring components operate as if the clustered element has been restarted.

Clustered hub monitoring server

When the hub server is configured as a cluster, and failover or failback occurs, the connected Tivoli Monitoring components operate as if the hub has been restarted. When failover or failback occurs, the other components automatically reconnect to the hub and some synchronization takes place.

After reconnection, as part of the remote monitoring server to hub synchronization, all situations that are the responsibility of the remote monitoring server (distributed to the monitoring server itself or to one of its connected agents) are restarted. This restarting of situations represents the current behavior for all reconnection cases between remote monitoring servers and the hub, despite the clustered environments. See “Situations” on page 46 for more information.

For agents directly connected to the hub, there might be periods in which situation thresholding activity on the connected agents does not occur because, when a connection failure to the reporting hub is detected, the situations are stopped. As soon as the connection is reestablished, the synchronization process takes place and situations are restarted. (Note that historical metric collection is not stopped.)

The portal server, the Summarization and Pruning Agent, and the warehouse proxy agent reconnect to the hub and perform any synchronization steps necessary.

The occurrence of a hub restart depends on the size of the environment (including the number of agents and situations). Initially, you are notified that the portal server has lost contact with the monitoring server, and views might be unavailable. When the portal server reconnects to the hub, the Enterprise default workspace is displayed, allowing access to the Navigator Physical view. However, some agents might have delays in returning online (due to the reconnection timers) and trigger polled situations again (when the situations are restarted by the agents).

While the hub failover or failback (including the startup of the new hub) might be quick (approximately 1-3 minutes), the resynchronization of all elements to their normal state might be delayed in large-scale environments with thousands of agents. This behavior is not specific to a cluster environment, but valid anytime the hub is restarted, or when connections from the other Tivoli Monitoring components to the hub are lost and later reestablished.

Clustered portal server

When the portal server is clustered and failover or failback occurs, the connected Tivoli Monitoring components operate as if the portal server has been restarted. The components that connect to the portal server include the portal consoles and the Summarization and Pruning Agent (at the start of its summarization and pruning interval).

When a portal client loses connection to the portal server, the user is notified and some views become unavailable. When the portal client reestablishes connection with the portal server, the home workspace is displayed and the Navigator refreshes.

If the Summarization and Pruning Agent is connected to the portal server at the time of portal server failover, it loses the connection and attempts to reestablish contact on the next Summarization and Pruning Agent interval.

Clustered data warehouse

When the data warehouse is set up as a cluster, the Tivoli Monitoring components connected to the data warehouse respond to the data warehouse failover or failback as a database restart. The components that connect to the data warehouse database include the summarization and pruning agent, Warehouse Proxy Agent, and portal server (when retrieving long term data collection for the portal workspace views).

When the Summarization and Pruning Agent loses contact with the data warehouse database, it attempts to reestablish contact on the next summarization and pruning agent interval and then restart its work.

When the Warehouse Proxy Agent loses contact with the data warehouse database, it attempts to reestablish the connection and restart its work.

When the portal server loses contact with the data warehouse, it reconnects on the next query request from a portal client.

Clustered Summarization and Pruning Agent

When the data warehouse cluster resource group includes the clustered Summarization and Pruning Agent, the agent fails together with the data warehouse database and must be restarted after the data warehouse. While the Summarization and Pruning Agent uses transactions for its operations to the data warehouse database, it resumes its summarization and pruning work where it left off prior to failure.

Clustered Warehouse Proxy Agent

When the Warehouse Proxy Agent is part of the data warehouse cluster resource group, the proxy agent fails together with the data warehouse database and must be restarted after the data warehouse. The Warehouse Proxy Agent then resumes the work of uploading the short-term data collection to the data warehouse.

Clustered agentless monitoring

With the Agentless Monitoring Server now maintaining connections to hundreds of servers, it becomes a more critical component in the infrastructure than a single agent instance.

The following two options are available:

- **Option 1:** Run the Agentless Monitoring Server within an Operating System cluster. The cluster provides the desired HA functionality.
- **Option 2:** Develop a set of custom situations, take action commands, and scripts in IBM Tivoli Monitoring to natively monitor the Agentless Monitoring Server status.
  - Two Agentless Monitoring Servers are configured, with environment files modified to register to IBM Tivoli Monitoring with the same system name. The primary Agentless Monitor is started, while the backup Agentless Monitor is offline.
  - Periodic scripting is done to synchronize the configurations between the two Agentless Monitors.
  - If the Primary Agentless Monitor goes offline, the self-monitoring situation triggers, forwarding information to the event console for problem determination.
  - A Take Action command associated with the situation issues a start command to the backup Agentless Monitor.

When you use Agentless Monitoring, a percentage of preparation time needs to be devoted to verifying the native data emitter configurations. Complete the following preparation tasks:

- Ensure that the SNMP daemons are installed, configured, and started
- Verify the MIB branches in SNMP configuration files
- Verify all Windows passwords and user account rights for Windows API collection
- Verify that the following patch levels for endpoint systems are installed:
  - AIX and SLES systems are most likely to require SNMP patches
  - Solaris CIM-XML systems are most likely to require CIM patches
  - RedHat, HP-UX, and Windows agents typically operate as delivered. If possible, use tools such as snmpwalk, WMIExplorer, and perfmon to verify the metrics are configured before configuring IBM Tivoli Monitoring to monitor the environments

Note: Single-computer installations have no problems for small environments where you want to monitor less than 50 remote systems.

For information on Agentless monitoring, see the *IBM Tivoli Monitoring: Installation Guide*.

Situations

Situations might be affected by failures on the hub, the remote monitoring server, and at the agent. During a hub cluster failover, situations are affected in the same way as a restart of the hub or a disconnection of the hub from the other components.

When a remote monitoring server loses connection with the hub and then reestablishes contact, the server synchronizes after reconnection. This process involves restarting all the situations under that remote monitoring server's responsibility. Polled situations are triggered on the next polling interval, but pure events that were opened before the failure are lost. Use an event management product, such as Tivoli NetCool/OMNIbus or Tivoli Enterprise Console, as the focal point for storing and manipulating historical events from all event sources.

If you have agents directly connected to the hub, the situations distributed to them are stopped when connection is lost, and the situations are restarted when the agent reconnects to the hub. This behavior also applies for agents that are connected to a remote monitoring server and lose connection to it.

Workflow policies

Workflow policies can be set to run at the hub. If the hub fails over while a workflow policy is running, the processing stops and then restarts at the beginning of the workflow policy (upon restart of the hub and triggering).

Short-term data collection

When a hub failover or failback occurs, remote monitoring servers and agents reconnect to it, causing all situations to be restarted.

Short-term data collection is performed at the agents through special internal situations called UADVISOR. These situations are also restarted and collect data only after the next full interval has passed, resulting in the loss of the data gathered in one collection interval.

Long-term data collection

The Warehouse Proxy and Summarization and Pruning Agents connect to the hub. After a hub failover or failback occurs, the proxy and agent reconnect to it without impacting their work on the data warehouse.

Tivoli Enterprise Console event integration

When Tivoli Monitoring is configured to forward events to the Tivoli Enterprise Console event server, it sends a "MASTER_RESET" event to the event server every time the hub is restarted. This behavior also occurs after failover or failback for a clustered hub. The purpose of this event is to signal to the event server that the monitoring server restarted all its situations and that the event server receives a new set of currently triggered situations. As a result of the default Tivoli Enterprise Console rule for Tivoli Monitoring integration, all the events that came from the hub Tivoli Enterprise Monitoring server are closed. Although this closing of events ensures that both Tivoli Monitoring and Tivoli Enterprise Console have consistent triggered situations and events, this behavior might not be desirable in some environments. For example, this behavior closes pure events that are not reopened until the next pure event occurs. In such cases, you can filter out the MASTER_RESET event either at the hub (event integration facility configuration file) or at the event server by using the Tivoli Enterprise Console rules. In this case, the rules must process duplicate events caused by the restarted situations.

Maintenance

The maintenance requirements of a Tivoli Monitoring environment (patches, fix packs, or release upgrades) when running in a cluster resemble those of running an unclustered Tivoli Monitoring environment. The cluster controls the starting and stopping of Tivoli Monitoring services. To return this control to the installation procedure, you must stop the cluster while the maintenance is being performed. Also, due to current restrictions of the Tivoli Monitoring installation procedures, some settings that are completed during Tivoli Monitoring cluster setup might need to be repeated after the maintenance is completed and before the cluster is restarted.

In a Tivoli System Automation for Multiplatform environment, you will need to switch off the automatic clustering when upgrading your environment. By switching off the clustering, you prevent the components, stopped by the installer, from restarting during the upgrade. However, you must keep the basic resources online (such as VIP and shared disk).

Also, when using the UNIX or LINUX installer, you must remove the inittab entries again after maintenance. The installer adds the inittab entries again despite having been removed in the previous installation of the cluster. The failure to remove the inittab entries might cause problems in your environment.

Chapter 6. Creating clusters with Tivoli Monitoring components in an HACMP environment

This chapter provides information on installing and configuring IBM Tivoli Monitoring components in High Availability Cluster Multiprocessing (HACMP) environments under the AIX operating system.

In the environment described, each cluster is set up with a single quorum device and two nodes. For HACMP-specific cluster terminology, see the *HACMP Concepts and Facilities Guide* .

Important: The following configurations and operational procedures for high availability have been tested by IBM Software Support and represent supported product functions. Other configurations or procedures are possible but might be outside of product design parameters; therefore, other configurations or procedures might not operate correctly or be supported.

Preparing for the base cluster

Before defining the base cluster, you must gather cluster nodes information and check the cluster nodes environment as described in the following sections.

Gathering cluster nodes information

Before defining the base cluster, you need the following information:

- The host names of both cluster nodes, for example:

  ```
  clusternode1 clusternode2
  ```

  Typically, failover clusters have a private pair of network interface cards (NICs) for their heartbeat communications and a public pair for communication with the outside environment. Therefore, you need the host names of the private NICs. It is not necessary to have these host names defined on the domain name system (DNS); they can be defined only in the local host tables (/etc/hosts) of each node.
- The mount point name of the shared file system. The name must be the same on both nodes. Example of the mount point name:

  ```
  /shareddisk
  ```
- The device name of the shared disk that will be mounted to that mount point. The name must be the same on both nodes. Example of the device name:

  ```
  dev/sdc1
  ```
- The type of file system on the shared disk, for example:

  ```
  jfs
  ```
- The virtual IP address and netmask.
- The Ethernet numbers of the public network cards of both nodes, which will host the virtual IP address, for example:

  ```
  en0 on clusternode1
  en3 on clusternode2
  ```
- The interface name. To determine the interface name, run the ***ifconfig*** command.

Checking the cluster nodes environment

Before you install the base cluster, check the environment thoroughly:

- Ensure that the domain name system for all network interface cards resolves correctly.

- Ensure that all entries for host names (in the local host table /etc/hosts) are set in the same order on both computers. For example, if clusternode1 /etc/hosts contains:

  ```
  10.0.0.1 host.domain.com host
  ```

  The `/etc/hosts` on clusternode2 should also contain

  ```
  10.0.0.1 host.domain.com host
  ```

  and not:

  ```
  10.0.0.1 host host.domain.com
  ```

- Ensure that the host tables have no double entries for the **local host**. Under SUSE SLES10, using 2 NICs, you might see the following two entries:

  ```
  localhost
  localhost
  ```

  If you find a second entry, be sure to remove it. If there are double entries, Tivoli Monitoring installation fails.
- Ensure that the domain name system works for the virtual IP address that will be used for the cluster. (The provided scripts assume that the virtual IP address is on the same subnet as the nodes' IP addresses.)
- Verify that the future virtual IP address is a valid address.
- Ensure that this IP address is valid for the configured addresses on the public network interface cards of the cluster nodes (same range and subnet).
- Verify, on both nodes, that the shared disk will not be mounted automatically during startup.

Defining the base cluster for Tivoli Monitoring

To install the Tivoli Monitoring platform components (hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, or Tivoli Data Warehouse components) on an HACMP cluster, you first need to set up a base cluster resource group with some prerequisite resources.

The base HACMP cluster resource group needed for the hub monitoring server includes the following resources:

- A shared file system for the monitoring server installation and data.
- A virtual (service) IP Address for the monitoring server.

The base HACMP cluster resource group needed for the portal server and data warehouse components includes the following resources:

- A shared file system for the portal server and data warehouse components installation.
- A virtual IP address for the portal server and data warehouse components.
- A relational database, which is a prerequisite for both the portal server and the data warehouse.

Building a base HACMP cluster for the monitoring server

To build a base HACMP cluster for the monitoring server, you must configure the following components:

- HACMP resource group for the monitoring server
- Service IP address for the monitoring server
- File system for the monitoring server
- Monitoring server resource group

Also verify your cluster configuration after completing these configurations.

Configuring an HACMP resource group for the monitoring server

Use the `smit hacmp` command to configure the resource group for the monitoring server.

Configuring a service IP address for the monitoring server

Make sure that the domain name system resolves the virtual (service) host name correctly. Create a resource by using the virtual IP address for the service IP address. (See the *HACMP for AIX Administration Guide*.)

Configuring a file system for the monitoring server as a shared resource

Create a file system on the shared disk, for example, `/sharedisk/IBM/ITM`. Also, create a cluster shared resource for this shared file system. (See the *HACMP for AIX Administration Guide*.)

Creating a monitoring server resource group

Create a resource group for the monitoring server using a unique resource group name. Assign the defined resources to this resource group. (See the *HACMP for AIX Administration Guide*.)

Verifying and synchronizing the cluster configuration

Verify your cluster configuration after the previous steps have been completed. Fix any errors found during verification before proceeding. (See the *HACMP for AIX Administration Guide*.) Review the `smit.log` (SMIT console messages) and ensure that the cluster services were started correctly and that all nodes are synchronized.

Building a base HACMP cluster for the portal server and data warehouse components

For the portal server and data warehouse components (Warehouse Proxy Agent and the Summarization and Pruning Agent), you need one additional resource, the underlying DB2 database.

In this case, you first install the IBM DB2 database locally on each node. The database is created later, when the cluster is up and running.

Installing DB2 for the base HACMP cluster

About this task

There are several ways to install a DB2 database in a clustered environment; the configuration described in this section is one of many scenarios.

The basic cluster resources required for this DB2 configuration, the shared file system, and the virtual IP address, might already be online on the active node. These resources will be required post-installation.

For installing the DB2 server, the DB2 Enterprise Server Edition is used.

Install the DB2 database and its services locally on each cluster node. Accept the default directory as the installation directory. Ensure that the DB2 user IDs are created using the same group ID (GID) and user ID (UID) on each node.

During the installation, create an instance, usually `db2inst1`.

Procedure

1. Turn off the automatic startup of the DB2 server on each node.
2. As the root user:

   ```
   cd sqllib/instance
   ./db2iauto -off db2inst1
   ```
3. Edit the file /etc/inittab on both nodes and comment out this line:

   ```
   fmc:2345:respawn:/opt/IBM/db2/V9.7/bin/db2fmcd #DB2 Fault Monitor Coordinator
   ```
4. Stop the DB2 on each node. To do so, switch to the user ID of the instance owner and run this command:

   ```
   db2stop
   ```

Creating the database for the portal server or data warehouse on clusternode1

About this task

The last step is to create the database, either for the portal server or for the data warehouse:

Procedure

1. Bring the basic cluster resources online on clusternode1 if you have not already done so.
2. Create a directory for the database on the shared file system:

   ```
   /sharedFS/DBdirectory
   ```
3. Grant full access to or ownership of this directory to the DB2 instance owner, usually to the user db2inst1.
4. Switch to the user ID of the instance owner.
5. Create a database using the DB2 CLI on the active node, using one of these commands:
   - for the portal server:

     ```
     db2 create database <DBName> on /sharedFS/DBdirectory
     ```
   - for the data warehouse:

     ```
     db2 create database <DBName> on /sharedFS/DBdirectory using codeset UTF-8
     territory US
     ```
6. Test the connection to the database using the command:

   ```
   db2 connect to <DBName>
   ```
7. After a successful connection, reset the connection with this command:

   ```
   db2 connect reset
   ```
8. Create a database user ID on both nodes with the same GID and UID, and grant full access to the database to this user ID on clusternode1. Because of the same IDs it should now also work for the same user on clusternode2.

Cataloging the portal server and the data warehouse database on clusternode2

About this task

To catalog the portal server and the data warehouse database on clusternode2, complete the following procedure:

Procedure

1. Shut down the DB2 on clusternode1 and switch the basic resource group to clusternode2.
2. Catalog the database (both for the portal server and the Tivoli data warehouse as the case might be) in this instance with this command:

```
db2 catalog database <DBName> on /sharedFS/DBdirectory
```

3. Test the connection to the database (see above).
4. Switch back to the root user.
5. Test the connection to the database (see above).
6. Now shut down the DB2 database on clusternode2.

What to do next

If you want to install a portal server in the cluster and the data warehouse is on another computer, also catalog the data warehouse's database server for this local DB2 as a remote database server on both nodes, so that the portal server can access this database later. For further information on this, consult the *IBM Tivoli Monitoring: Installation and Setup Guide* on the Tivoli Monitoring Information center.

Adding the database to the base cluster

About this task

To add the database to the base cluster, complete the following procedure:

Procedure

1. Add the database to cluster resource group.
2. Take the basic resource group offline using `smit hacmp`.
3. Create a start/stop script (or two scripts, one for each purpose) in the same local directory on each cluster node.

   The following examples show how the start/stop scripts can be created:

   ```
   db2start.sh
   DB2BINDIR=/opt/IBM/db2/V9.7/bin
   su db2inst1 "-c $DB2BINDIR/db2 db2start"
   ```

   ```
   db2stop.sh
   DB2BINDIR=/opt/IBM/db2/V9.7/bin
   su db2inst1 "-c $DB2BINDIR/db2 db2stop"
   ```

   Important: Ensure that the scripts have the executable flag set.
4. Create an application server resource for the DB2 using `smit hacmp`.
5. Add the application server resource to the basic cluster resource.
6. Synchronize the cluster.
7. Bring the cluster resource group online.

Results

DB2 now runs under the control of the HACMP, and you are now ready to install the portal server, data warehouse, or Summarization and Pruning Agent.

Installing the monitoring server on its base HACMP cluster

This section provides the information for the installation of the monitoring server on its base HACMP cluster.

Installing and setting up the monitoring server on clusternode1

Before installing the monitoring server, determine the directory on the shared file system into which the monitoring server will be installed. Also, decide on a HUBNAME for the monitoring server. You will need this HUBNAME during the monitoring server configuration.

These parameters have to be set during the installation of the monitoring server, and you will later need them to set the desired variables in the start/stop/monitor script for the monitoring server.

Assume you have the following environment:

- DNS 1st box: host1 (active cluster)
- DNS name 2nd box: host2
- Virtual host name (of the virtual network interface): virtualhost

When you have the base cluster for the monitoring server up and running, use the following procedures to install the monitoring server:

- "Upgrading the monitoring server installation"
- "Pristine installation"

Upgrading the monitoring server installation

Procedure

1. Determine which cluster node runs the resource group online by using the `smit hacmp` command.
2. On the *active* node, stop the monitoring server:

   ```
   cd /sharedisk/IBM/ITM/bin
   ./itmcmd stop server HUB_VIRTUALHOST
   ```
3. Start IBM Tivoli Monitoring installation and upgrade the Tivoli Enterprise Monitoring Server to latest version.
4. Reconfigure the monitoring server.

   You can change any settings except for the "TEMS Host Name". The "TEMS Host Name" value has to be preserved, pointing to the virtual host name ("virtualhost").

   ```
   cd /sharedisk/IBM/ITM/bin
   ./itmcmd config -S -t HUB_VIRTUALHOST
   ```
5. Create two symbolic links in the same directory named "host1_ms_HUB_VIRTUALHOST.config", "host2_ms_HUB_VIRTUALHOST.config" to file "virtualhost_ms_HUB_VIRTUALHOST.config":

   ```
   ln -s virtualhost_ms_HUB_VIRTUALHOST.config host1_ms_HUB_VIRTUALHOST.config
   ln -s virtualhost_ms_HUB_VIRTUALHOST.config host2_ms_HUB_VIRTUALHOST.config
   ```
6. Start the monitoring server and add application support for the desired agent types to the monitoring server. For ux, lz, and nt:

   ```
   cd /sharedisk/IBM/ITM/bin
   ./itmcmd server start HUB_VIRTUALHOST
   ./itmcmd support -t HUB_VIRTUALHOST ux lz nt
   ```

   Note: For an easier installation, install the application support for all planned agents at the same time. If you also need database agents, use the database agent installation image to install the application support for them.
7. Recycle the monitoring system:

   ```
   ./itmcmd server stop HUB_VIRTUALHOST
   ./itmcmd server start HUB_VIRTUALHOST
   ```
8. Test the monitoring server on clusternode1 and clusternode2 following the steps provided in "Testing the monitoring server on clusternode1" on page 56 and "Setting up the monitoring server on clusternode2" on page 56.

Pristine installation

Procedure

1. Determine which cluster node runs the resource group online using `smit hacmp`.
2. On the *active* node, install the monitoring server into your chosen directory on the shared file system, as described in the *IBM Tivoli Monitoring: Installation and Setup Guide*.
3. Install the monitoring server on host1 (the active cluster) into `/sharedisk/IBM/ITM` (the mounted file system of step1).

Important: During installation, you are asked for the monitoring server name (HUB_VIRTUALHOST). Make note of the installation directory and the HUBNAME for further use.

4. Specify the virtual host name of the cluster when configuring the Tivoli enterprise monitoring server host name.
5. Configure the monitoring server by using the following command:

   ```
   ./itmcmd config -S -t HUB_VIRTUALHOST
   ```

   During the configuration when you are asked for the monitoring server host name, enter:

   ```
   virtualhost
   ```

6. You are prompted to determine if you want to specify the Primary Network Adapter. Type Y(es) and set it to virtualhost.

   The virtualhost sets the KDCB0_HOSTNAME variable in the KBBENV file of this monitoring server, which is needed for binding onto the virtual adapter. This sets the primary network adapter to the virtual host name of the cluster, which ensures that the agents and the portal server can later connect to the monitoring server on this IP address.
7. Start up the monitoring server and add the application support for the desired agent types to the monitoring server. For ux, lz, and nt:

   ```
   ./itmcmd server start HUB_VIRTUALHOST
   ./itmcmd support -t HUB_VIRTUALHOST ux lz nt
   ```

   It is best to install the application support for all planned agents at the same time.

   If you also need database agents, use the database agent installation image to install the application support for them.
8. Recycle the monitoring server with these commands:

   ```
   ./itmcmd server stop HUB_VIRTUALHOST
   ./itmcmd server start HUB_VIRTUALHOST
   ```

Tivoli Enterprise Monitoring Server reconfiguration procedure for the AIX HACMP environment

About this task

When Tivoli Enterprise Monitoring Server reconfiguration is needed, complete the following procedure:

Procedure

1. Configure the monitoring server:

   ```
   cd /sharedisk/IBM/ITM/bin
   ./itmcmd config -S -t HUB_VIRTUALHOST
   ```

   Adjust settings as needed, however keep "TEMS Host Name" set to "virtualhost".
2. Create two symbolic links in the same directory named “host1_ms_HUB_VIRTUALHOST.config”, “host2_ms_HUB_VIRTUALHOST.config” to file “virtualhost_ms_HUB_VIRTUALHOST.config”:

   ```
   ln -s virtualhost_ms_HUB_VIRTUALHOST.config host1_ms_HUB_VIRTUALHOST.config
   ln -s virtualhost_ms_HUB_VIRTUALHOST.config host2_ms_HUB_VIRTUALHOST.config
   ```

3. Recycle the monitoring system:

   ```
   ./itmcmd server stop HUB_VIRTUALHOST
   ./itmcmd server start HUB_VIRTUALHOST
   ```

4. Test the monitoring server on clusternode1 and clusternode2. Test the monitoring server on clusternode1 and clusternode2, following the same steps as for pristine installation. See “Testing the monitoring server on clusternode1” on page 56 and “Testing the monitoring server failover to clusternode2” on page 61.

Testing the monitoring server on clusternode1

The following sections provide information on testing the monitoring server on clusternode1.

Setting up the monitoring server cluster configuration

Perform additional verification of your cluster configuration by testing the monitoring server failover scenarios:

- Manually migrate resource groups through the System Management (C-SPOC) panel under the Manage HACMP Services SMIT panel.
- Manually stop services (can also be done through the System Management (C-SPOC) panel) on the higher priority node.
- Create a custom test plan to automate the monitoring server failover testing. (See *HACMP for AIX Administration Guide*.)

Connect one OS agent from any supported system (ux, lz, or nt) to this monitoring server, pointing onto the host name `virtualhost`.

You can connect a portal server to this monitoring server or use the CLI:

- Log on to the monitoring server using `./tacmd login`
- List the managed systems using `./tacmd listsystems`

You see the hub monitoring server and the agent as two managed systems.

Setting up the monitoring server on clusternode2

Procedure

1. Shut down the monitoring server on clusternode1.
2. Start `smit hacmp` and switch the cluster to host2.
3. Log on at host2. You should see `/sharedisk/IBM/ITM` with the installed monitoring server.
4. Change to the following directory:
   `/sharedisk/IBM/ITM/config`
5. You find **two** configuration files for the monitoring server:
   ```
   host1_ms_HUB_VIRTUALHOST.config
   virtualhost_ms_HUB_VIRTUALHOST.config
   ```
6. Create a symbolic link to this file in the same directory:
   `host1_ms_HUB_VIRTUALHOST.config`
   Name of the link:
   `host2_ms_HUB_VIRTUALHOST.config`

   This is required. Otherwise, the startup of the monitoring server fails.

What to do next

Now you can start the monitoring server on host2 with this command:

```
./itmcmd server start HUB_VIRTUALHOST
```

Check the managed systems with `./tacmd listsystems`. After a few minutes, you see that the agent is online again. If the monitoring server is configured to validate users IDs, ensure that the users are registered with the local operating system on both computers with the same user name as the one used to log on to the portal server and a password.

Adding the monitoring server to the resource group of the base cluster

After the installation of the monitoring server is complete, you now have to integrate it into your base cluster so that it can run under the cluster's control.

Creating start and stop scripts for the monitoring server

Create a start script and stop script in the same local directory on each node. The script can resemble this script:

```
export CANDLEHOME=/sharedisk/IBM/ITM
/$CANDLEHOME/bin/itmcmd server start HUB_VIRTUALHOST
```

The start and stop scripts contain the export of the CANDLEHOME variable, which is set to the installation directory of the monitoring server, such as `/opt/IBM/ITM`. The second line of the script runs the start command of the monitoring server. The stop script is almost identical except the stop command replaces the start command. Be sure that the scripts have the executable flag set.

Adding the monitoring server as an application server to the base cluster

Now you can create a new application server using `smit hacmp`. When it is created, stop the resource group that already contains the shared file system and the virtual IP address of the monitoring server, add the monitoring server resource as a new application server to this group, and synchronize the cluster.

Important: Do not try to add the server to the group while it is running.

Bring the resource group online. The monitoring server should be up and running.

Adding HACMP monitoring for the monitoring server processes

Use HACMP monitoring to monitor the processes of any defined HACMP application server. In the event that the processes are not running, an attempt is made to restart the component. Failover from the application server to the other node occurs if the attempt fails; this behavior depends on the script that is defined to do the recovery action.

Because the monitoring server has two processes that must be running, both processes should be stopped to restart the monitoring server. The two Tivoli Enterprise Monitoring Server processes are *kdsmain* and *cms*.

A modified start script, that first checks if one or both processes are running, might resemble the following script. Note that the following parameters were used on the monitor setup:

- Monitor Mode: *both*
- Owner: *root*
- Instance count: *1*
- Stabilization interval: *60*
- Restart count: default *3* (depends on your environment)
- Restart interval: 3 x 60 = *180*
- Action on application failure: *fallover*
- Restart method: *startscript*

Additional scripts that call this script with the correct parameters must be created.

For additional information on how to set up an application server process monitor, see the *HACMP for AIX Administration Guide*.

A modified start script:

```
#!/bin/bash
#-------------------------------------------------------------------
#
# HACMP: HA start/stop/status script for <TEMSSRV>
#
# Control <TEMSSRV>
# ==================
#
# This script is to be used as Start/Stop/MonitorCommand
# Invocation:
#
# <$0>.ksh <Action> itm_installation_directory hubname
#
# arg $1 <Action> is any one of: {start|stop|status}
# arg $2 ITM installation directory
# arg $3 ITM hub name
#-------------------------------------------------------------------
# intended IBM.Application definition
# PersistentResourceAttributes::
# Name="SA-<TEMSSRV>-rs"
# ResourceType=1
# NodeNameList="{${clusternode1},${clusternode2}}"# StartCommand=
# StartCommandTimeout=
# StopCommand=
# StopCommandTimeout=
# MonitorCommand=
# MonitorCommandPeriod=
# MonitorCommandTimeout=
# UserName=root
# RunCommandsSync=1
#-------------------------------------------------------------------
## Static
#-------------------------------------------------------------------
INVOCATION="$0 $@"
Myname=`/bin/basename $0`
USAGE="Usage: ${Myname} [start|stop|status] itm_installation_directory hubname"
STATUS_UNKNOWN=0STATUS_ONLINE=1
STATUS_OFFLINE=2
STATUS_FAILED_OFFLINE=3
STATUS_STUCK_ONLINE=4
STATUS_PENDING_ONLINE=5
STATUS_PENDING_OFFLINE=6

STARTSTOP_OK=0
STARTSTOP_ERR=1

RC=0
#-------------------------------------------------------------------
## Arguments
NumArgs=3
#-------------------------------------------------------------------
Action=${1:-status}
CANDLEHOME=${2}
HUBNAME=${3}
#-------------------------------------------------------------------
## Var (non-configurable)
#-------------------------------------------------------------------
## ...
#-------------------------------------------------------------------
## Var (configurable)
#-------------------------------------------------------------------
# SYSLOG_LVL - 0 is least detailed, 2 is most detailed.
# written to syslog
SYSLOG_LVL=1

#--- Verify the OS and set absolute paths to the binaries
OS=`uname -s`
```

```
OSVERSION="`uname -v`"
# do distro stuff
if [ -a /etc/SuSE-release ]
then
         loggerPATH=/bin
else
         loggerPATH=/usr/bin
fi
case $OS in
  AIX)
     INST_INTERP="AIX"
     BIN="/usr/bin"
     AWK_CMD="$BIN/awk"
     CAT_CMD="$BIN/cat"
     DATE_CMD="$BIN/date"
     PS_CMD="$BIN/ps"
     SU_CMD="$BIN/su"
     GREP_CMD="$BIN/grep"
     TEE_CMD="$BIN/tee"
     PIDMON_CMD="$BIN/pidmon"
     LOGGER_CMD="$BIN/logger"
     KILL_CMD="$BIN/kill"
     ;;
  Linux)
     USRBIN="/usr/bin"
     BIN="/bin"
     AWK_CMD="$BIN/gawk"
     DATE_CMD="$BIN/date"
     CAT_CMD="$USRBIN/cat"
     PS_CMD="$BIN/ps"
     SU_CMD="$BIN/su"
     KILL_CMD="$BIN/kill"
     GREP_CMD="$USRBIN/grep"
     TEE_CMD="$USRBIN/tee"
     PIDMON_CMD="$USRBIN/pidmon"
     if [ -a /etc/SuSE-release ]; then
       LOGGER_CMD="$BIN/logger"
     else
       LOGGER_CMD="$USRBIN/logger"
     fi
     case `uname -m` in
       *390*)
          INST_INTERP="LINUX_S390"
          ;;
       *86*)
     INST_INTERP="LINUX_I386"
     ;;
         *)
     INST_INTERP="LINUX_OTHER"
     ;;
      esac
      ;;
esac
#--------------------------------------------------
# function: logit
# arg $1 log level
# arg $2 message
#--------------------------------------------------
function logit {
   if [ $SYSLOG_LVL -ge $1 ]; then
      echo ${Myname} "$2"
      ${LOGGER_CMD} -i -t ${Myname}: "$2"
   fi
} #logit
#------------------------------------------------------------------------
## Main Section
#------------------------------------------------------------------------
```

```
if [ $# != ${NumArgs} ]; then
   echo ${USAGE}
   logit 0 "Bad Usage returning: 0"
   exit 0
fi

export CANDLEHOME
BINARCH=$CANDLEHOME/*/sy/bin
export BINARCH
export HUBNAME

case ${Action} in
   start)
         logit 1 "Start command issued"
                kdsmainproc=$($PS_CMD -ef | $AWK_CMD '/kdsmain/ &&
!/awk/ {print $2}')
                cmsproc=$($PS_CMD -ef | $AWK_CMD '/cms start/ &&
!/awk/ {print $2}')
                restart=0
                start=1
                if [[ $kdsmainproc != "" ]]; then
                   if [[ $cmsproc != "" ]]; then
                    start=0
                  else
                    $KILL_CMD -9 $kdsmainproc
                    start=1
                   fi
               else
                 if [[ $cmsproc != "" ]]; then
                    $KILL_CMD -9 $cmsproc
                    start=1
                 fi
              fi

              if [[ $start = 1 ]]; then
                 $SU_CMD - root -c "$CANDLEHOME/bin/itmcmd server
start $HUBNAME"
                 RC=$?
              else
                 RC=0
              fi
        logit 0 "Start command returned: $RC"
        ;;
   stop)
        logit 1 "Stop command issued"
              kdsmainproc=$($PS_CMD -ef | $AWK_CMD '/kdsmain/ &&
!/awk/ {print $2}')
              cmsproc=$($PS_CMD -ef | $AWK_CMD '/cms start/ &&
!/awk/ {print $2}')
              if [[ $kdsmainproc != "" ]]; then
                 if [[ $cmsproc != "" ]]; then
                    $SU_CMD - root -c "$CANDLEHOME/bin/itmcmd server
stop $HUBNAME"
                 else
                    $KILL_CMD -9 $kdsmainproc
                 fi
               else
                 if [[ $cmsproc != "" ]]; then
                    $KILL_CMD -9 $cmsproc
                 fi
               fi
               RC=0
      logit 0 "Stop command returned: $RC"
      ;;
   status)
      logit 2 "Status command issued"
```

```
      cmsprocstat=$($PS_CMD -ef | $AWK_CMD '/kdsmain/ && !/awk/
{print $2}')

      if [[ $cmsprocstat != "" ]]; then
               # the kdsmain process is running
               echo "cms running"
               cmsStatus=1
      else
               # the kdsmain process isn't running
               cmsStatus=2
      fi
      kdsmainproc=$($PS_CMD -ef | $AWK_CMD '/kdsmain/ &&
!/awk/ {print $2}')

      start=1;

      if [[ $kdsmainproc != "" ]]; then
              # the kdsmain process is running
              kdsStatus=1
      else
              # the kdsmain process isn't running
              kdsStatus=2
      fi
      if [[ $cmsStatus = "1" && $kdsStatus = "1" ]]; then
         # HACMP expects 0 if application running
         RC=0;
      else
         # and non-zero if not running
         RC=2;
      fi
   logit 2 "Status command returned: $RC"
   ;;
*)
   RC=${UNKNOWN}
   echo ${USAGE}
   logit 0 "Bad Action returning: ${RC}"
   ;;
esac

exit $RC
```

Testing the monitoring server failover to clusternode2

Use `smit hacmp` to stop services on the primary node, or use the cluster test tool to automate the monitoring failover testing. Similarly, test the failover of services from the secondary node back to the primary node.

Note: If you are actively using IBM Tivoli Monitoring (navigating workspaces, and so forth) when the failover occurs, it might take up to 10 minutes for the portal server and the agents to reconnect to the monitoring server due to portal server and agent internal timeout intervals.

Installing the portal server on its base HACMP cluster

To install the portal server on its base HACMP cluster, you need to use the installation procedures in the following sections.

Installing and setting up the portal server on clusternode1

About this task

When the DB2 database is up and running, the installation of the portal server follows the instructions in the *IBM Tivoli Monitoring: Installation and Setup Guide*.

To set up the portal server on clusternode1, complete the following steps:

Procedure

1. Make sure that the resource group is online and the shared file system, virtual IP address, and the DB2 are available on clusternode1.
2. Run the installation script `install.sh` from the product image as described in the *IBM Tivoli Monitoring: Installation and Setup Guide*.

   Make sure you install the portal server into the shared file system. For easier handling of the configuration, also install the IBM Tivoli Monitoring Service Console.

   After installation, export the DISPLAY variable to point to your workstation. If you have an X server running, you can use the graphical IBM Tivoli Monitoring Service Console to configure the portal server. Otherwise, use the command line configuration.
3. Configure the portal server interface and URL such that the portal clients can connect to portal server on its virtual host name. Add the following line to the `<CANDLEHOME>/config/cq.ini` file:

   `KFW_INTERFACE_cnps_HOST=<virtualhostname>`

   where `<virtualhostname>` is the virtual host name of the portal server.
4. Use one of the following 2 methods to configure the portal server:

   To invoke a GUI application, use `<CANDLEHOME>/bin/itmcmd manage`, which requires a connection to an X Server.

   To invoke a text application, use `<CANDLEHOME>/bin/itmcmd config –A cq`, which prompts you for the configuration values.

   Note: If you are configuring the portal server on Linux on zSeries in 64-bit mode, you need to run the command from a 31-bit session. To open a 31-bit session, run the following command:

   ```
   s390 sh
   ```
5. Configure the monitoring server host name so that it is the same as the virtual host name of the monitoring server (assuming the hub monitoring server is clustered).
6. Configure the portal server to use the virtual host name.
   a. Make sure Tivoli Enterprise Portal Server and Tivoli Enterprise Portal Server Extensions are stopped.
   b. Back up your configuration data located in the `$CANDLEHOME/<architecture>/iw/profiles` directory.
   c. Run the UNIX terminal.
   d. Change the current directory to `$CANDLEHOME/<architecture>/iw/scripts`.
   e. Run `sh updateTEPSEHostname.sh <old_hostname> <new_hostname>`.

      Under <old_hostname> substitute the current host name, do not include domain name.

      Under <new_hostname> substitute the valid virtual host name for your Tivoli Enterprise Portal Server cluster.

      Note: If you repeat the `updateTEPSEHostname`script execution, make sure to use the correct "old_hostname". For example, the following sequence it correct:

      `updateTEPSEHostname.bat h1 crocodile`, the ewas host name was changed to "crocodile".

      `updateTEPSEHostname.bat crocodile hippi`, as old host name, you must use "crocodile" or the script will have no effect.
   f. If the operation is successful, 'BUILD SUCCESSFUL' is displayed.
   g. Set the Primary Network Name to the virtual host name of the portal server. This configuration is necessary for the communication between the portal server and the rest of the Tivoli Monitoring infrastructure, including the monitoring server.

The Primary Network Name in the **Network Name** field, which is available after selecting **Specify Optional Primary Network** on the GUI **TEMS Connection** tab, and the **Optional Primary Network Name** field in the text application.

7. Configure the portal server database on the shared file system in the database settings of the portal server.

   After successful configuration, start the portal server manually and check the connection via portal client.

8. Stop the portal server, and remove the autostart of the portal server.

   Under AIX, the installer creates a file under `/etc`, all named `rc.itm1`. This file has to be removed to ensure that the portal server does not start up during reboot.

9. Change **ServerName** in `<CANDLEHOME>/<platform>/iu/ihs/conf/httpd.conf` to clustername.

Testing the portal server on clusternode1

Start the portal server and make sure it connects to the monitoring server. Connect a Tivoli Enterprise Portal client to the portal server and ensure that agent workspaces are populated.

Setting up the portal server on clusternode2

Procedure

1. Move the resource group to clusternode2.
2. Install the GSKit locally on clusternode2 (for further instructions also see "Installing and setting up the Warehouse Proxy Agent and Summarization and Pruning Agent on clusternode1" on page 64 for the monitoring server).
3. Start the portal server manually on clusternode2 and test it like you did on clusternode1.
4. Stop the portal server on clusternode2.

Adding the portal server to the resource group of the base cluster

Having completed the installation of the portal server, you now have to integrate it into your base cluster, so that it can then run under the cluster's control. (See *HACMP for AIX Administration Guide*).

Creating start and stop scripts for the portal server

Procedure

1. Use `smit hacmp` to take the basic resource group offline.
2. Create a start/stop script (or two scripts, one for each purpose) in the same local directory on each cluster node.
3. The scripts look very similar to the control scripts of the monitoring server; they should, at least, have these entries:

   ```
   export CANDLEHOME=/sharedisk/IBM/ITM
   /$CANDLEHOME/bin/itmcmd agent start cq
   ```

4. For the stop script, use the stop parameter.

What to do next

Make sure that the scripts have the executable flag set.

Adding the portal server as Application Server to base cluster

Procedure

1. Use `smit hacmp` to create an application server resource for the portal server.
2. Make sure the resource group is offline, and then add the application server resource to the basic cluster resource with DB2, shared file system, and IP address.
3. Synchronize the cluster.

4. Bring the cluster resource group online.

Adding monitoring for the portal server process

Use HACMP monitoring to monitor the processes of any defined HACMP application server. In the event that the processes are not running, an attempt is made to restart the component. Failover from the application server to the other node occurs if the attempt fails; this behavior depends on the script that is defined to do the recovery action. Because the portal server has only one process, you can use the start script to restart it again if needed.

The name of the portal server process for monitoring is *KfwServices*.

For more information on setting up an application server process monitor, refer to *HACMP for AIX Administration Guide*.

Testing the portal server failover to clusternode2

Use `smit hacmp` to stop cluster services on the primary node (this should force a failover), or use the cluster test tool to automate the monitoring failover testing. Similarly, test the failover of services from the secondary node back to the primary node.

Installing the Warehouse Proxy Agent or Summarization and Pruning Agent on its base HACMP cluster

To install the Warehouse Proxy Agent and the Summarization and Pruning Agent follow the same steps described for the portal server.

Installing and setting up the Warehouse Proxy Agent and Summarization and Pruning Agent on clusternode1

Because the Warehouse Proxy Agents and the Summarization and Pruning Agents are using JDBC on UNIX and Linux, the TCP/IP listener for the DB2 has to be configured correctly on both nodes.

To do so, follow the steps in the *IBM Tivoli Monitoring: Installation and Setup Guide*, "Tivoli Data Warehouse solutions: common procedures."

When this is done, install the Warehouse Proxy Agent and the Summarization and Pruning Agent using the same procedure mentioned above for the portal server (See "Installing the portal server on its base HACMP cluster" on page 61).

- For the Warehouse Proxy:

  Edit the `/sharedisk/ITM/IBM/config/hd.config` and specify the virtualhost in the CTIRA_HOSTNAME property value.
- For Summarization and Pruning:

  Edit the `/sharedisk/ITM/IBM/config/sy.config` and specify the virtualhost in the CTIRA_HOSTNAME property value.

Testing the Tivoli Data Warehouse components in the cluster

Verify that Warehouse Proxy Agent is correctly exporting data to Tivoli Data Warehouse and Summarization and Pruning Agent is correctly summarizing and pruning.

Setting up the Warehouse Proxy Agent and Summarization and Pruning Agent on clusternode2

Procedure

1. Move the resource group to clusternode2.

2. Install the GSKit locally on clusternode2 (for further instructions also see “Setting up the monitoring server on clusternode2” on page 81 for the monitoring server).
3. Start the Warehouse Proxy Agent and Summarization and Pruning Agent manually on clusternode2 and test it like you did on clusternode1.
4. Stop the Warehouse Proxy Agent and the Summarization and Pruning Agent on clusternode2.

Adding the Warehouse Proxy Agent and the Summarization and Pruning Agent to the resource group of the base cluster

When the installation of the Warehouse Proxy Agent and the Summarization and Pruning Agent is done, integrate it into your base cluster, so that it can then run under the cluster's control.

Creating start and stop scripts for the Warehouse Proxy Agent and the Summarization and Pruning Agent

About this task

The only difference is in the start and stop scripts for the application server under HACMP due to the different product codes:

Table 4. Component product codes

Component	Product code
Portal server	`cq`
Warehouse Proxy Agent	`hd`
Summarization and Pruning Agent	`sy`

Procedure

1. Take the basic resource group offline using `smit hacmp`.
2. Create a start/stop script (or two scripts, one for each purpose) in the same local directory on each cluster node.

   The scripts look very similar to the control scripts of the portal server. For the Warehouse Proxy Agent, for example, the script should have, at least, these entries:

   ```
   export CANDLEHOME=/sharedisk/IBM/ITM
   $CANDLEHOME/bin/itmcmd agent start hd
   ```

   for the Summarization and Pruning Agent:

   ```
   export CANDLEHOME=/sharedisk/IBM/ITM
   $CANDLEHOME/bin/itmcmd agent start sy
   ```

   For the stop script, use the stop parameter.

   Ensure the scripts have the executable flag set.

Adding the Warehouse Proxy Agent and the Summarization and Pruning Agent as Application Server to base cluster

Procedure

1. Create an application server resource for each of the Warehouse Proxy Agent and the Summarization and Pruning Agents using `smit hacmp`.
2. Make sure the resource group is offline, and then add the application server resource to the basic cluster resource with DB2, shared file system and IP address.
3. Synchronize the cluster.
4. Bring the cluster resource group online.

Adding monitoring for the Warehouse Proxy Agent and the Summarization and Pruning Agent process

Use HACMP monitoring to monitor the processes of any defined HACMP application server. In the event that the processes are not running, an attempt is made to restart the component. Failover from the application server to the other node occurs if the attempt fails; this behavior depends on the script that is defined to do the recovery action.

The monitoring is defined for each defined application server.

Because the Warehouse Proxy Agent, as well as the Summarization and Pruning Agent, have only one process each and also because they are defined as separated application servers, you can use the start scripts to restart them again if needed.

The name of the Warehouse Proxy Agent process for monitoring is:

```
khdxprtj
```

The name of the Summarization and Pruning Agent process is:

```
ksy610
```

For further information on how to set up an application server process monitor, refer to *HACMP for AIX Administration Guide*.

Testing the Warehouse Proxy Agent and the Summarization and Pruning Agent failover to clusternode2

Use `smit hacmp` to stop cluster services on the primary node (this should force a failover), or use the cluster test tool to automate the agent failover testing. Similarly, test the failover of services from the secondary node back to the primary node.

Move the resources from clusternode1 to clusternode2, and verify that Warehouse Proxy Agent and Summarization and Pruning Agent restart their processing.

Known problems and limitations

It is important to remember specific characteristics and constraints of Tivoli Monitoring installation and set up, and their effects on the cluster setup.

During the certification test for Tivoli Monitoring clustering, issues encountered when setting up the clustered environment are formally reported as defects. These defects are typically related to the setup of Tivoli Monitoring in a non-default manner, instead of being specific to the cluster environment. These defects are handled as part of the Tivoli Monitoring service stream. Here is a list of known problems and workarounds:

- The Tivoli Monitoring installer configures the components to be autostarted by default. It does not give the user an option to configure the components to not autostart. Under this limitation, the user has to edit an operating system script to remove this behavior.

  This same behavior occurs whether installing for the first time or applying Fix Packs.
- GSKit is a prerequisite of Tivoli Monitoring. As of the current release of Tivoli Monitoring, this component can only be installed into a specific location: `/usr/local/ibm/gsk7`. Under this limitation, GSKit cannot be installed into a user defined location, such as the file system for the shared disk.

  The workaround is to use the Tivoli Monitoring installation program to install GSKit 15 on both nodes.
- If there are user console activities being performed, such as clicking an object on the workspace to request data, while the failover process is occurring, then the console takes a longer period of time to become active. This delays the time it takes for the agent to become available to the console user. This is due to the internal delay timers for the reconnection algorithm.

- There are some IBM Tivoli Monitoring components that are dependent on other components, like the Tivoli Enterprise Portal Server requiring DB2. In this case, you will either want to create a cluster Application Server that includes all of your IBM Tivoli Monitoring components with their dependencies, or write scripts that locks the processes that are waiting on their dependencies, otherwise the cluster will fail on startup.
- If the clustered Tivoli Enterprise Portal Server does not start after failover/switch, follow the following instructions:
  1. Make sure Tivoli Enterprise Portal Server starts successfully on the first computer in your cluster.
  2. Switch to the first cluster computer.
     a. For Windows:

        Change the current directory to `%CANDLE_HOME%\CNPSJ\profiles\ITMProfile\config\cells\ITMCell\nodes\ITMNode`.

     b. For UNIX or Linux systems:

        Change the current directory to `$CANDLE_HOME/<architecture>/iw/profiles/ITMProfile/config/cells/ITMCell/nodes/ITMNode`.
  3. Locate and open serverindex.xml file.
  4. Check if all "hostName" and "host" properties are equal to virtual host name for your Tivoli Enterprise Portal Server cluster. Any invalid value must be changed to the correct value.
  5. For each unique and invalid host name found in step 5, apply steps 2–5 accordingly.
- In order to configure the Ethernet adapters for communication by IBM Tivoli Monitoring, you can use the configuration parameter: KDEB_INTERFACELIST. See 2 on page 133 for information on how to remove this entry in the registry.
- In order for Java Web Start to work, you must change all occurrences of $HOST$ to your fully qualified host name in the .jnlpt file. You can locate the .jnlpt file in the `CANDLE_HOME\config` directory.

Chapter 7. Creating clusters with monitoring components in a System Automation for Multiplatforms environment

In this chapter, the implementation and design of high-availability IBM Tivoli Monitoring Linux and AIX environments are described in conjunction with IBM Tivoli System Automation for Multiplatforms. Guidance is provided for high-availability strategies using IBM Tivoli Monitoring components. Practical considerations regarding design, implementation, testing, and maintenance are also discussed.

"Preparing for the base Tivoli Monitoring cluster with Tivoli System Automation for Multiplatform" on page 70 lists the Tivoli Monitoring requirements for High Availability (HA) when used in conjunction with Tivoli System Automation for Multiplatforms. The requirements of Tivoli System Automation for Multiplatforms to automate Tivoli Monitoring are discussed in "Installing Tivoli System Automation for Multiplatforms on the cluster nodes" on page 73. For guidance on implementing highly available strategies when using Tivoli Monitoring components on Windows, see Chapter 8, "Creating clusters with Tivoli Monitoring components in a Microsoft Cluster Server environment," on page 93.

Tivoli System Automation for Multiplatforms provides high availability by automating the control of resources such as processes, file systems, IP addresses, and other resources in clusters. The product facilitates the automatic switching of users, applications, and data from one system to another in the cluster after a hardware or software failure. A complete High Availability (HA) setup includes many parts, one of which is the HA software. As well as tangible items such as hardware and software, a good HA solution includes planning, design, customizing, and change control. An HA solution reduces the amount of time that an application is unavailable by removing single points of failure. For more information visit the following website: http://www-306.ibm.com/software/tivoli/products/sys-auto-multi/.

Important: These configurations and operational procedures for high availability have been tested by IBM and represent supported product functions. Other configurations or procedures are possible but might be outside of product design parameters; therefore, other configurations or procedures might not operate correctly or be supported.

Scenarios tested

For clustering purposes, Tivoli Monitoring is divided into three primary components: Tivoli Enterprise® Monitoring Server (monitoring server), Tivoli Enterprise Portal Server (portal server), and Tivoli Data Warehouse. The monitoring server component is the hub Tivoli Enterprise Monitoring Server. The portal server component includes the Tivoli Enterprise Portal Server and Portal Server database. The Tivoli Data Warehouse component consists of the Data Warehouse database, Warehouse Proxy Agent, and Summarization and Pruning Agent. As the DB2 agent can be used for the Portal Server database, consider adding support for it.

Because of the variety of configurations that are available with Tivoli Monitoring and Tivoli System Automation for Multiplatforms, all combinations are not tested.

Note: The information provided in this document has been tested and updated with Tivoli System Automation for Multiplatforms version 3.1.0.0 for IBM Tivoli Monitoring V6.2.2.

Table 5. Scenarios Tested

Hardware	Operating System	Database	Tiebreaker type	Component Locations
xSeries®	SLES10	DB2 9.1	SCSI	Monitoring server, portal server, and Tivoli Data Warehouse components on separate systems
xSeries	SLES10	DB2 9.1	Network	Monitoring server, portal server, and Tivoli Data Warehouse components on separate systems
xSeries	SLES10	DB2 9.1	Network	Monitoring server, portal server, and Tivoli Data Warehouse components on the same system
xSeries	RHAS4	DB2 9.1	Network	Monitoring server, portal server, and Tivoli Data Warehouse components on the same system
xSeries	RHAS4	DB2 9.1	SCSI	Monitoring server, portal server, and Tivoli Data Warehouse components on the same system
xSeries	AIX 5.3	DB2 9.1	Network	Monitoring server, portal server, and Tivoli Data Warehouse components on the same system
xSeries	SLES9	DB2 9.1	Network	Monitoring server, portal server, and Tivoli Data Warehouse components on the same system
xSeries	SLES9	DB2 9.1	ECKD™	Monitoring server, portal server, and Tivoli Data Warehouse components on the same system

Preparing for the base Tivoli Monitoring cluster with Tivoli System Automation for Multiplatform

This section provides information on preparing for the base Tivoli Monitoring with Tivoli System Automation for Multiplatforms cluster.

Gathering cluster nodes information

About this task

Before defining the Base cluster, you must have the following information available:

Procedure

1. The host names of both cluster nodes, such as:

   ```
   node1
   node2
   ```

2. In order to set up Tivoli Monitoring with Tivoli System Automation for Multiplatforms, you will need two cluster nodes with at least 1 NIC (Network Interface Card) in each computer.

For a more robust cluster, use two NICs for each computer, so that the Tivoli System Automation for Multiplatforms heartbeat function has a failover path, in case one of the NICs goes down. The NICs you use for System Automation for Multiplatforms must be on the same subnet.

You will need the host names of the NICs. It is not necessary to have these host names defined in the domain name system, they only need to be defined in the local host table of each node.

3. The mount point name of the shared file system - must be the same on both nodes (and mounted only on one node), such as:

   `/shareddisk`

4. The device name of the shared disk which is mounted to that mount point – it also must be the same on both nodes, such as:

   `/dev/sdc1`

   - During testing on Red Hat Enterprise Linux, the **Logical Volume Manager** was used to define the shared disk. The following syntax was used for the device name (only for shared drives created with the **Logical Volume Manager**):

     `/dev/mapper/`*`volume_group_name-lv_name`*

     Where *`volume_group_name`* is the name of the volume group containing the shared disk logical volume, and *`lv_name`* is the name of the logical volume. For example: if the volume group name is shared_vg and the logical volume name is *`shared_lv`*, the device name would be:

     `/dev/mapper/shared_vg-shared_lv`

   - On AIX, the following syntax was used for the device name:

     `/dev/lv_device_name`

     If the logical volume device name is `datalv`, the device name would be:

     `/dev/datalv`

5. The type of the file system on the shared disk, such as:

   For Linux: `reiserfs`

   For AIX: `jfs2`

6. The virtual IP address and netmask (must be in the same subnet as the real IP addresses)
7. The Ethernet numbers of the public network interfaces of both nodes, which will host the virtual IP address, such as:

   Running Linux: `eth0`

   Running AIX: `en1`

   (Run the `ifconfig` command to determine the Ethernet numbers)

Checking the cluster nodes environment

Before you install the Base cluster, check the environment thoroughly:

- Ensure that the name resolution for all network interface cards works properly
- Ensure that all entries for host names in the local host table, `/etc/hosts`, are set in the same order on both computers. For example, if `/etc/hosts` on node1 contains:

  10.0.0.1 host.domain.com host

  `/etc/hosts` on node2 should also contain:

  10.0.0.1 host.domain.com host

  And **not**:

  10.0.0.1 host host.domain.com

- Ensure that the host tables have no double entries for the `localhost`. Under SUSE SLES10, using two network interface cards, you might see the following two entries:

  ```
  localhost
  localhost
  ```

  Remove the second entry, otherwise, clustering fails.

- Ensure that the name resolution works for the virtual IP address that will be used for the cluster (the provided scripts assume that the virtual IP address is on the same subnet as the IP addresses of the nodes).
- Verify that the virtual IP address of the cluster is a valid address, to avoid conflicts.
- Verify that this Virtual IP address is not defined (doesn't show up in `ifconfig – a`) on any node in the cluster.
- Ensure that this IP address is valid for the configured addresses on the public network interface cards of the cluster nodes.
- Verify on both nodes that the shared disk will not be mounted automatically during startup.
- Verify on both nodes that the shared disk file system is not checked on start up of the node (`fsck` is not run).

Planning for the cluster tiebreaker network device

For a two-node Tivoli System Automation for Multiplatforms cluster, a *tiebreaker* is required. A tiebreaker is a network device (on Linux, Linux on zSeries®, and AIX systems), a SCSI device (on AIX and Linux systems), or an ECKD device (on Linux on zSeries systems), which is accessible from both nodes.

The tiebreaker is used when the cluster splits, and the nodes don't have contact with each other anymore. In this rare case, the tiebreaker makes sure that only one node has communication with the shared resources. This communication protects the resources from concurrent access through both cluster nodes.

Network tiebreaker

For the Network tiebreaker on Linux, Linux on zSeries, or AIX systems, determine the IP address of the device that is to be used for the tiebreaker. Make sure that this device is accessible from both nodes.

The Network tiebreaker is the easiest to implement because it requires only an IP address that is accessible by both nodes. The Network tiebreaker must also be on the same subnet as both nodes.

(To use the Network tiebreaker, use the **net** parameter for the `itm6/BuildScripts/mkitmcluster.sh` script.)

SCSI tiebreaker on Linux systems

For the SCSI tiebreaker on Linux, for both nodes, the following parameters for the SCSI device are required:

SCSI Host number

Channel number

ID number

LUN number

To determine the values, run the following command (on both nodes):

```
dmesg | grep 'Attached scsi disk'
```

If this command doesn't return anything (because the messages might have been overwritten in the buffer), then run this command:

```
cat /proc/scsi/scsi
```

The output shows the exact parameters of each SCSI device attached to the system.

The SCSI device must be accessible to both nodes.

(To use the SCSI tiebreaker, use the **scsi** parameter for the `itm6/BuildScripts/mkitmcluster.sh` script.)

SCSI tiebreaker on AIX systems

For the SCSI tiebreaker on AIX, you need the name of the hard disk device (/dev/hdiskX). The hard disk device must support the SCSI protocol.

Note: Drives attached via Fiber Channel, iSCSI, and Serial Storage Architecture support the SCSI protocol, and can be used as tiebreaker devices.

IDE drives do not support the SCSI protocol, and can **not** be used as tiebreaker devices.

Use the following command to list all physical volumes:

```
lspv
```

Use the following command to verify that the specific drive supports SCSI:

```
lsdev -C -l <devicename>
```

(To use the Disk tiebreaker, use the **disk** parameter for the `itm6/BuildScripts/mkitmcluster.sh` script.)

ECKD tiebreaker on Linux on zSeries systems

For the ECKD tiebreaker on Linux on zSeries, you need the ECKD device number. To gather this information, run the following command:

```
cat /proc/dasd/devices
```

The ECKD device number is in the first column.

(To use the ECKD tiebreaker, use the **eckd** parameter for the `itm6/BuildScripts/mkitmcluster.sh` script.)

For further information, see the *IBM Tivoli System Automation for Multiplatforms Base Component Administrator's Guide* and the *Reliable Scalable Cluster Technology (RSCT) Administrator's Guide*.

Installing Tivoli System Automation for Multiplatforms on the cluster nodes

About this task

To set up Tivoli Monitoring with Tivoli System Automation for Multiplatforms, you will need two cluster nodes with at least one NIC (Network Interface Cards) in each computer. For more information regarding the configuration of the cluster creation, see Appendix A, "Configuring the cluster creation," on page 137.

For a more robust cluster, use two NICs for each computer, so that the Tivoli System Automation for Multiplatforms heartbeat function has a failover path, in case one of the NICs goes down.

You should then execute the following steps on each of the cluster nodes (depending on how you received Tivoli System Automation for Multiplatforms, you might have it already installed):

Procedure

1. From the Tivoli System Automation for Multiplatforms version 3.1.0.0 image or CD, run the `prereqSAM.sh` command to verify that you have the correct prerequisite software. If all prerequisites are present continue to step 2. Otherwise install the required prerequisites. (`prereqSAM.sh` displays the missing prerequisites, and writes to `/tmp/prereqSAM.1.log`.)

   Note: Installing Tivoli System Automation for Multiplatforms on AIX is different from installing it on Linux mainly because Tivoli System Automation for Multiplatforms for AIX does not come with the prerequisite packages.
2. From the Tivoli System Automation for Multiplatforms version 3.1.0.0 image or CD, run the `installSAM.sh` command.

Note: If you are using Tivoli System Automation for Multiplatforms version 2.2.0.0, follow the fix pack installation instructions to install the latest Tivoli System Automation for Multiplatforms Version 2.2.0.0 Fix Pack (Fix Pack 2 was used for testing.)

3. If the nodes will be running DB2 in the cluster, download the DB2 policies `db2salinux.tar.gz`, from

   ftp://ftp.software.ibm.com/software/data/db2/linux/

   Expand the file. A suggested location is `/usr/sbin/rsct/sapolicies` because this is where other products might install their policies.
4. Download additional policies:
   a. For AIX, download additional policies `sam.policies.aix.1.2.2.1`, from:

      http://www-01.ibm.com/software/brandcatalog/portal/opal/details?catalog.label=1TW10SA01

      To install the policies, use the System Management Interface Tool (more commonly known as SMIT). The policies install to`/usr/sbin/rsct/sapolicies`.
   b. For Linux:

      For Linux on Intel: `sam.policies-1.2.2.1-06212.s390.rpm`

      For Linux: http://www-01.ibm.com/software/brandcatalog/portal/opal/details?catalog.label=1TW10SA02

      To install the policies, use the `rpm –i filename` command. The policies install to `/usr/sbin/rsct/sapolicies`.

   Use the latest available version available on the site.

What to do next

Make sure that you run the steps provided above on both cluster nodes.

Note: The Tivoli System Automation for Multiplatforms Version 2.2.0.0 Fix Pack procedure allows you to install the fix pack by either bringing down the entire cluster, or by upgrading the cluster node by node.

With the first method, all of the resources are taken offline, which means that the Tivoli Monitoring components in that cluster is stopped.

With the second method, one node is taken offline and upgraded, while the active node is left online. If a failure occurs on the active node, failover does not occur until the inactive node has been brought online.

For further information, refer to the "Installing and uninstalling service" chapter of the *Tivoli System Automation for Multiplatforms Installation and Configuration Guide*, and refer to "Applying a fix pack to the Summarization and Pruning Agent" on page 90 for an issue with the service installation documentation.

Creating a cluster with all Tivoli Monitoring Components

About this task

In the case of a demo or proof of concept, you might want to create a cluster that contains the four IBM Tivoli Monitoring components on one system. This is not recommended for a production environment. To accomplish this, complete the steps in the following table:

Table 6. Creating a cluster containing all four IBM Tivoli Monitoring components

Task	Section in this document
1. Build a base cluster including DB2	"Setting up a cluster for Tivoli Monitoring" on page 75
2. Install and set up the monitoring server	"Installing the monitoring server on its base Tivoli System Automation for Multiplatforms cluster" on page 80
3. Install and set up the portal server	"Installing the portal server on the Tivoli System Automation for Multiplatforms cluster" on page 82

Table 6. Creating a cluster containing all four IBM Tivoli Monitoring components (continued)

Task	Section in this document
4. Install and set up the Warehouse Proxy Agent	"Installing the Warehouse Proxy Agent on a Tivoli System Automation for Multiplatforms cluster" on page 84
5. Install and set up the Summarization and Pruning Aent	"Installing the Summarization and Pruning Agent on a Tivoli System Automation for Multiplatforms cluster" on page 86

When the previous steps have been completed:

Procedure

1. On one of the nodes, take the resource group RG_NAME offline:
   `chrg –o offline RG_NAME`
2. Verify that the resource group is offline, which can take 1 or 2 minutes:
   `lssam -top`
3. Switch to the `itm6/BuildScripts` directory.
4. Run the `allrscbuild.sh` command as root from the same directory.
5. Start the resource group again using the command:
   `chrg –o online RG_NAME`
6. Verify that the resource group is online:
   `lssam –top`
7. Test the failover by running the `rgreq –o move RG_NAME` command, and use `lssam -top` to verify that monitoring server starts on the other node. If a component doesn't start:
   a. Fix the values entered in the `itm6/BuildScripts/clustervariables.sh` file
   b. Remove the Tivoli System Automation for Multiplatforms domain using the following commands:
      `stoprpdomain –f domain_name rmrpdomain domain_name`
   c. Re-execute the steps in "IBM Tivoli Monitoring cluster setup" on page 41.

   Note: The terminal session might end after the move completes.

Setting up a cluster for Tivoli Monitoring

To install Tivoli Monitoring components (monitoring server hub, portal server or Tivoli Data Warehouse components) on a Tivoli System Automation for Multiplatforms cluster, you need to first set up a Base cluster resource group with some prerequisite resources.

The Base Tivoli System Automation for Multiplatforms cluster resource group needed for the monitoring server hub includes:

- A shared File System for the installation and data
- A virtual IP Address

The Base Tivoli System Automation for Multiplatforms cluster resource group needed for portal server and Tivoli Data Warehouse components includes:

- A shared File System for installation
- A virtual IP Address
- A Relational Database, which is a prerequisite for both portal server and Tivoli Data Warehouse.

Predefined Tivoli System Automation for Multiplatforms Cluster for Tivoli Monitoring

To simplify the setup of a Tivoli System Automation for Multiplatforms cluster for Tivoli Monitoring, a set of predefined definitions, also called policies, are provided in a "cluster package" for Tivoli Monitoring. Note that DB2 includes Tivoli System Automation for Multiplatforms. If you select "install SAMP", the latest Tivoli System Automation for Multiplatforms and DB2 policies are provided.

This package consists of scripts to build and run the cluster with all required resources as quickly and easily as possible. The "cluster package" can be downloaded from: http://www-01.ibm.com/software/brandcatalog/portal/opal/details?catalog.label=1TW10TM4F. Note that these scripts are provided "as is", without support.

Depending on what you want to run in the Tivoli System Automation for Multiplatforms cluster (monitoring server, portal server or Tivoli Data Warehouse components); there are different steps to define and to start up the cluster. With this package, the base cluster can be defined and started in less than 10 minutes.

The package comes in a tar file named `itm6.sam.policies-2.0.tar`, which has to be installed *locally* in the same directory on *each* future cluster node. This installation ensures that all required resources like control scripts etc. are available on all nodes with the same file path. The directory on all nodes are:

```
/<untar-directory>/itm6
```

A suggested location is `/usr/sbin/rsct/sapolicies` because this is where other products might install their policies.

The tar file contains the following directories:

`itm6/BuildScripts`
: Contains scripts used to create and configure the cluster. See Appendix C, "Predefined scripts," on page 145 for details about the scripts.

`itm6/ControlScripts`
: Contains scripts that are used by Tivoli System Automation for Multiplatforms to control the Tivoli Monitoring components.

  Note: These scripts have "root" hard-coded into them. If you do not run ITM as "root", modify the scripts.

`itm6/DefFiles`
: This is initially empty, but is populated with resource definition files, by the `itm6/BuildScripts/generateclusterfiles.sh` script.

Note: Do not modify this directory structure. Clustering will not work correctly if this is changed.

There are two different types of a Base cluster, depending on which Tivoli Monitoring components you want to install:

1. Base cluster with a shared file system and IP Address.
2. Base cluster with shared file system, IP Address, and DB2. The DB2 resource is used if the cluster contains the portal server and/or Tivoli Data Warehouse components.

Both types of base clusters are built with the same basic steps:

1. Prepare the nodes to be Tivoli System Automation for Multiplatforms cluster-ready.
2. Configure the required parameters in the `itm6/BuildScripts/clustervariables.sh` file for your selected type of cluster. The file contains explanations for all needed parameters.

3. Run the `itm6/BuildScripts/generateclusterfiles.sh` script to generate the definition files.
4. After ensuring that all resources are down or offline, run the `itm6/BuildScripts/mkitmcluster.sh` script to build the cluster.

"Installing and setting up the monitoring server on clusternode1" on page 80 describes the detailed steps necessary to build a cluster on a system without the DB2 resource. This option is used if the monitoring server component is being installed on this system, and the components requiring DB2 (portal server and Tivoli Data Warehouse) are not on this system.

"Testing the monitoring server on clusternode1" on page 81 describes the detailed steps necessary to build a cluster on a system with the DB2 resource. This option is used if a component requiring DB2 (portal server and Tivoli Data Warehouse) is being installed on this system. Use this section if you are creating a system with all components (monitoring server, portal server, and Tivoli Data Warehouse).

Note: The `mkitmcluster.sh` script calls other scripts that create the cluster domain, resource group, tiebreaker, shared file system and virtual IP resources, and optionally create the DB2 resource.

In certain cases, you might already have these resources defined, or you might want to define them separately. See Appendix C, "Predefined scripts," on page 145 for more information.

Building a Base Tivoli System Automation for Multiplatforms Cluster for the Monitoring Server component only

About this task

This procedure is used to create the base cluster without the DB2 resource. This option is used if the components requiring DB2 (portal server and Tivoli Data Warehouse) are not on this system.

Procedure

1. Follow the steps in Appendix A, "Configuring the cluster creation," on page 137.
2. Change to the `itm6/BuildScripts` directory, and run the build script with the type of the tiebreaker (scsi, net, disk, or eckd) as its parameter, such as:

   ```
   ./mkitmcluster.sh scsi
   ```

What to do next

When the cluster is up and running, you can verify that the cluster domain and the resources are online by running the commands:

`lsrpdomain`
: Verify that the domain ITM is now online

`lsrg -m`
: Verify that the resources are online

`lssam` A more detailed view

`rgreq –o move RG_NAME`
: Move the resource group RG_NAME to the other node

You are now ready to install the monitoring server in the Tivoli System Automation for Multiplatforms cluster.

Building a Tivoli System Automation for Multiplatforms Cluster for the portal server and Tivoli Data Warehouse components

For the portal server and Tivoli Data Warehouse components (Warehouse Proxy Agent and Summarization and Pruning Agent) you need one more resource, the underlying DB2.

The first step is to install the DB2 locally on each node. The database is not created yet. It is created when the cluster is up and running.

Note: The DB2 Tivoli System Automation for Multiplatforms policies that are used in this solution is a small subset of the DB2 Tivoli System Automation for Multiplatforms policies that are included with DB2.

Installing/Configuring DB2:
About this task

Use the DB2 Enterprise Server Edition.

Note: If you are running Linux on zSeries, you must install the 31-bit version of DB2 (regardless of whether you are running in 31-bit or 64-bit mode).

Procedure

1. Install DB2 locally on each cluster node into the same directory.
   This step ensures that the start, stop, and control scripts for the cluster work without modifications.
2. Make sure that you create the DB2 user IDs with the *same* GUUID and UUID on each node.
3. During the installation, create the same instance on both nodes.
4. Switch off the automatic startup of DB2 after the installation on each node:
   Switch to the instance owner user: `su – <instance_owner>`, and run:
   `db2iauto –off <instance>`
5. Because Warehouse Proxy Agent and Summarization and Pruning Agent use JDBC on UNIX and Linux, the TCP/IP listener for DB2 has to be configured correctly on both nodes.
   To do so, run the following commands for the appropriate instance:
   ```
   db2set -i instance_name DB2COMM=tcpip
   db2 update dbm cfg using SVCENAME port_number
   db2stop
   db2start
   ```
6. Change back to root.
7. Edit the `/etc/inittab` file on both nodes and comment out the following line:
   `fmc:2345:respawn:/opt/IBM/db2/V9.1/bin/db2fmcd #DB2 Fault Monitor Coordinator`
8. Stop DB2 on each node. To do so, switch to the user ID of the instance owner and run the command: `db2stop`

Building the portal server and Tivoli Data Warehouse Cluster:
Procedure

1. Follow the steps in Appendix A, "Configuring the cluster creation," on page 137.
2. Change to the `itm6/BuildScripts/` directory. Run the build script with the type of the tiebreaker (scsi, net, disk, or eckd), and db2 as the parameters, directing the script to create the DB2 resource:
   `./mkitmcluster.sh net db2`
   The script builds the complete Base cluster including the DB2 resource and starts it up.
   When the cluster is up and running, you can verify that the cluster domain and the resources are online by running the commands:

   `lsrpdomain`
   Verify that the domain ITM is now online

   `lsrg -m`
   Verify that the resources are online

   `lssam` A more detailed view

rgreq –o move RG_NAME
Move the resource group RG_NAME to the other node

Creating the Database for Portal server/Tivoli Data Warehouse on node1:
About this task

The next step is to create the database, either for the portal server or for the Tivoli Data Warehouse:

Use `lsrg -m` to find the active cluster node. On the active node, perform the following steps:

Procedure

1. Create a directory for the database on the shared file system, such as:
   `/shareddisk/DBdirectory`
2. Grant full access or ownership of this directory to the DB2 instance owner, typically to the user **db2inst1**, by running:
   `chown db2inst1 /shareddisk/DBdirectory`
3. Switch to the user ID of the instance owner.
4. Create a database using the DB2 CLI on the active node, using one of these commands:
   For the portal server: `db2 create database <DBName> on /shareddisk/DBdirectory`
   For Tivoli Data Warehouse: `db2 create database <DBName> on /shareddisk/DBdirectory using codeset UTF-8 territory US`
5. Test the connection to the database using the command:
   `db2 connect to <DBName>`
6. After successful connection, reset the connection using the command:
   `db2 connect reset`
7. Create a database user ID on both nodes with the same GUUID and UUID.
8. Grant full database access to this user ID on node1.
   Using the same IDs on both nodes allows the database operations to work correctly.

Cataloging the Portal server/Tivoli Data Warehouse Database on node 2:
Procedure

1. Switch the RG_NAME resource group to the other node:
   `rgreq –o move RG_NAME`

   Note: The terminal session might end after the move completes.
2. Wait until the resource group is online on node2, using:
   `lssam -top`
3. Grant full database access to the database user on node2.
4. On the active node, switch to the user ID of the instance owner.
5. Catalog the database in this instance using the command:
   `db2 catalog database <DBName> on /shareddisk/DBdirectory`
6. Test the connection to the database using the command:
   `db2 connect to <DBName>`
7. After successful connection, reset the connection using the command:
   `db2 connect reset`
8. Change back to the root user.
9. If you want to install a portal server in the cluster, and the Tivoli Data Warehouse is on another computer, also catalog the Tivoli Data Warehouse database and database server for the local DB2 as a

remote database server on both nodes, so that the portal server can access this database later. For further information, refer to the *IBM Tivoli Monitoring: Installation and Setup Guide*.

Results

You are now ready to install the portal server or Tivoli Data Warehouse components in this cluster.

Installing the monitoring server on its base Tivoli System Automation for Multiplatforms cluster

This section provides information on the installation of the monitoring server on its base Tivoli System Automation for Multiplatforms cluster.

Installing and setting up the monitoring server on clusternode1

About this task

Use the value that was entered for the **CANDLEHOME** variable in `clustervariables.sh` for the installation directory.

Use the value that was entered for the **HUBNAME** variable in `clustervariables.sh` for the Hub name.

Procedure

1. Check which cluster node is active (online):

   ```
   lsrg -m
   ```

2. Change the cluster's mode to Manual:

   ```
   samctrl –M T
   ```

3. On the **active node**, install the monitoring server into your chosen directory on the shared file system, according to the *IBM Tivoli Monitoring: Installation and Setup Guide*.
4. Use one of the following two methods to configure the monitoring server:

   To invoke a GUI application, use `<CANDLEHOME>/bin/itmcmd manage`, which requires a connection to an X Server.

   To invoke a text application, use `<CANDLEHOME>/bin/itmcmd config –S –t hub_name`, which prompts you for the configuration values.
5. Configure the monitoring server host name so that it is the same as the virtual host name of the monitoring server.

   The monitoring server host name is the **Host Name** field on the GUI **Basic Settings** tab, and the **TEMS Host Name** field in the text application.
6. Set the Primary Network Name to the virtual host name of the monitoring server. That ensures that the agents and portal server can connect to the monitoring server on this IP address.

   The Primary Network Name is the **Network Name** field in the **Optional Primary Network Name** box on the GUI **Advanced Settings** tab, and the **Optional Primary Network Name** field in the text application.
7. Start up the monitoring server and add the application support for the desired agent types.
8. Change the cluster's mode to Automatic:

   ```
   samctrl –M F
   ```

9. Disable the automatic startup of the monitoring server:
   a. Edit the `/shareddisk/ITM/config/kcirunas.cfg` file.
   b. Add the following section after the <agent> line by including the following syntax:

```
<productCode>ms</productCode>
<default>
  <autoStart>no</autoStart>
</default>
```

c. Save the file.

d. Run the `/shareddisk/ITM/bin/UpdateAutoRun.sh` command as root user.

Note: The path to the `kcirunas.cfg` file and the `UpdateAustoRun.sh` command (`/shareddisk/ITM`) might be different on your system.

Testing the monitoring server on clusternode1

Verify that you can start the monitoring server and other components that you installed.

Setting up the monitoring server on clusternode2

Procedure

1. Shut down the monitoring server and change to the config directory, such as:
   <CANDLEHOME>/config
2. Copy the monitoring server configuration file into the config directory for node2.
   To do so, copy the file:
   node1_ms_HUB_NAME.config
   to
   node2_ms_HUB_NAME.config
3. Verify that the monitoring server is stopped.

Adding the monitoring server to the resource group of the Base Cluster

About this task

When the installation of the monitoring server is complete, you have to integrate it into your Base cluster, so that it can then run under the cluster's control.

Procedure

1. On one of the nodes, take the resource group RG_NAME offline:
   ```
   chrg –o offline RG_NAME
   ```
2. Verify that the resource group is offline, it can take 1 or 2 minutes:
   ```
   lssam -top
   ```
3. Change to the **itm6/BuildScripts** directory.
4. Run the `temsrscbuild.sh` command as root from the same directory.
5. Start the resource group again using the following command:
   ```
   chrg –o online RG_NAME
   ```
6. Verify that the resource group is online:
   ```
   lssam -top
   ```

Results

The monitoring server is now running in the Tivoli System Automation for Multiplatforms cluster.

Testing the monitoring server failover to clusternode2
About this task

Run the `rgreq –o move RG_NAME` command; then, invoke `lsrg –m` to verify that monitoring server starts on the other node. If it doesn't start:

Procedure

1. Ensure the values entered in the ITM/BuildScripts/clustervariables.sh file are correct.
2. Remove the Tivoli System Automation for Multiplatforms domain using the following commands:

   ```
   stoprpdomain &endash;f domain_name
   rmrpdomain domain_name
   ```
3. Re-execute the steps in "IBM Tivoli Monitoring cluster setup" on page 41.

Installing the portal server on the Tivoli System Automation for Multiplatforms cluster

This section includes information on the installation and setup of the portal server on the Tivoli System Automation for Multiplatforms cluster.

Installing and setting up the portal server on clusternode1
About this task

To install and set up the portal server on clusternode1, complete the following procedure:

Procedure

1. Check which cluster node is active (online):

   ```
   lsrg –m
   ```
2. If you installed other Tivoli Monitoring components to the shared file system, use the same installation path for the portal server.
3. Change the cluster's mode to Manual:

   ```
   samctrl –M T
   ```
4. On the active node, install the portal server, according to *IBM Tivoli Monitoring: Installation and Setup Guide*.

   Note: If you are installing the portal server on Linux on zSeries in 64- bit mode, you need to run the install script from a 31-bit session. To open a 31-bit session, run the following command:

   ```
   s390 sh
   ```
5. Configure the portal server interface and URL such that the portal clients can connect to portal server on its virtual host name. Add the following line to the `<CANDLEHOME>/config/cq.ini` file:

   ```
   KFW_INTERFACE_cnps_HOST=<virtualhostname>
   ```

   where `<virtualhostname>` is the virtual host name of the portal server.
6. Use one of the following 2 methods to configure the portal server:

   To invoke a GUI application, use `<CANDLEHOME>/bin/itmcmd manage`, which requires a connection to an X Server.

   To invoke a text application, use `<CANDLEHOME>/bin/itmcmd config –A cq`, which prompts you for the configuration values.

   Note: If you are configuring the portal server on Linux on zSeries in 64-bit mode, you need to run the command from a 31-bit session. To open a 31-bit session, run the following command:

s390 sh

7. Configure the monitoring server host name so that it is the same as the virtual host name of the monitoring server (assuming the hub monitoring server is clustered).

   The monitoring server host name is the **TEMS Hostname** field on the GUI **TEMS Connection** tab, and the **CMS Host Name** field in the text application.

8. Configure the portal server to use the virtual host name.
   a. Make sure Tivoli Enterprise Portal Server and Tivoli Enterprise Portal Server Extensions are stopped.
   b. Back up your configuration data located in the `$CANDLEHOME/<architecture>/iw/profiles` directory.
   c. Run UNIX terminal.
   d. Change the current directory to `$CANDLEHOME/<architecture>/iw/scripts`
   e. Run `sh updateTEPSEHostname.sh <old_hostname> <new_hostname>`.

      Under <old_hostname> substitute the current host name, do not include domain name.

      Under <new_hostname> substitute the valid virtual host name for your Tivoli Enterprise Portal Server cluster.

      Note: If you repeat the `updateTEPSEHostname` script execution, make sure to use correct `old_hostname`. For example, the following sequence it correct:

      `updateTEPSEHostname.bat h1 crocodile`, the ewas host name was changed to "crocodile".

      `updateTEPSEHostname.bat crocodile hippi`, as old host name, you must use "crocodile" or the script will have no effect.

   f. If the operation is successful, 'BUILD SUCCESSFUL' is displayed.
   g. Set the Primary Network Name to the virtual host name of the portal server. This configuration is necessary for the communication between the portal server and the rest of the Tivoli Monitoring infrastructure, including the monitoring server.

      The Primary Network Name in the **Network Name** field, which is available after selecting **Specify Optional Primary Network** on the GUI **TEMS Connection** tab, and the **Optional Primary Network Name** field in the text application.

9. Configure the portal server to have access to the database which you created before (see "Creating the Database for Portal server/Tivoli Data Warehouse on node1" on page 79).
10. Change the cluster's mode to Automatic:

    ```
    samctrl –M F
    ```

11. Disable the automatic startup of the portal server:
    a. Edit the `/shareddisk/ITM/config/kcirunas.cfg` file.
    b. Add the following section after the <agent> line by including the following syntax:

    ```
    <productCode>cq</productCode>
    <default>
      <autoStart>no</autoStart>
    </default>
    ```

    c. Save the file.
    d. Run the `/shareddisk/ITM/bin/UpdateAutoRun.sh` command as root user.

    Note: The path to the `kcirunas.cfg` file and the `UpdateAutoRun.sh` command (`/shareddisk/ITM`) might be different on your system.

Testing the portal server on clusternode1

Verify that you can bring up the Tivoli Enterprise Portal, and that you can perform standard operations.

Note: If you are starting the portal server on Linux on zSeries in 64-bit mode, you need to run the command from a 31-bit session. To open a 31- bit session, run the following command:

```
s390 sh
```

Adding the portal server to the resource group of the Base Cluster

About this task

When the installation of the portal server is complete, you have to integrate it into your cluster, so that it can then run under the cluster's control.

The portal server is now running in the Tivoli System Automation for Multiplatforms cluster.

Procedure

1. On one of the nodes, take the resource group RG_NAME offline:
   ```
   chrg –o offline RG_NAME
   ```
2. Verify that the resource group is offline, it can take 1 or 2 minutes:
   ```
   lssam -top
   ```
3. Change to the `itm6/BuildScripts` directory.
4. Run the `tepsrscbuild.sh` command as root from the same directory.
5. Start the resource group again using the following command:
   ```
   chrg –o online RG_NAME
   ```
6. Verify that the resource group is online:
   ```
   lssam -top
   ```

Testing the portal server failover to clusternode2

About this task

Run the `rgreq –o move RG_NAME` command, and use `lssam -top` to verify that the portal server starts on the other node. If it doesn't start:

Procedure

1. Fix the values entered in the `itm6/BuildScripts/clustervariables.sh` file. (See "Predefined Tivoli System Automation for Multiplatforms Cluster for Tivoli Monitoring" on page 76)
2. Remove the Tivoli System Automation for Multiplatforms domain by using the following commands:
   ```
   stoprpdomain –f domain_name
   rmrpdomain domain_name
   ```
3. Re-execute the steps in "IBM Tivoli Monitoring cluster setup" on page 41.

   Note: When you configure the portal server, the portal server is stopped. If Tivoli System Automation for Multiplatforms is in normal mode, it will start the portal server causing the configuration to fail. Put Tivoli System Automation for Multiplatforms in manual mode (**`samctrl -M T`**) before configuring the portal server.

Installing the Warehouse Proxy Agent on a Tivoli System Automation for Multiplatforms cluster

This section includes information on the installation of the warehouse proxy agent on a Tivoli System Automation for Multiplatform cluster.

Installing and setting up the Warehouse Proxy Agent on clusternode1

Procedure

1. Check which cluster node is active (online):

   ```
   lsrg -m
   ```

2. If you installed other Tivoli Monitoring components to the shared file system, use the same installation path for the Warehouse Proxy Agent.
3. Change the cluster's mode to Manual:

   ```
   samctrl –M T
   ```

4. On the **active** node, install the warehouse proxy agent, according to the *IBM Tivoli Monitoring: Installation and Setup Guide*.
5. Use the following method to configure the warehouse proxy agent:

   To invoke a GUI application, use `<CANDLEHOME>/bin/itmcmd manage` (which requires a connection to an X Server).
6. Configure the monitoring server host name so that it is the same as the virtual host name of the monitoring server (assuming the hub monitoring server is clustered).

   The monitoring server host name is the **TEMS Hostname** field on the GUI **TEMS Connection** tab.
7. Set the Primary Network Name to the virtual host name of the warehouse proxy agent system.

   This is the **Network Name** field, which is available after selecting **Specify Optional Primary Network** on the GUI **TEMS Connection** tab.
8. Configure the Warehouse Proxy Agent to allow access to the database which you created previously. On the **Agent Parameters** tab:
   a. Add the DB2 JDBC drivers. (To add JDBC drivers, use the scroll bar in the **Sources** section, and scroll to the right, exposing the **Add** and **Delete** buttons.)
   b. Update the **Warehouse URL** so that it refers to the correct database and port.
   c. Set the **Warehouse User** and **Warehouse Password**.
   d. Check the parameters by using the **Test database connection** button.
   e. Save the configuration.
9. Stop the agent if it is running.
10. Change the cluster's mode to Automatic:

    ```
    samctrl –M F
    ```

11. Disable the automatic startup of the Warehouse Proxy Agent:
    a. Edit the `/shareddisk/ITM/config/kcirunas.cfg` file.
    b. Add the following section after the <agent> line by including the following syntax:

       ```
       <productCode>hd</productCode>
       <default>
         <autoStart>no</autoStart>
       </default>
       ```

    c. Save the file.
    d. Run the `/shareddisk/ITM/bin/UpdateAutoRun.sh` command as root user.

    Note: The path to the `kcirunas.cfg file` and the `UpdateAutoRun.sh` command (`/shareddisk/ITM`) might be different on your system.

Adding the Warehouse Proxy Agent to the resource group of the Base Cluster

About this task

When the installation of the Warehouse Proxy Agent is complete, integrate it into the cluster, so that it can run under the cluster's control.

Procedure

1. On one of the nodes, take the resource group RG_NAME offline:
   ```
   chrg –o offline RG_NAME
   ```
2. Verify that the resource group is offline, it can take 1 or 2 minutes:
   ```
   lssam -top
   ```
3. Change to the `itm6/BuildScripts` directory.
4. Run the `tdwproxyrscbuild.sh` command as root from the same directory.
5. Start the resource group again using the following command:
   ```
   chrg –o online RG_NAME
   ```
6. Verify that the resource group is online:
   ```
   lssam -top
   ```

Results

The Warehouse Proxy Agent is now running in the Tivoli System Automation for Multiplatforms cluster on top of the clustered DB2 for the Tivoli Data Warehouse.

Testing the Warehouse Proxy Agent failover to clusternode2

About this task

Run the `rgreq –o move RG_NAME` command, and use `lssam -top` to verify that the warehouse proxy agent starts on the other node. If it doesn't start:

Procedure

1. Fix the values entered in the `itm6/BuildScripts/clustervariables.sh` file.
2. Remove the Tivoli System Automation for Multiplatforms domain using the following commands:
   ```
   stoprpdomain –f domain_name
   rmrpdomain domain_name
   ```
3. Re-execute the steps in "IBM Tivoli Monitoring cluster setup" on page 41.

Note: The terminal session might end after the move completes.

Installing the Summarization and Pruning Agent on a Tivoli System Automation for Multiplatforms cluster

This section provides information on the installation of the Summarization and Pruning Agent on a Tivoli System Automation for Multiplatforms cluster.

Installing and setting up the Summarization and Pruning Agent on clusternode1

Procedure

1. Check which cluster node is active (online):
   ```
   lsrg -m
   ```

2. If you installed other Tivoli Monitoring components to the shared file system, use the same installation path for the Summarization and Pruning Agent.
3. Change the cluster's mode to Manual:

   ```
   samctrl –M T
   ```
4. On the **active** node, install the warehouse proxy agent, according to the *IBM Tivoli Monitoring: Installation and Setup Guide*.
5. Use the following method to configure the Summarization and Pruning Agent:

   To invoke a GUI application, use `<CANDLEHOME>/bin/itmcmd manage`, which it requires a connection to an X Server.
6. Configure the monitoring server host name so that it is the same as the virtual host name of the monitoring server (assuming the hub monitoring server is clustered).

   The monitoring server host name is the **TEMS Hostname** field on the GUI **TEMS Connection** tab.
7. Set the Primary Network Name to the virtual host name of the Summarization and Pruning Agent system.

   The Primary Network Name is the **Network Name** field, which is available after selecting **Specify Optional Primary Network** on the GUI **TEMS Connection** tab.
8. Configure the Warehouse Proxy Agent to allow access to the database which you created previously. On the **Agent Parameters** tab:
   a. Add the DB2 JDBC drivers. (To add JDBC drivers, use the scroll bar in the **Sources** section, and scroll to the right, exposing the **Add** and **Delete** buttons.)
   b. Update the **Warehouse URL** so that it refers to the correct database and port.
   c. Set the **Warehouse User** and **Warehouse Password**.
   d. Check the parameters by using the **Test database connection** button.
   e. Save the configuration.
9. Shut down the agent if it is running.
10. Change the cluster's mode to Automatic:

    ```
    samctrl –M F
    ```
11. Disable the automatic startup of the Summarization and Pruning Agent:
    a. Edit the `/shareddisk/ITM/config/kcirunas.cfg` file.
    b. Add the following section after the <agent> line by including the following syntax:

       ```
       <productCode>sy</productCode>
       <default>
         <autoStart>no</autoStart>
       </default>
       ```
    c. Save the file.
    d. Run the `/shareddisk/ITM/bin/UpdateAutoRun.sh` command as root user.

    Note: The path to the `kcirunas.cfg` file and the `UpdateAutoRun.sh` command (`/shareddisk/ITM`) might be different on your system.

Adding the Summarization and Pruning Agent to the resource group of the Base Cluster

About this task

When the installation of the Summarization and Pruning Agent is complete, integrate it into the cluster, so that it can run under the cluster's control.

Procedure

1. On one of the nodes, take the resource group RG_NAME offline:
   ```
   chrg –o offline RG_NAME
   ```
2. Verify that the resource group is offline, it can take 1 or 2 minutes:
   ```
   lssam -top
   ```
3. Change to the `itm6/BuildScripts` directory.
4. Run the `spagentrscbuild.sh` command as root from the same directory.
5. Start the resource group again using the following command:
   ```
   chrg –o online RG_NAME
   ```
6. Verify that the resource group is online:
   ```
   lssam -top
   ```

Results

The Summarization and Pruning Agent is now running in the Tivoli System Automation for Multiplatforms cluster on top of the clustered DB2 for the Tivoli Data Warehouse.

Testing the Summarization and Pruning Agent failover to clusternode2

About this task

Run the `rgreq –o` move RG_NAME command, and use `lssam -top` to verify that Summarization and Pruning Agent starts on the other node. If it doesn't start:

Procedure

1. Fix the values entered in the `itm6/BuildScripts/clustervariables.sh` file.
2. Remove the Tivoli System Automation for Multiplatforms domain using the following commands:
   ```
   stoprpdomain –f domain_name
   rmrpdomain domain_name
   ```
3. Re-execute the steps in "IBM Tivoli Monitoring cluster setup" on page 41.

Note: The terminal session might end after the move completes.

Performing IBM Tivoli Monitoring Maintenance on a Tivoli System Automation for Multiplatforms cluster

Performing maintenance on Tivoli Monitoring while it is running in a Tivoli System Automation for Multiplatforms cluster requires you to gracefully stop the component, without unmounting the shared disk. This is done using the Tivoli System Automation for Multiplatforms **rgmbrreq** command.

Applying a fix pack to the Hub Tivoli Enterprise Monitoring Server

Procedure

1. On one of the nodes, run the following command:
   ```
   rgmbrreq –o stop IBM.Application:TEMSSRV
   ```
2. Check which cluster node is active (online), and verify that the **IBM.Application:TEMSSRV** member is offline:
   ```
   lsrg –m
   ```
3. On the active node, verify that the shared disk is available.
4. Follow the fix pack instructions to install the monitoring server component upgrade to the shared disk.

5. Disable the automatic startup of the monitoring server:
   a. Edit the /shareddisk/ITM/config/kcirunas.cfg file.
   b. Add the following section after the <agent> line by including the following syntax:
      ```
      <productCode>ms</productCode>
      <default>
        <autoStart>no</autoStart>
      </default>
      ```
   c. Save the file.
   d. Run the /shareddisk/ITM/bin/UpdateAutoRun.sh command as root user.

   Note: The path to the kcirunas.cfg file and the UpdateAutoRun.sh command (/shareddisk/ITM) might be different on your system.
6. Run the following command:
   ```
   rgmbrreq –o cancel IBM.Application:TEMSSRV
   ```
7. Verify that the resource group is online:
   ```
   lsrg -m
   ```
8. Refer to "Testing the monitoring server failover to clusternode2" on page 82 to test failover.

Applying a fix pack to the Tivoli Enterprise Portal Server

Procedure

1. On one of the nodes, run the following command:
   ```
   rgmbrreq –o stop IBM.Application:TEPSSRV
   ```
2. Check which cluster node is active (online), and verify that the **IBM.Application:TEPSSRV** member is offline:
   ```
   lsrg –m
   ```
3. On the active node, verify that the shared disk is available.
4. Follow the fix pack instructions to install the portal server component upgrade to the shared disk.
5. Disable the automatic startup of the portal server:
   a. Edit the /shareddisk/ITM/config/kcirunas.cfg file.
   b. Add the following section after the <agent> line by including the following syntax:
      ```
      <productCode>ms</productCode>
      <default>
        <autoStart>no</autoStart>
      </default>
      ```
   c. Save the file.
   d. Run the /shareddisk/ITM/bin/UpdateAutoRun.sh command as root user.

   Note: The path to the kcirunas.cfg file and the UpdateAutoRun.sh command (/shareddisk/ITM) might be different on your system.
6. Run the following command:
   ```
   rgmbrreq –o cancel IBM.Application:TEPSSRV
   ```
7. Verify that the resource group is online:
   ```
   lsrg -m
   ```
8. Refer to "Testing the portal server failover to clusternode2" on page 84 to test failover.

Applying a fix pack to the Warehouse Proxy Agent

Procedure

1. On one of the nodes, run the following command:

   ```
   rgmbrreq –o stop IBM.Application:TDWProxy
   ```

2. Check which cluster node is active (online), and verify that the **IBM.Application:TDWProxy** member is offline:

   ```
   lsrg –m
   ```

3. On the active node, verify that the shared disk is available.
4. Follow the fix pack instructions to install the Warehouse Proxy agent component upgrade to the shared disk.
5. Disable the automatic startup of the Warehouse Proxy agent:
   a. Edit the `/shareddisk/ITM/config/kcirunas.cfg` file.
   b. Add the following section after the <agent> line by including the following syntax:

      ```
      <productCode>hd</productCode>
      <default>
        <autoStart>no</autoStart>
      </default>
      ```

   c. Save the file.
   d. Run the `/shareddisk/ITM/bin/UpdateAutoRun.sh` command as root user.

   Note: The path to the `kcirunas.cfg` file and the `UpdateAutoRun.sh` command (`/shareddisk/ITM`) might be different on your system.
6. Run the following command:

   ```
   rgmbrreq –o cancel IBM.Application:TDWProxy
   ```

7. Verify that the resource group is online:

   ```
   lsrg -m
   ```

8. Refer to "Testing the Warehouse Proxy Agent failover to clusternode2" on page 86 to test failover.

Applying a fix pack to the Summarization and Pruning Agent

Procedure

1. On one of the nodes, run the following command:

   ```
   rgmbrreq –o stop IBM.Application:SPAgent
   ```

2. Check which cluster node is active (online), and verify that the **IBM.Application:SPAgent** member is offline:

   ```
   lsrg –m
   ```

3. On the active node, verify that the shared disk is available.
4. Follow the fix pack instructions to install the Summarization and Pruning agent component upgrade to the shared disk.
5. Disable the automatic startup of the Summarization and Pruning agent:
   a. Edit the `/shareddisk/ITM/config/kcirunas.cfg` file.
   b. Add the following section after the <agent> line by including the following syntax:

      ```
      <productCode>ms</productCode>
      <default>
        <autoStart>no</autoStart>
      </default>
      ```

   c. Save the file.
   d. Run the `/shareddisk/ITM/bin/UpdateAutoRun.sh` command as root user.

Note: The path to the `kcirunas.cfg` file and the `UpdateAutoRun.sh` command (`/shareddisk/ITM`) might be different on your system.

6. Run the following command:

   ```
   rgmbrreq –o cancel IBM.Application:SPAgent
   ```

7. Verify that the resource group is online:

   ```
   lsrg -m
   ```

8. Refer to "Testing the Summarization and Pruning Agent failover to clusternode2" on page 88 to test failover.

Known problems and limitations

It is important to remember specific characteristics and constraints of Tivoli Monitoring installation and set up, and their effects on the cluster setup.

During the certification test for Tivoli Monitoring clustering, issues encountered when setting up the clustered environment are formally reported as defects. These defects are typically related to the setup of Tivoli Monitoring in a non-default manner, instead of being specific to the cluster environment. These defects are handled as part of the Tivoli Monitoring service stream. Here is a list of known problems and workarounds.

- The Tivoli Monitoring installer configures the components to be auto started by default. It does not give the user an option to configure the components to not auto start. Under this limitation, the user has to edit an operating system script to remove this behavior.

  Under Linux, another limitation is that the Tivoli Monitoring installer places the auto start commands under every OS run level. Because all of these files are actually links, editing one of them removes this behavior at all run levels.

  This same behavior occurs whether installing for the first time, or applying Fix Packs.

  See the installation sections and the maintenance section for details on how to configure the components not to auto start.

- Chapter 5, "Installing and uninstalling service," in the *Tivoli System Automation for Multiplatforms Installation and Configuration Guide* is missing 2 steps required to install service on a node:

  After running the `samctrl -u a Node` command in step 2, you need to run the following command to stop the node:

  ```
  stoprpnode Node
  ```

  Before running the `samctrl -u d Node` command in step 6, you need to run the following command to start the node:

  ```
  startrpnode Node
  ```

- In some cases, removing a cluster domain from one node does not remove it from the second node. This is evident when you attempt to create a domain with the same name that was just removed, and you get an error that the domain already exists.

  To remedy this, when removing a domain, run the `rmrpdomain` command on both nodes.

- If a resource stays in **Stuck online** status for a long amount of time, as shown in the results of `lsrg –m` or `lssam –top`, this could mean that the configured stop command timeout for the resource is too low.

  You should time how long it takes to stop the resource, and double that, to get a good timeout value. When you have this value, do the following to change the value for the resource:

  1. Run the following commands to take the resource group offline (this causes Tivoli Monitoring to stop):

     ```
     export CT_MANAGEMENT_SCOPE=2
     chrg –o offline RG_NAME
     ```

  2. Stop any remaining processes that have not ended for the resource.

3. Run the appropriate command for the resource that you would like to change, where ***n*** is the new timeout value, in seconds:

Table 7. Change Resource (chrsrc) commands for setting the Timeout value

Resource	Command
Hub monitoring server	`chrsrc –s "Name = 'TEMSSRV'" IBM.Application StopCommandTimeout=`*n*
Portal server	`chrsrc –s "Name = 'TEPSSRV'" IBM.Application StopCommandTimeout=`*n*
Warehouse Proxy Agent	`chrsrc –s "Name = 'TDWProxy'" IBM.Application StopCommandTimeout=`*n*
Summarization and Pruning Agent	`chrsrc –s "Name = 'TDWProxy'" IBM.Application StopCommandTimeout=`*n*

4. Run the following command to bring the resource group online:

   ```
   chrg –o online RG_NAME
   ```

- IBM Tivoli System Automation for Multiplatforms on AIX 5.3 related problems
  - During testing, a problem was encountered where shared disks were not being properly released or mounted after failures. This problem was solved after installing the latest storageRM file set, which can be downloaded from http://www-1.ibm.com/support/docview.wss?rs=1207&context=SG11P&dc=DB510&dc=DB550&q1=rsct.opt.storagerm.2.4.7.1&uid=isg1fileset-671500801&loc=en_US&cs=UTF-8&lang=all.
  - In the test bed, there was a line in the inittab of the OS that prevented RSCT processes from starting automatically after the computer reboot. If there is such a line in your environment, and it is located above the lines that related to RSCT processes, then make sure that you comment out this line:

    ```
    install_assist:2:wait:/usr/sbin/install_assist </dev/console>/dev/console 2>&1
    ```

- In order for Java Web Start to work, you must change all occurrences of $HOST$ to your fully qualified host name in the .jnlpt file. You can locate the .jnlpt file in the `CANDLE_HOME\config` directory.

Chapter 8. Creating clusters with Tivoli Monitoring components in a Microsoft Cluster Server environment

Highly available IBM Tivoli Monitoring environments are designed and implemented by using a Microsoft Cluster Server. This server has both a GUI Cluster Administrator and a CLI command (cluster.exe). Use the Cluster Administrator to install and configure Tivoli Monitoring in Microsoft Cluster Server.

This chapter includes guidance for implementing highly available strategies by using the Tivoli Monitoring components; practical considerations regarding design, implementation, testing, maintenance work with the system, and upgrade are also discussed.

- "Setting up the hub monitoring server in a Microsoft Cluster Server" covers the hub Tivoli Enterprise Monitoring Server (monitoring server) running on a Microsoft Cluster Server.
- "Setting up the portal server in a Microsoft Cluster Server" on page 108 includes information about running the Tivoli Enterprise Portal Server (portal server) on a Microsoft Cluster Server.
- "Setting up Tivoli Data Warehouse components in a Microsoft Cluster Server" on page 126 includes information about running the Tivoli Data Warehouse and related components (Summarization and Pruning Agent and Warehouse Proxy Agent) on a Microsoft Cluster Server.

Note: If any of these IBM Tivoli Monitoring components will be running on the same cluster, many of the base steps are common and must be completed only once.

For an overview of Tivoli Monitoring components on clustered environments, see Chapter 5, "The clustering of IBM Tivoli Monitoring components," on page 37 and to *IBM Tivoli Monitoring Resiliency and High Availability*.

Important: These configurations and operational procedures for high availability have been tested by IBM Software Support and represent supported product functions. Other configurations or procedures are possible but might be outside of product design parameters; therefore, other configurations or procedures might not operate correctly or be supported. In addition, these configurations and operational procedures cover only DB2 as the RDBMS used by the Tivoli Monitoring infrastructure. IBM Tivoli Monitoring V6.3 includes a restricted-use version of IBM DB2 10.1 Enterprise Server Edition for use with the Tivoli Enterprise Portal Server and the Tivoli Data Warehouse.

Setting up the hub monitoring server in a Microsoft Cluster Server

To set up a highly-available hub monitoring server in a Microsoft Cluster Server, you must complete the following procedures:

1. "Setting up basic cluster resources"
2. "Installing and setting up the monitoring server on clusternode1" on page 98
3. "Adding the monitoring server resource to your resource group" on page 103
4. "Setting up the monitoring server on clusternode2" on page 106

Setting up basic cluster resources

For hub monitoring server installation and setup on a Microsoft Cluster Server, the following four resources are necessary:

- Shared disk R: You can assign a different drive letter for the shared disk.
- IP address: Must be a valid domain name system registered IP, using static IP.
- Host name: Must be a valid domain name system registered host name with the IP above.

- Hub monitoring server: Contains all required information.

Note: These resources might have already been created by your cluster administrator. If they are not created in your environment, complete the following steps to create these resources.

In the test environment, Windows Server 2008 R2 Enterprise 64-bit was used. Each cluster was set up with a single quorum device and two nodes.

To set up of the basic cluster resources, start with clusternode1:

1. Start the Cluster Administrator and connect it to the cluster that you are working in. Make sure that the cluster is running on clusternode1.

   By default, the Cluster Administrator contains a Cluster Group (for internal cluster setup) and a Group 0. Use Group 0 as the resource group for IBM Tivoli Monitoring. The Cluster Group contains a Cluster Name (in this case, IVTCS001), which is used for follow on configurations.

   The disk R is by default held in "Group 0". You can now rename this group according to your needs. In the testing environment, it was renamed to HUBTems. You can also create a new group and then move this shared disk into that new group.

   Note: Do not put any Tivoli Monitoring related resources into the default group named Cluster Group, which contains Cluster IP, Name, and the quorum disk.

   The shared disk resource has been set, and you must set the IP address and the host name. You define the hub monitoring server service after the installation process.
2. To set up the resource for the IP address, complete these steps:
   a. In the left-hand pane of the Failover Cluster Manager, right-click **Services and applications**, click **More actions > Create Empty service or application**.
   b. Right-click **new service or application**.
   c. Select **Add a resource > More resources... > Add IP Address** and then specify the name you want, as shown in Figure 14:

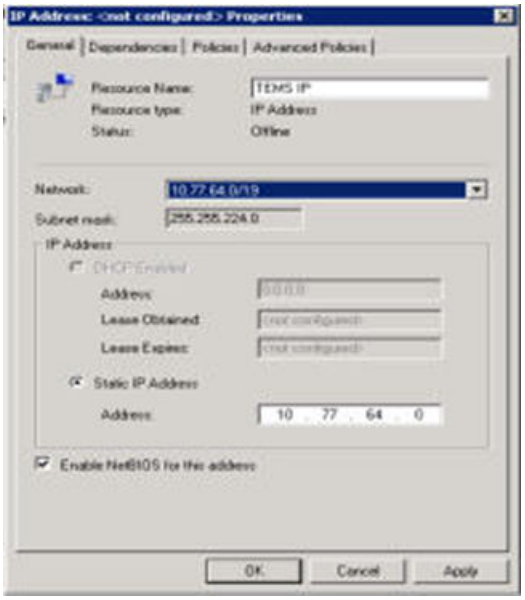

Figure 14. Specify the resource name and virtual IP address

   d. Specify the dependencies. Ensure that the shared disk is available before anything else, so you can specify the shared disk R as the dependency for the IP Address.

Right-click on the New Service or Application, and seleect **Add Storage**.

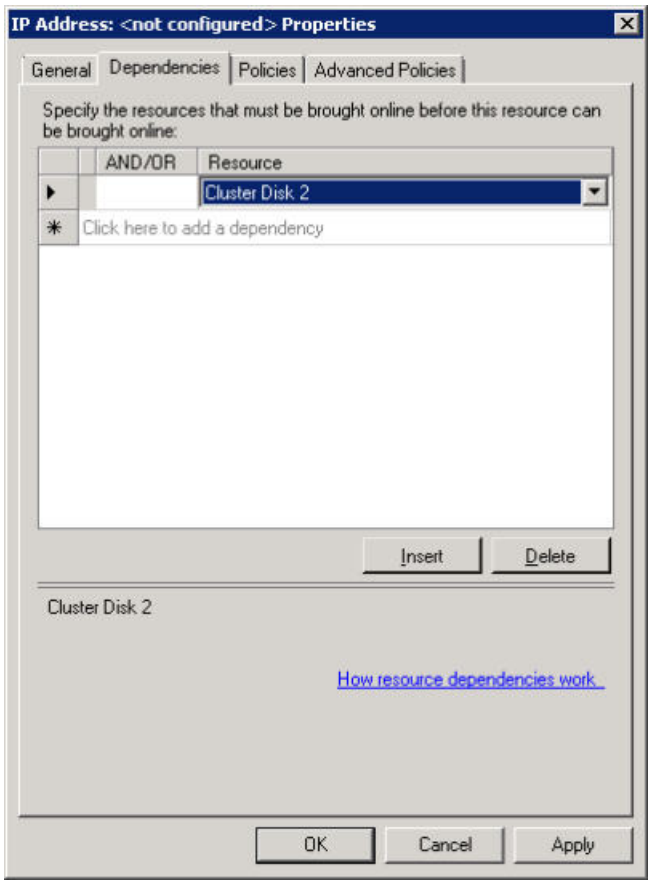

Figure 15. Specify the resources that must be brought online before this resource can be brought online

e. Select both of the node names in the **Possible Owners** window:

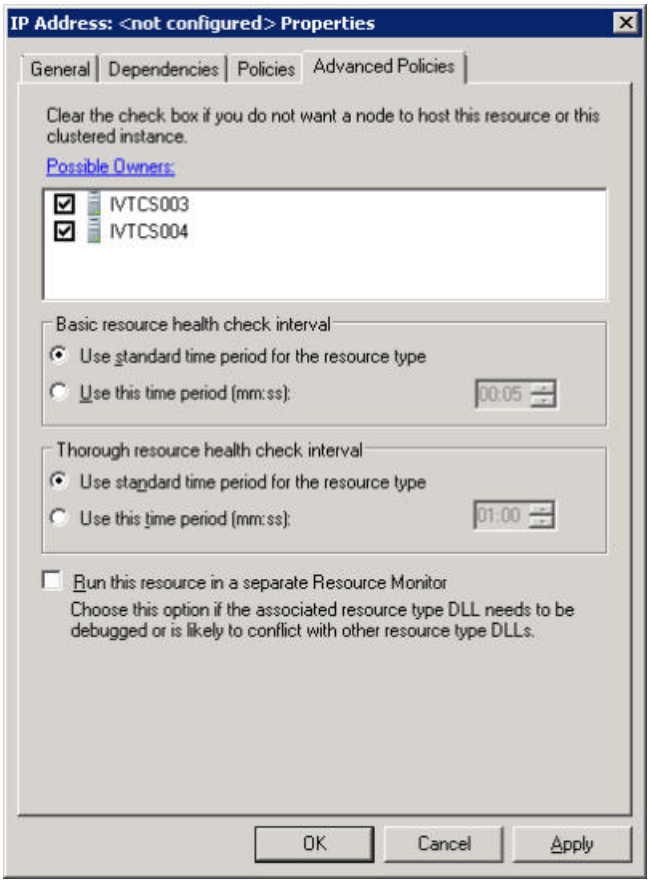

Figure 16. Specification of Possible Owners

Note: Node names are selected as possible owners by default.

f. Specify the virtual IP Address to be used for the hub TEMS cluster.

Note: The Cluster IP Address is the virtual IP Address. If a virtual IP Address already exists, this address is the cluster name.

g. Click **Finish**.

The following message is displayed: `Cluster resource "TEMS IP" created successfully`

3. Right-click this resource and then click **Bring Online**, as shown in Figure 17 on page 97:

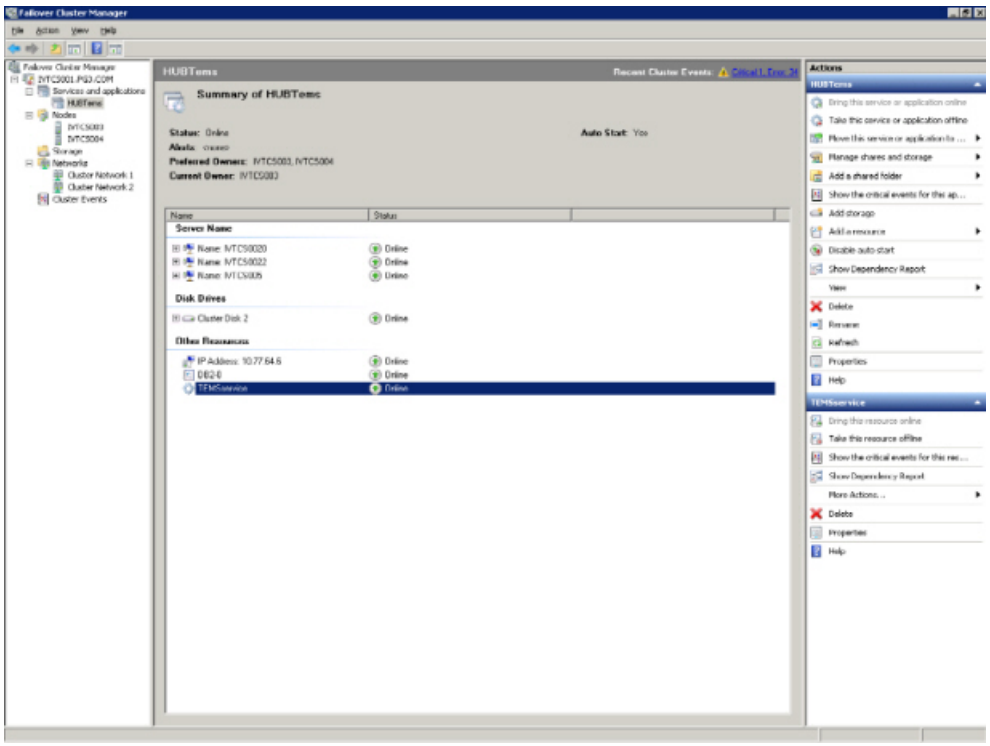

Figure 17. Cluster Administrator before Bring Online function

4. Create the resource for the network name, by following a similar procedure. The only difference is that you select the **Client Access Point** resource type when you add a resource in the Failover Cluster Manager. In the test environment, the Network Name is *IVTCS007*.

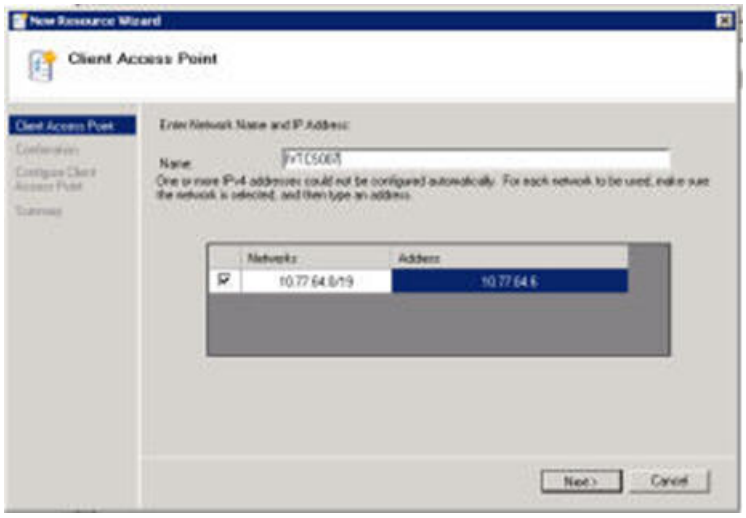

Figure 18. Create the resource for the network name

5. Follow the prompts to create a virtual host name for your virtual IP address. The dependencies are the disk R and the IP address. Also bring this host name online. Test the virtual host name by pinging it from another computer.

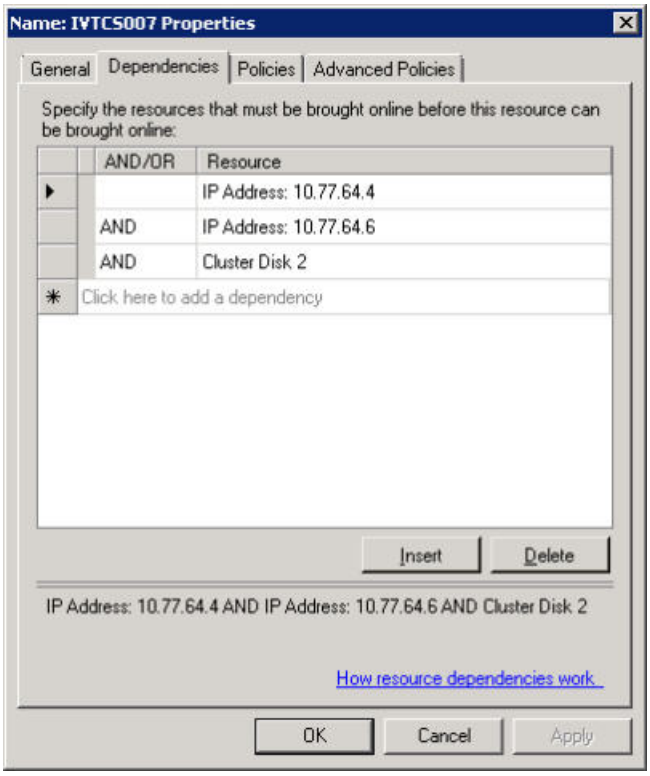

Figure 19. Specify the dependencies of the Client Access Point

6. Create the Tivoli Monitoring default user ID sysadmin as a local user on the first and on the second node of the cluster.

You are now ready to install the monitoring server.

Installing and setting up the monitoring server on clusternode1

About this task

You may install only one installation of IBM Tivoli Monitoring on a Windows computer. The installation on the shared disk removes any Tivoli Monitoring V6.x component that was previously installed locally on the node. Previously installed components are removed because the installation directory changed from the local disk to the shared disk. Therefore, you must make sure that no Tivoli Monitoring 6.x component is installed locally before you start the setup.

Before installing the monitoring server, complete the following prerequisite steps:

- If necessary, uninstall IBM Tivoli Monitoring, remove all related registry settings, Tivoli Enterprise Portal Server and Warehouse ODBC data sources, and all IBM Tivoli Monitoring users.
- Within the Failover Cluster Manager, move resources to the IVTCS003 node (Node 1) by right-clicking on HUBTems group and choosing Move Group.
- Before installing or configuring any components in a Microsoft Cluster environment, all other components that are already added to resource groups on this cluster should be offline. This precautionary measure prevents conflicts between the cluster management tool and the Tivoli Monitoring MTEMS tool. Within the Failover Cluster Manager, stop cluster service after right-clicking on IVTCS004 (Node 2).

Procedure

1. Start the installation by using the **setup.exe** command. After the Welcome to Tivoli Monitoring window, the InstallShield Wizard displays the Software License agreement.
2. Click **Accept** to accept the Software License agreement. The window to specify the destination folder is displayed.

Figure 20. Specifying a destination folder

3. Install Tivoli Monitoring on the shared disk (the default is `R:\IBM\ITM`, but you can specify a different folder).
4. Click **Next** after you specify a destination folder. The window to specify features you want to install is displayed. Select only **Tivoli Enterprise Monitoring Server - TEMS**. Additional features are selected automatically.

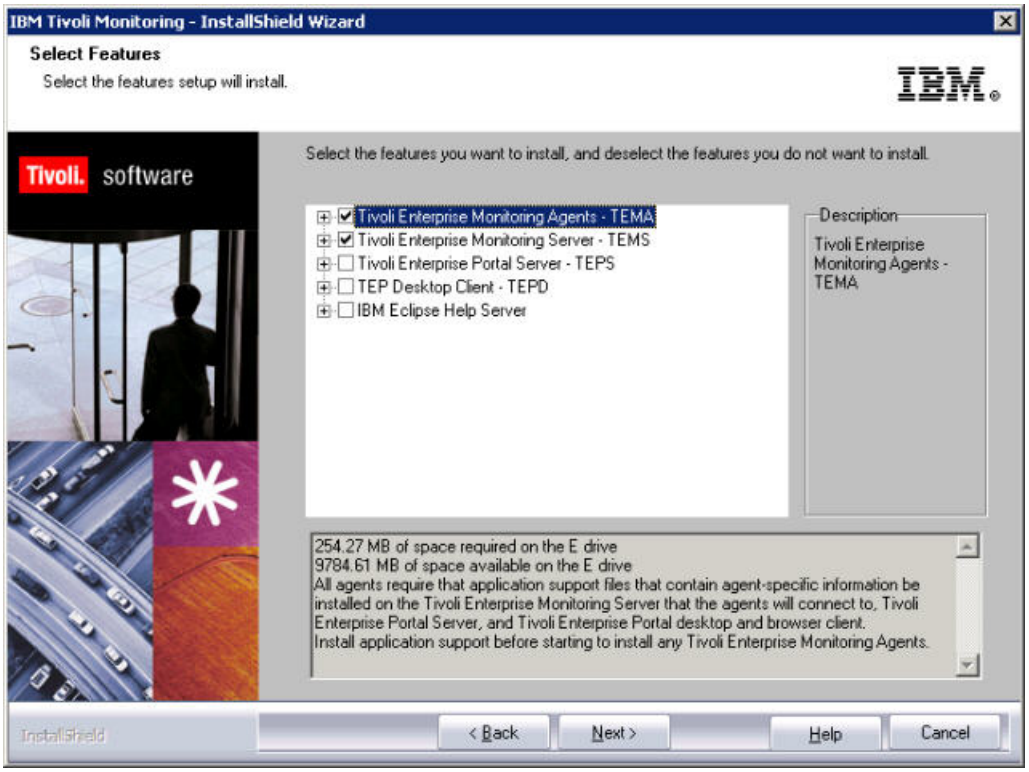

Figure 21. Starting the IBM Tivoli Monitoring InstallShield Wizard

5. Provide the User data Encryption Key and perform a typical setup with application support for the agents you need. The TEPS Desktop and Browser Signon ID and Password window is displayed.

Figure 22. Specifying the TEPS Desktop and Browser Signon ID and Password

6. The setup prompts you for the monitoring server name. The monitoring server name is not the host name, so you can choose your preferred hub Tivoli Enterprise Monitoring Server name. Avoid using the node name as the Tivoli Enterprise Monitoring Server name because Tivoli Enterprise Monitoring Server will be running on both nodes of the cluster.

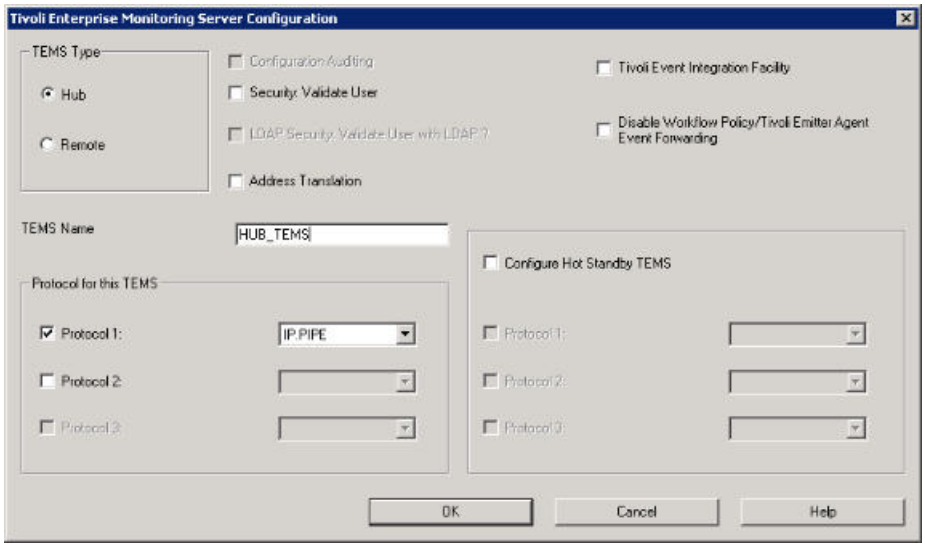

Figure 23. Specifying the Tivoli Enterprise Monitoring Server name

7. When the setup window prompts you for the host name of the monitoring server, type in the virtual host name that you just created and brought online as part of the cluster resources setup. In this case, the virtual host name is *IVTCS005*.

8. Select **OK** when the following warning message displays.

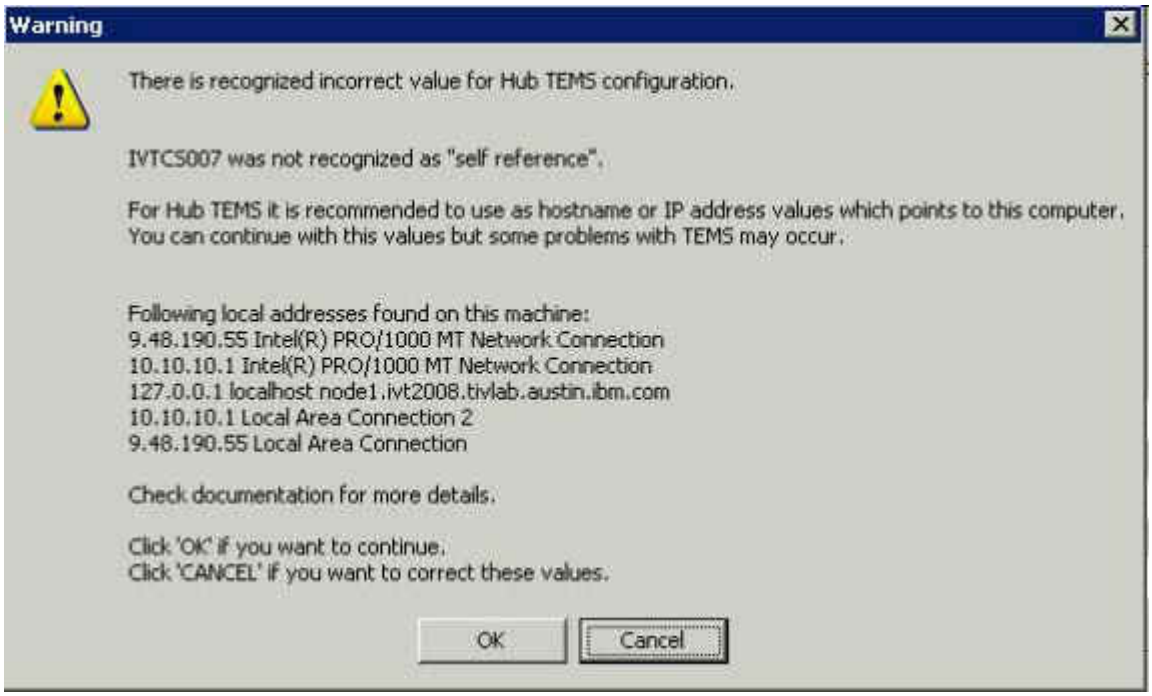

Figure 24. Incorrect value for the Hub TEMS configuration

To complete the setup, the monitoring server starts up and loads the selected application support.

9. When the setup is complete, open the **Manage Tivoli Enterprise Services** tool. Then, stop the monitoring server and switch the startup to **manual**. Changing the monitoring server startup to manual is required because the cluster must handle the startup and shutdown processes of the monitoring server.

What to do next

To prepare for the next step (which requires a restart of clusternode1), stop cluster service for clusternode2 to prevent Microsoft Cluster Server from moving the resource to the clusternode2.

1. Right-click the monitoring server again. Select **Advanced**, and then select **Set Network Interface**.
2. Set the network interface to the virtual host name. In this example, the virtual host name is IVTCS005.
3. Run regedit.exe. Navigate to the following directory:
   `HKEY_Local_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment`

   The key
   `KDEB_INTERFACELIST`

   must specify IVTCS005.

   Note: The registry replication entries might not be identical to the ones in this document.

   This setting affects all components installed on the same computer (node).

   Although the monitoring server is configured to use the virtual host name, the monitoring server logs are named after each specific node in the cluster.
4. Restart machine IVTCS003.
5. When the computer is back online, make sure that the cluster resources are still active on clusternode1. Then, start cluster service on clusternode2.

Adding the monitoring server resource to your resource group

About this task

The final step for the setup, before you configure clusternode2, is to configure the resource for the monitoring server service in the HUBTems group on the cluster:

Procedure

1. Open the Cluster Administrator on clusternode1 again.
   a. Right-click the group **HUBTems** group.
   b. Select **New > Resource**.
   c. Select **Generic Service** as the resource type.
2. Assign a name to the resource. In this testing environment, use *TEMSservice*.

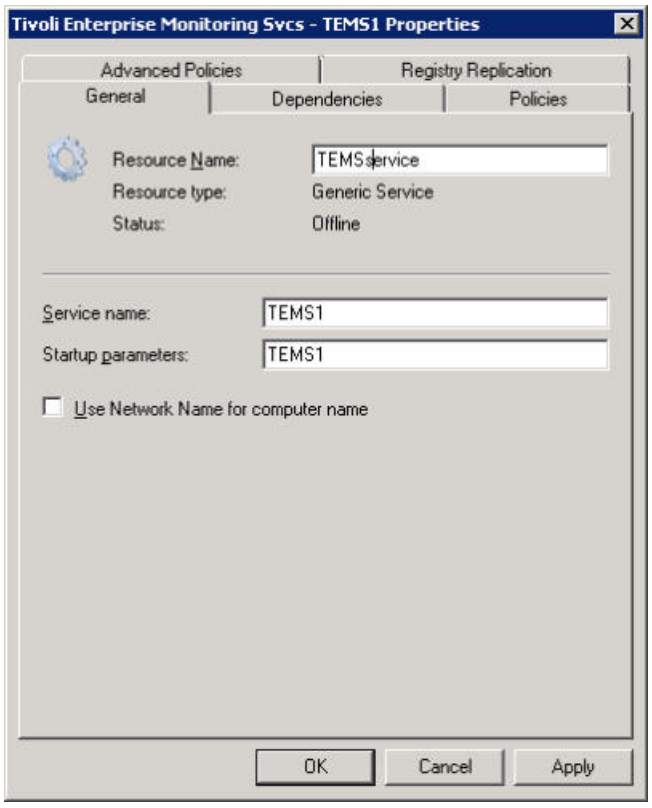

Figure 25. Define the Generic Service resource TEMSservice

3. Define the dependencies:

   Make sure that the service dependencies include all other resources, shared disk, IP address, and network name to ensure that the monitoring server service does not start up before the other resources are online.

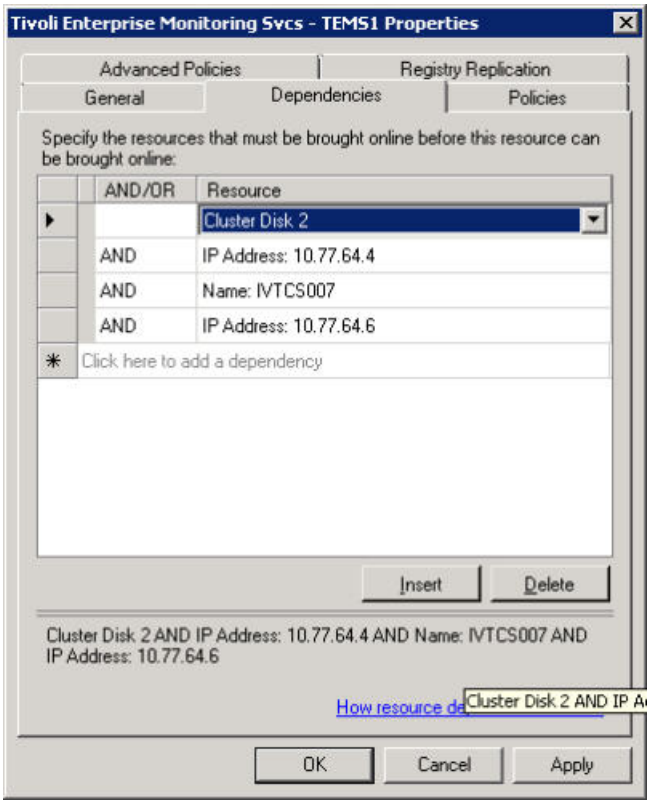

Figure 26. Confirm the Disk R, TEMS IP, and TEMS Network Name

4. On the Advanced Policies tab, make sure that both nodes are on the list of **Possible Owners**.
5. Do not change the Generic Service Parameters. For the service name, specify "*TEMS1*," which is the service name used by the monitoring server.
6. After you have added the monitoring server resource to your resource group, replicate the registry keys.

Registry Keys replication

On the Registry Replication panel, specify the registry entries that must to be replicated to the second cluster node.

See the following list of high level registry entries used by IBM Tivoli Monitoring Tivoli Enterprise Monitoring Servers:

```
SOFTWARE\Wow6432Node\Candle
SYSTEM\CurrentControlSet\Services\TEMS1
SYSTEM\CurrentControlSet\Control\Session Manager\Environment
```

Note: The registry replication entries might not be identical to the ones in this document.

In the registry settings, include only registry names below the HKEY_LOCAL_MACHINE entry.

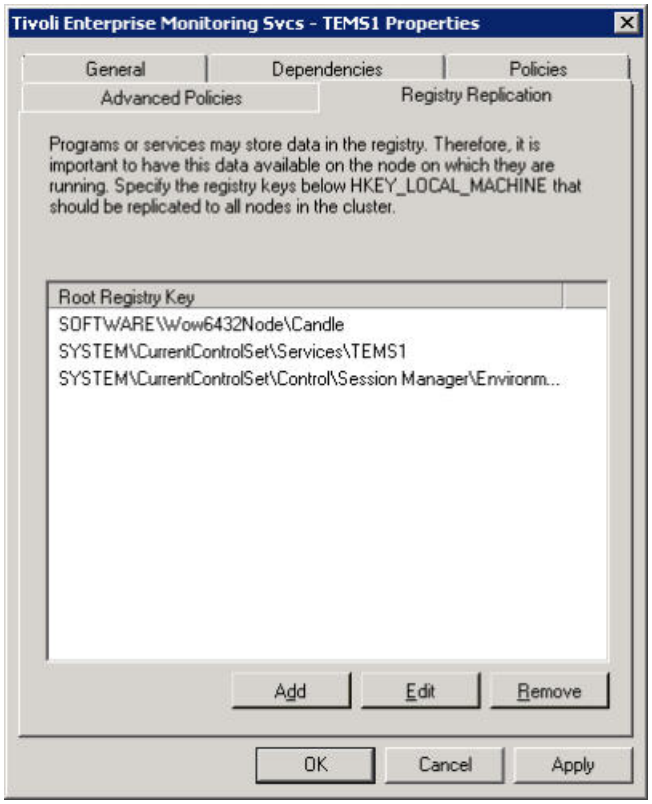

Figure 27. Listed registries on the Registry Replication panel

Click **OK** and complete the resource creation. After that, you can bring the resource online as shown in the following window:

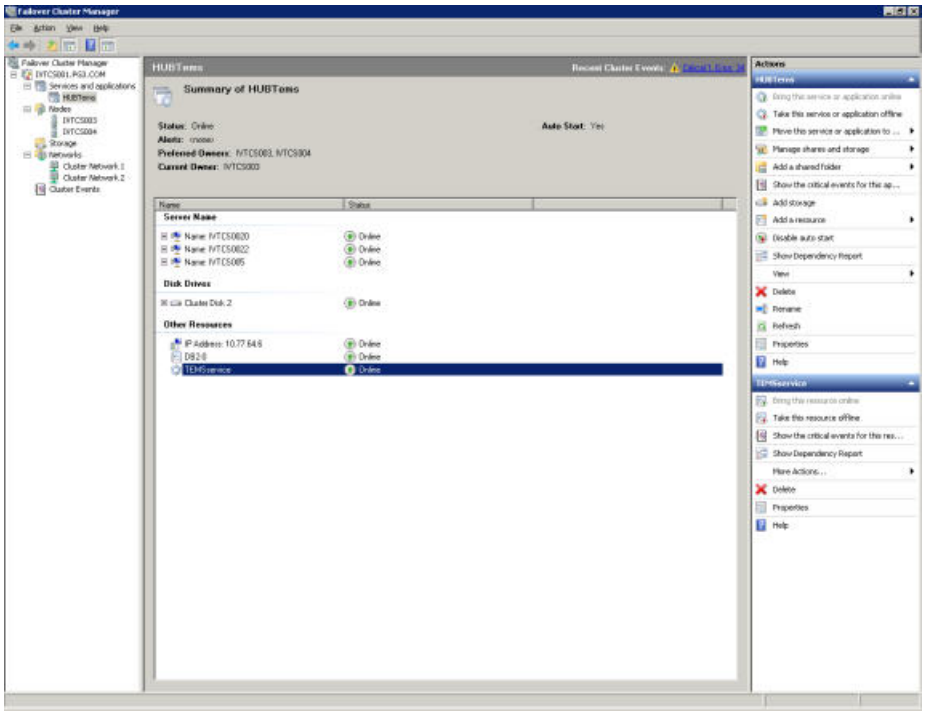

Figure 28. Bring TEMS Service resource online

When this resource is online, open the Manage Tivoli Monitoring Services GUI to verify that the monitoring server is started by the cluster software and running.

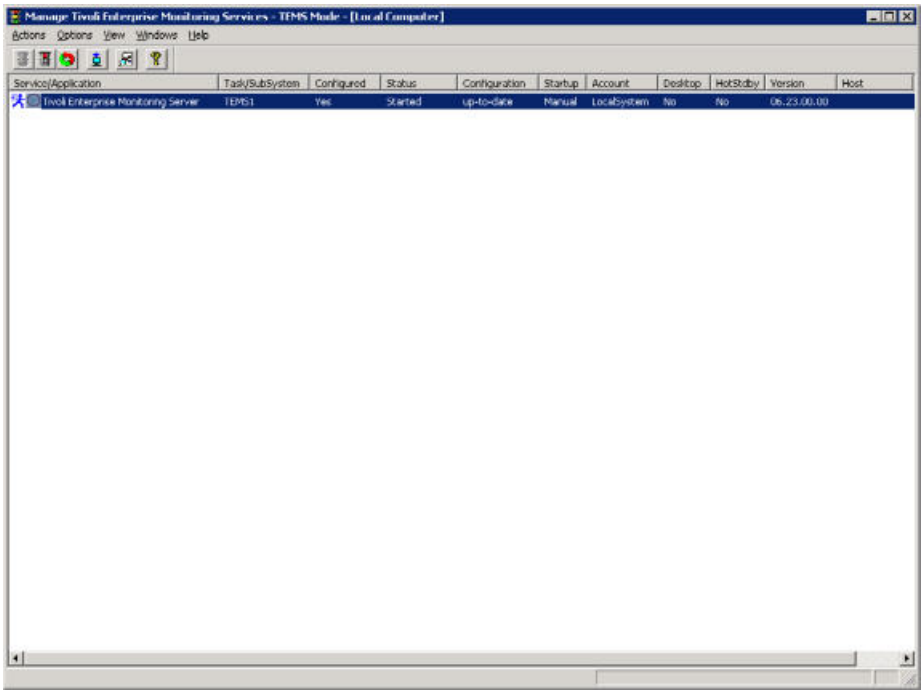

Figure 29. Manage Tivoli Enterprise Monitoring Services

Testing the monitoring server on clusternode1

About this task

Now, you can test the monitoring server.

For example, connect an agent to the monitoring server from another system. Specify the virtual host name for the agent connection to the hub Tivoli Enterprise Monitoring Server. In the testing environment, the monitoring server virtual host name is *IVTCS005*.

On clusternode1:

Procedure

1. Open a command line window and switch to R:\IBM\ITM\bin
2. Log on to the monitoring server using **tacmd login**.
3. List the managed systems using **tacmd listsystems**.
4. You should see the hub and the agent as two Managed Systems.

Setting up the monitoring server on clusternode2

Because the monitoring server is already installed on the shared disk that is accessible by clusternode2, do not installed the monitoring server on clusternode2.

The registry settings for the monitoring server on clusternode1 are replicated to clusternode2 by the cluster software. To ensure that clusternode2 includes the same monitoring server configuration as clusternode1, you must check the environment variables and monitoring server user validation on clusternode2:

- "Checking environment variables" on page 107

- "Monitoring server user validation"

Checking environment variables

The monitoring server installation creates environment variables that must be available on the computer running the monitoring server. Because you installed the monitoring server on clusternode1, these environment variables are set for clusternode1 under the registry key:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment
CANDLE_HOME
KDEB_INTERFACELIST
KUI_FILEPATH
LIBPATH
PATH

Note: The registry replication entries might not be identical to the ones in this document.

Make sure that these variables are also available on clusternode2 such that the Tivoli Enterprise Monitoring Server can correctly function when running on clusternode2. Verify the availability of these variables by including the Environment Registry Key in the registry keys for replication as described in "Adding the monitoring server resource to your resource group" on page 103. However, including the Environment Registry Key will replicate the entire environment from clusternode1 to clusternode2. A more selective way of ensuring the environment variables are available to clusternode2 is by exporting the registry keys of clusternode1, editing the file to leave only the values above, and then importing the file into the registry of clusternode2. Finally, restart clusternode2 so that the changes take affect.

Monitoring server user validation

If, in your environment, you run the monitoring server with user validation configured, make sure that the users are configured locally on both computers, and that they are using the same user ID, assigned user group, and password.

Testing the monitoring server on clusternode2

Procedure

1. While clusternode2 is online, check that the *TEMS1* service, Tivoli *Enterprise Monitoring Svcs – TEMS1*, is listed on Windows Services. If it is, then the monitoring server on clusternode2 is ready.

   If the service is not listed on Windows Services, then check whether the correct registry entries are replicated. If the registry entries are not replicated, you must force replication to occur by completing the following steps:
2. On clusternode1, in Cluster Administrator:
   a. Right-click on the TEMS resource group.
   b. Select **Move Group**.

      This **Move Group** fails and the cluster is back on clusternode1 again.
3. You can restart clusternode2 for the registry entries to take effect.

   When clusternode2 is available again, it is ready.
4. On clusternode1, in Cluster Administrator:
   a. Right-click on the TEMS resource group.
   b. Select **Move Group.**

   The Cluster Resources on the "*HUBTems*" resource group start the failover: the lower resource in the dependency goes off line first (and then the others in order). Next, one by one, the resources are available online on clusternode2.
5. You can test the monitoring server on clusternode2 by using the same method you used on clusternode1.
6. To see the monitoring server on the Manage Tivoli Monitoring Services GUI, start it on clusternode2 by running the following command:

```
R:\IBM\ITM\InstallITM\kinconfig.exe
```

Setting up the portal server in a Microsoft Cluster Server

About this task

To have the portal server highly available in a Microsoft Cluster, you must complete the following procedures:

Procedure

1. "Setting up basic cluster resources"
2. Install DB Software and add the software to the resource group
3. Install the portal server clusternode1, and configure the portal server on clusternode2
4. "Adding the portal server resource to the resource group" on page 120

Setting up basic cluster resources

Installation and setup of the Tivoli Data Warehouse components on a Microsoft Cluster Server is similar to the portal server and requires the following five clustered resources:

- Shared disk
- IP address
- Host name
- Portal Server DB
- Portal server service with its associated information

See "Setting up basic cluster resources" on page 93 for the setup of the basic cluster resources for the portal server. Use a different name from the one chosen for the hub resource group.

In addition to the basic cluster resources as in the Monitoring server case, you also need the portal server database as one of the resources. This procedure is described in the following sections.

Installing and setting up DB2 on a Microsoft Cluster Server

DB2 is a cluster-enabled database server, which means that the DB2 database is officially supported in a clustering environment.

The DB2 database must be installed on both cluster nodes.

Installing DB2 on clusternode1

Procedure

1. Switch the cluster to clusternode1.
2. Install DB2 as you typically do on a local drive on clusternode1
3. As in a default installation, create the default instance.
4. Switch all DB2 services to manual so that they can be controlled by the cluster. The DB2 services are:
   - DB2 - DB2-0
   - DB2 Governor
   - DB2 JDBC Applet Server
   - DB2 License Server
   - DB2 Remote Command Server
   - DB2 Security Server
   - DB2DAS - DB2DAS00
   - DB2 Management Service
5. Complete the following steps to change the DB2 configuration so that the default drive (for creating the database to be used by Tivoli Monitoring) is the shared disk:

a. Start the DB2 Administration Server by issuing the **db2start** command in a DB2 command line.
b. Start the DB2 Control Center.
c. Right-click the instance name (*DB2* by default).
d. Select option **Confígure parameters**.
   The following window opens:

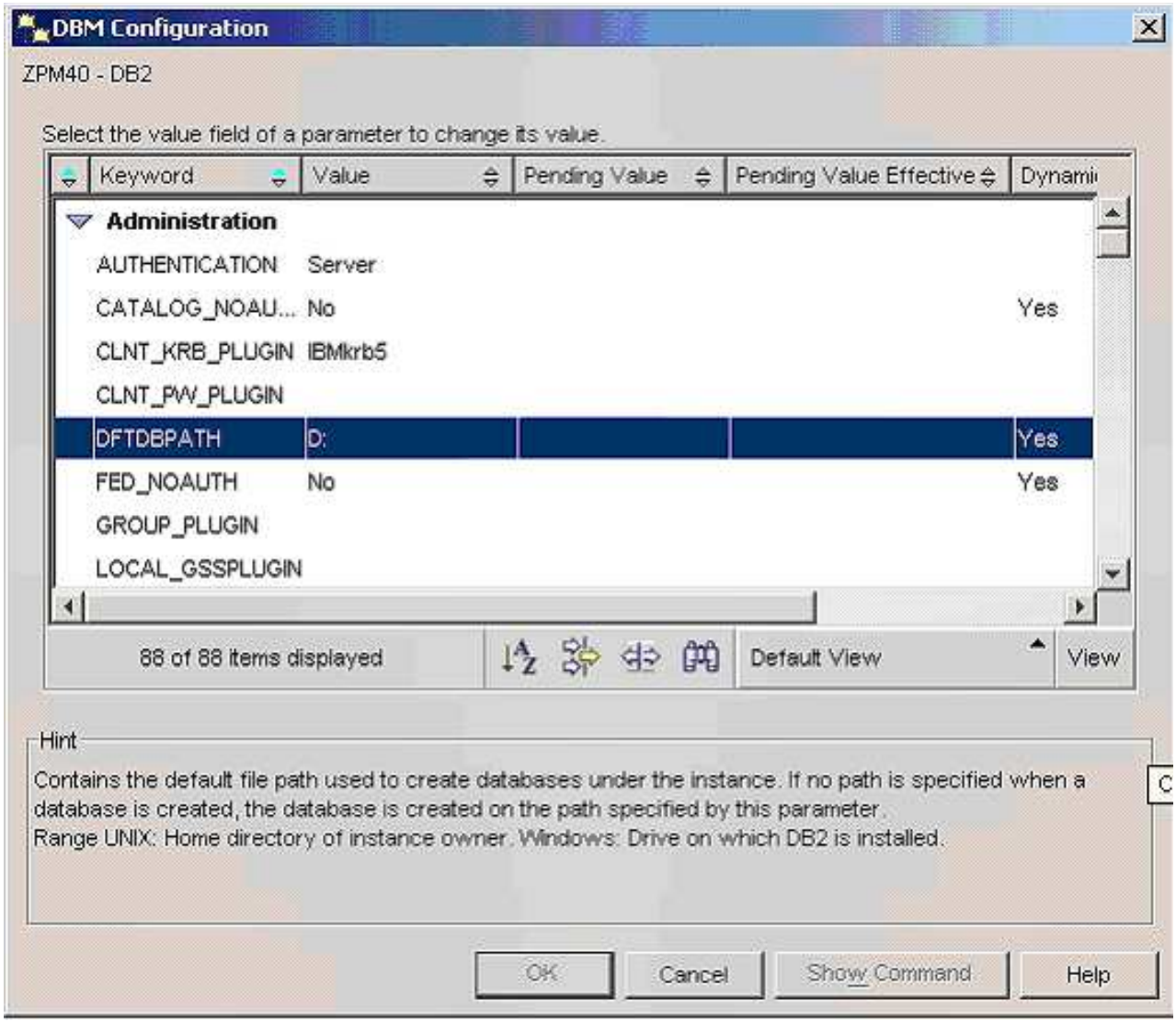

Figure 30. Configuring parameters in the DBM Configuration window

e. Select **DFTDBPATH**, click the value field, and then click the "..." to change the default drive.

   Note: If you cannot save your changes, log in to your system as the db2admin user and update the default drive value as previously described.

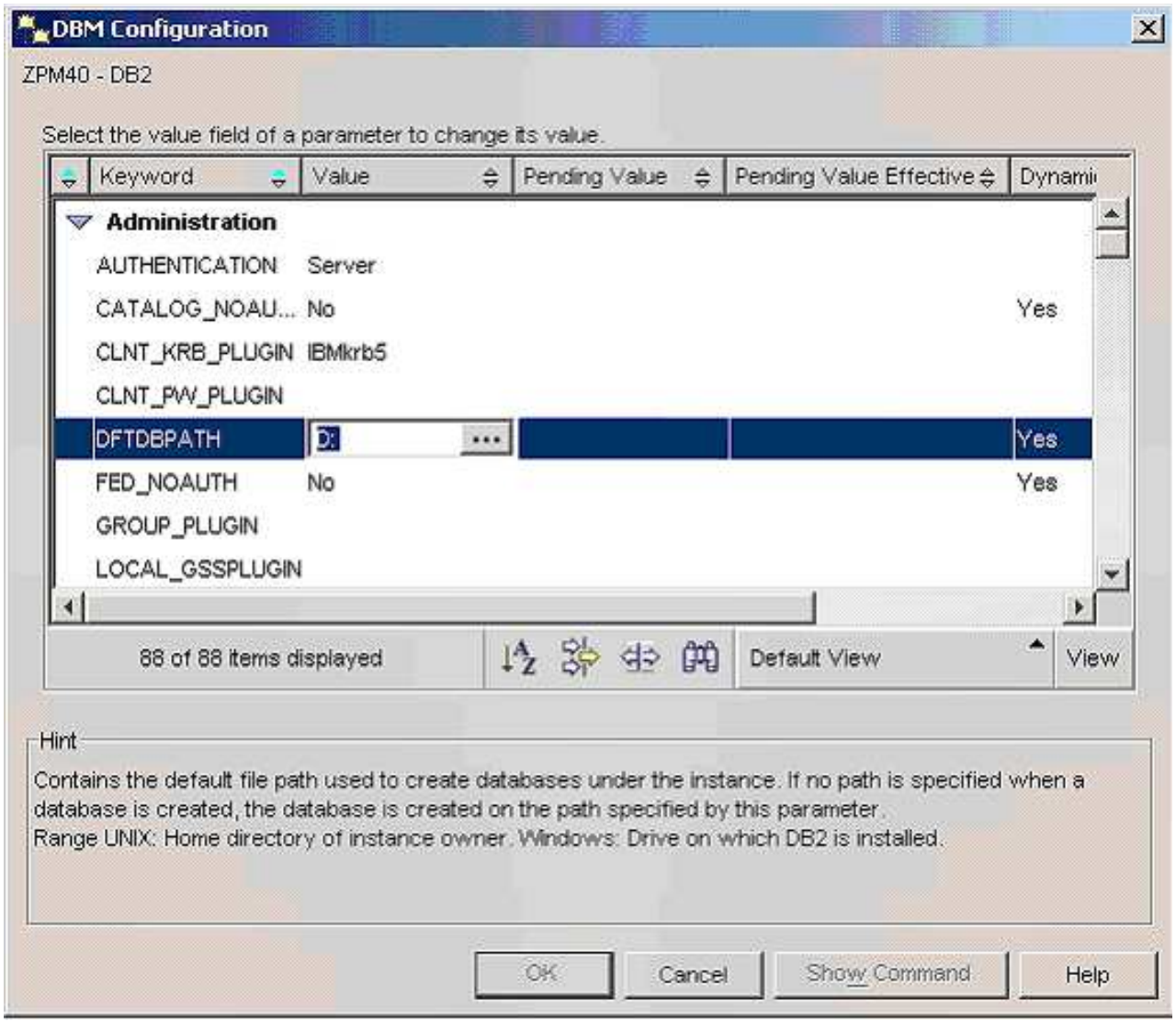

Figure 31. Changing the default drive

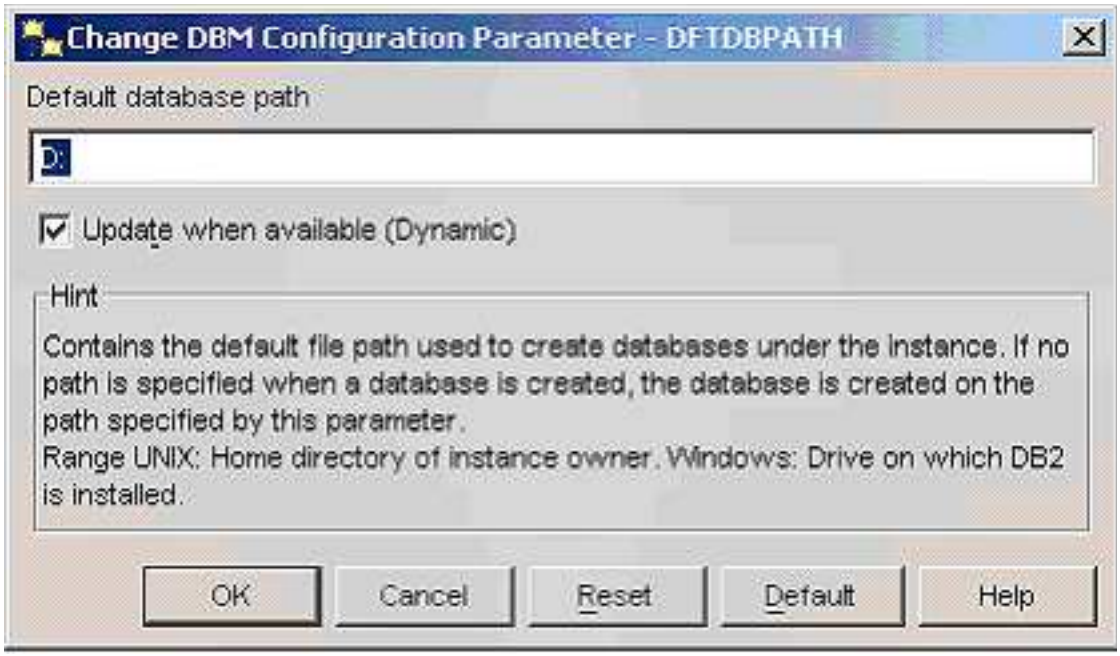

Figure 32. Setting the default drive to R

f. Change the drive to the shared drive for the cluster (*R* in the testing environment).

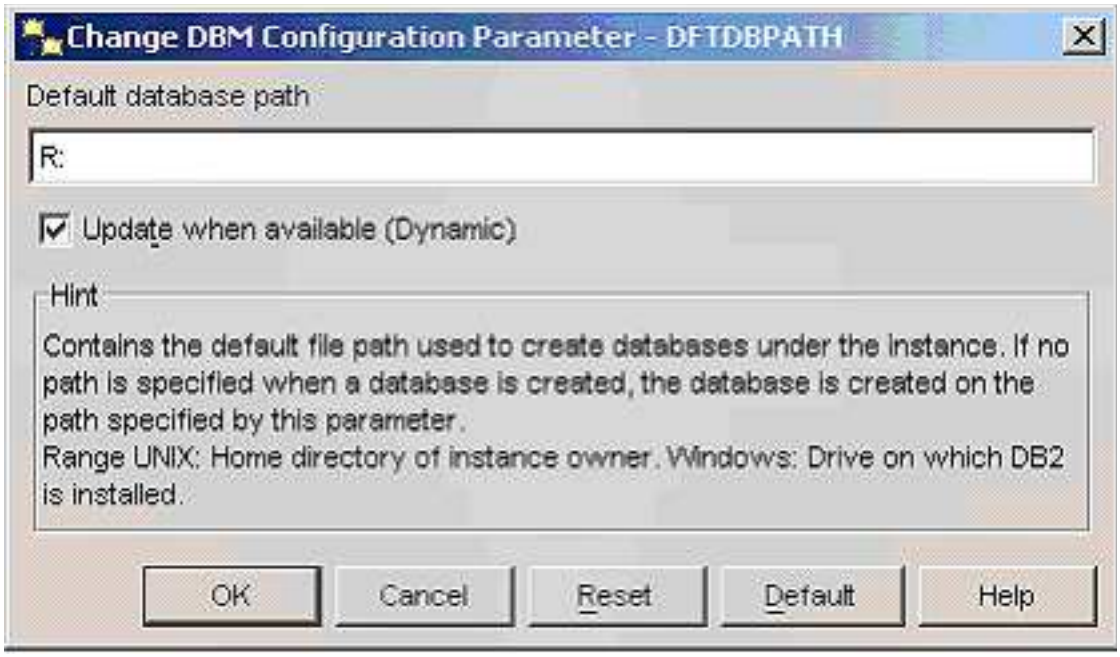

Figure 33. Setting the default drive to R

6. Stop clusternode1 service so that the changes on DB2 take effect and the cluster switches to clusternode2.

Installing DB2 on clusternode2

On clusternode2, perform the the steps previously described for clusternode1.

Adding the DB2 resource type to the Cluster Administrator

About this task

To use DB2 clustering in the Microsoft Cluster Server, you need to add the required resource types to the Microsoft Cluster Server Administrator.

DB2 provides a command to add the resource type to the cluster. To add the resource type, complete the following steps:

Procedure

1. Move the resource group back to clusternode1 and open a command line window on clusternode1.
2. At the prompt, type the **db2wolfi i** command.

   The output is shown as follows:

   ```
   c:>db2wolfi i
   ok
   ```

   Note: Error 183 indicates that the resource type has already been added.
3. Restart the cluster nodes so that these changes take effect.

Transforming the DB2 instance into a clustered instance

About this task

The DB2 instances (by default, DB2) on both cluster nodes must be transformed into a clustered instance by using a set of commands that are provided by DB2 in Windows. You must also specify a directory for the DB2 instance on the shared drive *R*.

To transform the instance, complete the following steps:

Procedure

1. Ensure that the cluster is switched to clusternode1 so that the shared disk drive is accessible on clusternode1.
2. Stop all DB2 services if they are not already stopped.
3. Create a directory on the shared drive *R*, such as `\db2Teps`, to hold the shared DB2 instance information.
4. Open a command-line window, and type the following command:

   ```
   db2iclus migrate /i:db2 /c:<clustername> /m:<clusternode1> /p:r:\db2Teps
   ```

   You do not need to specify the cluster name; the cluster you are currently in is the default cluster.

   You should see the following CLI output:

   ```
   DBI1912I The DB2 Cluster command was successful.
   Explanation: The user request was successfully processed.
   User Response: No action required.
   ```
5. Add the second cluster node to the DB2 cluster by running the following command, again on clusternode1:

   ```
   db2iclus add /i:db2 /m:<clusternode2>
   /u:<db2adminName>,<password>
   ```

   You should see the following CLI output:

   ```
   DBI1912I The DB2 Cluster command was successful.
   Explanation: The user request was successfully processed.
   User Response:No action required.
   ```

Results

You are now ready to configure the DB2 resource in the Cluster Administrator.

Instructions for how to migrate DB2 in a Microsoft Cluster Server are also available in the Migrating DB2 servers in Microsoft Cluster Server environments topic at the DB2 for Linux, UNIX, and Windows information center (http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.uprun.doc/doc/t0022647.htm).

Adding the DB2 resource to the resource group

About this task

To configure DB2 in the Cluster Administrator, complete these steps:

Procedure

1. Open the Cluster Administrator Tool.
2. Add a new resource to the resource group. Make sure that the name of the resource is the same as the name of the DB2 instance. This name can be "*DB2-0.*"

   To verify the name, check the name in the windows services, as shown in the following window:

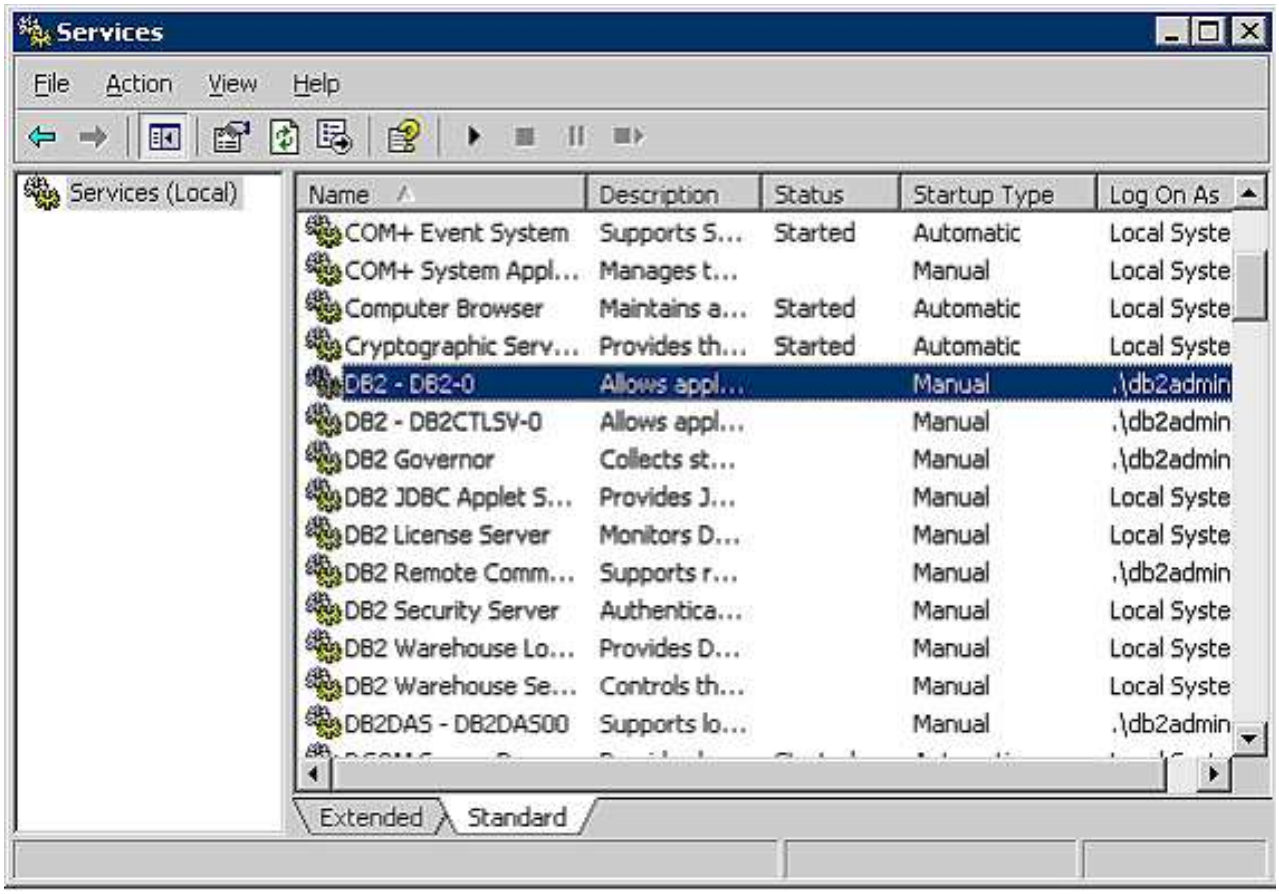

Figure 34. Selecting DB2-DB2-0 in the Services window

3. In the Failover Cluster Manager, right-click on the HUBTems entry, then select **Add a resource...More resources** and the resource type **Add DB2 Server**, as shown in the following window:

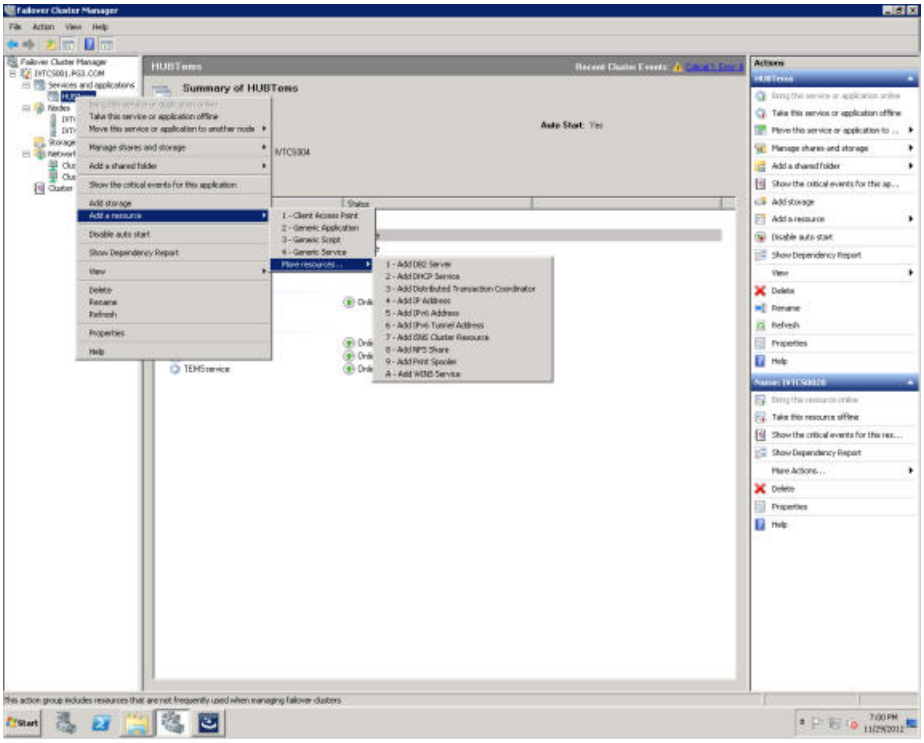

Figure 35. Selecting Add DB2 Server

4. Add both cluster nodes to the resource.
5. Set the shared disk *R*, the IP address, and the host name as dependencies, as shown in the following window:.

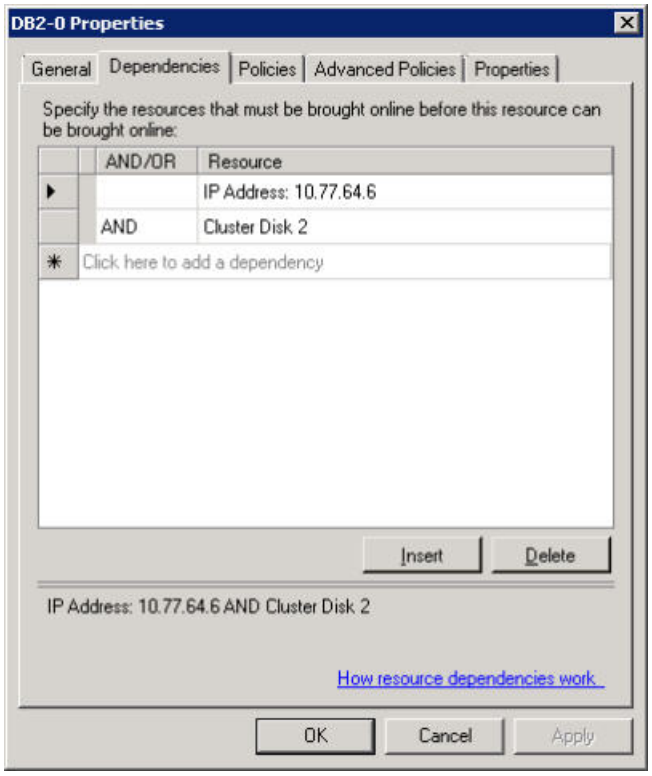

Figure 36. Setting dependencies

6. Click **Finish**. The following window is displayed.

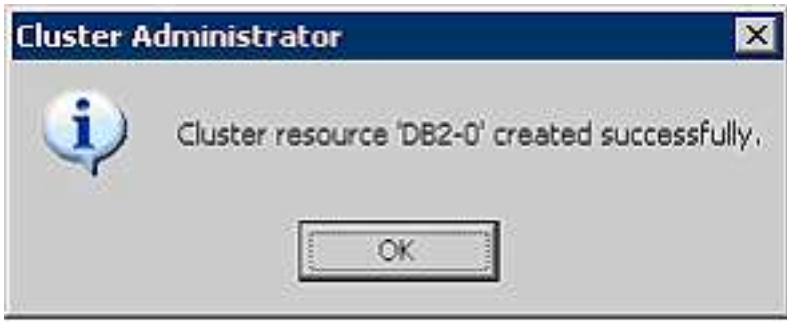

Figure 37. Click OK in the Cluster Administrator window

When you finish the resource configuration, you can bring the resource online on clusternode1.

7. Move the resource group to clusternode2, and make sure DB is working on clusternode2.

   If you have problems with this configuration, you might need to restart both nodes in the cluster so these changes can take place.

Results

You now have the DB2 instance running in the cluster.

Installing and setting up the portal server in the cluster

Installing the portal server in the cluster is very similar to the monitoring server installation. The major difference is that you must install the DB2 database and the ODBC data source.

The ODBC data source must be set up and tested before the portal server is installed.

Setting up the portal server database and ODBC DSN in the cluster
About this task

Set up the portal server database by completing the following steps:

Note: Make sure the Portal Server resource group is running on clusternode1.

Complete this procedure on clusternode1:
Procedure

1. Create a local user on clusternode1 for the portal server database access, for example, *ITMUser*.
2. Create the portal server database (for instance, *TEPS*) under the clustered instance on the shared drive R.
3. Grant access to the portal server database to the user.
4. In the ODBC data sources on the computer, create a System DSN (Data Source Name) named *TEPS2*, point to the previously created portal server database, and test the connection. See the *IBM Tivoli Monitoring: Installation and Setup Guide* for more details.
5. If the portal server must connect to a Tivoli Data Warehouse, make sure that the required ODBC DSN to the Tivoli Data Warehouse DB is also configured and tested.
6. Move the portal server resource group to clusternode2.

Complete this procedure on clusternode2:
Procedure

1. Create a local user (same as in clusternode1) on clusternode2 for the portal server DB access.
2. Start the DB2 Control Center and make sure that access has been granted to the user.
3. In the ODBC data sources, create a System DSN named TEPS2, pointed to the portal server database, and test the connection.
4. If the portal server needs to connect to a Tivoli Data Warehouse, make sure that the required ODBC DSN to the Tivoli Data Warehouse DB is also configured and tested.
5. Move the portal server resource group back to clusternode1.

Installing and setting up the portal server in clusternode1
About this task

Before installing or configuring any components in a Microsoft Cluster environment, all other components which are already added to resource groups on this cluster should be offline. This precautionary measure prevents conflicts between the cluster management tool and the Tivoli Monitoring MTEMS tool. Also, you should shut down clusternode2 to prevent the Microsoft Cluster Server moving the resource to clusternode2 during installation. Ensure that basic cluster resources are online while installing components.

Before installing the portal server, complete the following prerequisite steps:

- Within the Failover Cluster Manager, move resources to the IVTCS001 node (Node 1) by right-clicking on HUBTems group and choosing Move Group.
- Right-click on TEMSservice resource in HUBTems group and bring the resource offline.
- Within the Failover Cluster Manager, stop the cluster service after right-clicking on IVTCS004 node (Node 2).

Install the portal server by completing the following steps:

1. Start the installation by using the **setup.exe** command.

2. On the feature tree, choose only TEPS and TEPD (Eclipse Help Server will be chosen automatically).

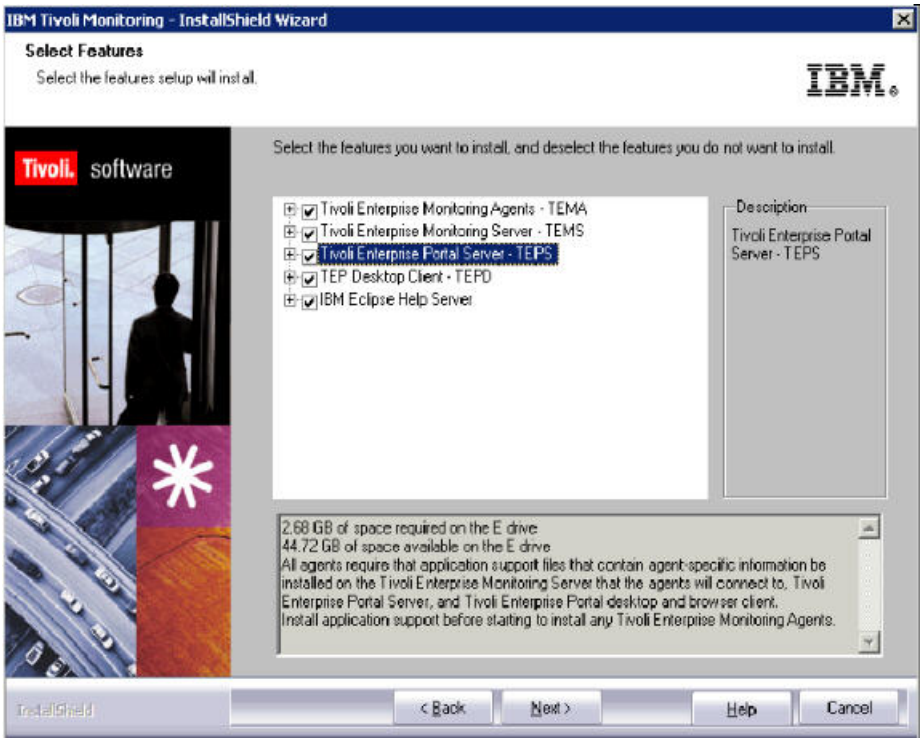

Figure 38. Selecting the features that setup will install

3. Click Next and enter the host name of the machine where the TEP Server resides.

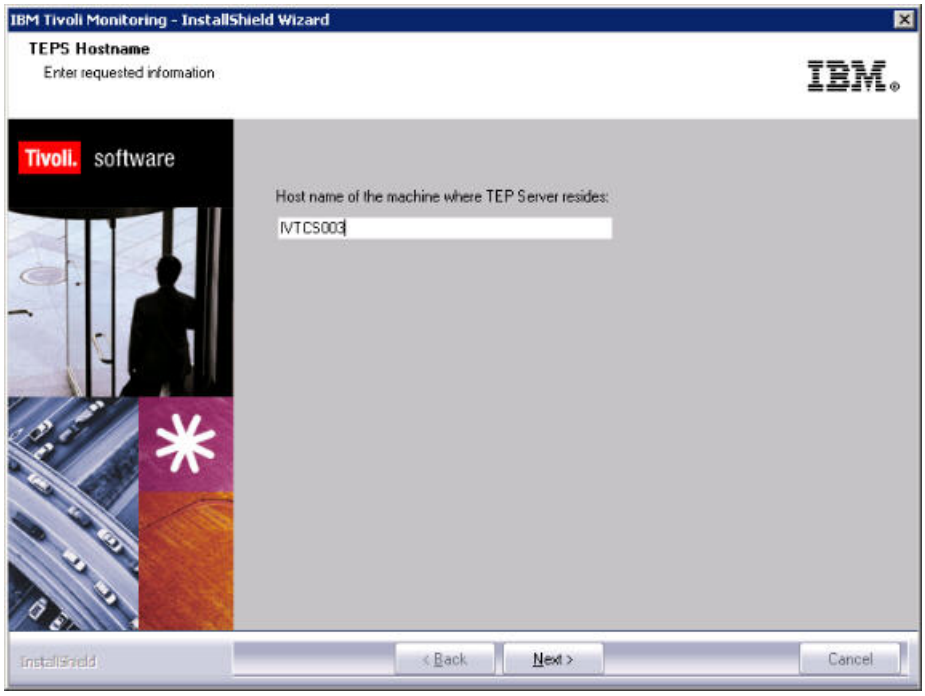

Figure 39. Selecting the features that setup will install

4. See "Setting up basic cluster resources" on page 93 for specifying the installation drive and location.

5. Start the setup and make sure that the installation directory is on the shared disk *R*.
6. Select the required Application Support.
7. Configure the virtual host name as the host name of the portal server.
8. Assuming the hub monitoring server is clustered, specify the virtual host name of the hub monitoring server as the monitoring server host name.
9. For the portal server database access, specify the DSN (*TEPS2*) and the user (*ITMUser*) just created.
10. The installation now uses the existing database and user ID to continue.
11. Finish the installation as usual.

Configure the portal server to use the virtual host name .

1. Make sure Tivoli Enterprise Portal Server and Tivoli Enterprise Portal Server Extensions are stopped.
2. Back up your configuration data `%CANDLE_HOME%\CNPSJ\profiles` directory.
3. Run the cmd shell.
4. Change the current directory to `%CANDLE_HOME%\CNPSJ\scripts`
5. Run `updateTEPSEHostname.bat <old_hostname> <new_hostname>`.

   Under <old_hostname> substitute the current host name, do not include domain name.

   Under <new_hostname> substitute the valid virtual host name for your Tivoli Enterprise Portal Server cluster.
6. If the operation is successful, 'BUILD SUCCESSFUL' is displayed.
7. Configure the portal server interface and URL such that the portal clients can connect to the portal server on its the virtual host name . On the Manage Tivoli Enterprise Monitoring Services GUI, select portal server and then:

   Advanced->Configure TEPS Interfaces

   Click **Edit** on *CNPS* and type in the portal server virtual host name. in this case *IVTCS020*.

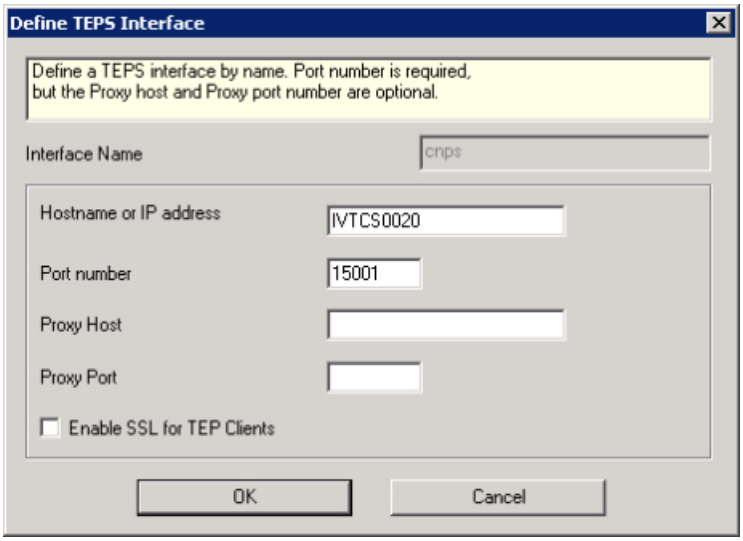

Figure 40. Editing the portal server virtual host name

8. The second configuration is for the communication between the portal server and the rest of the Tivoli Monitoring infrastructure, including the monitoring server.

   Right-click the portal server, select **Advanced**, and then **Set Network Interface**. Set the network interface also to the virtual host name, (*IVTCS020*).

This setting creates an environment variable for the portal server, so it does not take effect until the computer is restarted. Also, this setting affects every IBM Tivoli Monitoring component that is installed on the same computer (node).

9. Start the portal server.
10. Configure a Tivoli Enterprise Portal Console to connect to this portal server virtual host name.
11. Start the Console and make sure the Console to portal server communication is working.
12. As a last step of the portal server configuration on the cluster, set the portal server service to **manual** startup.

Note: In some cases, the portal server environment entries might not be displayed correctly in the registry, especially the entries for the authorization key and the DSN.

When the portal server entries are not displayed correctly, the startup of the portal server fails. If you encounter this problem, complete the following steps, based on Microsoft Windows-specific behavior.

1. Check this key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Candle\KFW\CNPS\KFWSRV\
   Environ ment\KFW_AUTHORIZATION_KEY
   ```
   It might reflect: @Authorizationkey@
   But it should be: **AABA-AAAA-ACDI-AFgA**

   Note: The registry replication entries might not be identical to the ones in this document.
2. Also check this key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Candle\KFW\CNPS\KFWSRV\
   Environ ment\KFW_DSN
   ```
   It might be: @ODBCDSN@
   But it should be: **TEPS2**

   Note: The registry replication entries might not be identical to the ones in this document.
3. Correct the keys, start the tool Manage Tivoli Enterprise Services, right-click the TEPS, and select:

   Advanced-> Utilities-> Build TEPS Database

   Then select DB2 and follow the few steps to configure the parameters again as you did during the installation.
4. Start the portal server.
5. Open a command prompt and change the directory to *R:\IBM\ITM\CNPS* on the shared disk.
   a. Run the `Buildpresentation.bat` command.
6. Restart the portal server.

Adding the IBM Eclipse Help Server resource to the resource group

Procedure

1. Right-click the resource group that you created for the portal server.
2. Go to **New -> Resource** and complete the following fields in the dialog:
   a. **Resource Type**: select **Generic Service**
   b. **Name**: can be any user-defined name. In this case, use "HELPSVRservice."
3. In the **Possible Owners** list, add both nodes.
4. As **Resource dependencies**, add shared disk, virtual IP, and virtual host name.
5. For Generic Service Parameters, the Service name must be KKF_HELPSVR.
6. For Registry Replication, include the following entries:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Candle
   ```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\KKF_HELPSVR
```

Note: The registry replication entries might not be identical to those listed in this document.

As noted in the following window, only the part below HKEY_LOCAL_MACHINE\ must be specified to replicate to all nodes in the cluster: If you need to be more specific on the registry replication for the IBM and Environment keys, see "Adding the portal server resource to the resource group" and "Adding the Warehouse Proxy Agent resource to the resource group" on page 129.

Note: If you add any component to an existing one in the cluster, refresh the registry entries for the `HKEY_LOCAL_MACHINE\SOFTWARE\Candle` path (by removing and adding this entry again). This entry prevents overriding the current registry settings by the cluster registry snapshot.

Results

After the settings are finished, you can bring the Eclipse Help Server resource group online.

Adding the portal server resource to the resource group

About this task

See the following final steps for setting up the portal server in the cluster. In the Cluster Administrator:

Procedure

1. Right-click the resource group that you created for portal server.
2. Go to **New -> Resource** and complete the following fields in the dialog:
   a. **Resource Type**: select **Generic Service**
   b. **Name**: can be any user defined name. In this case, use *"TEPSservice"*.

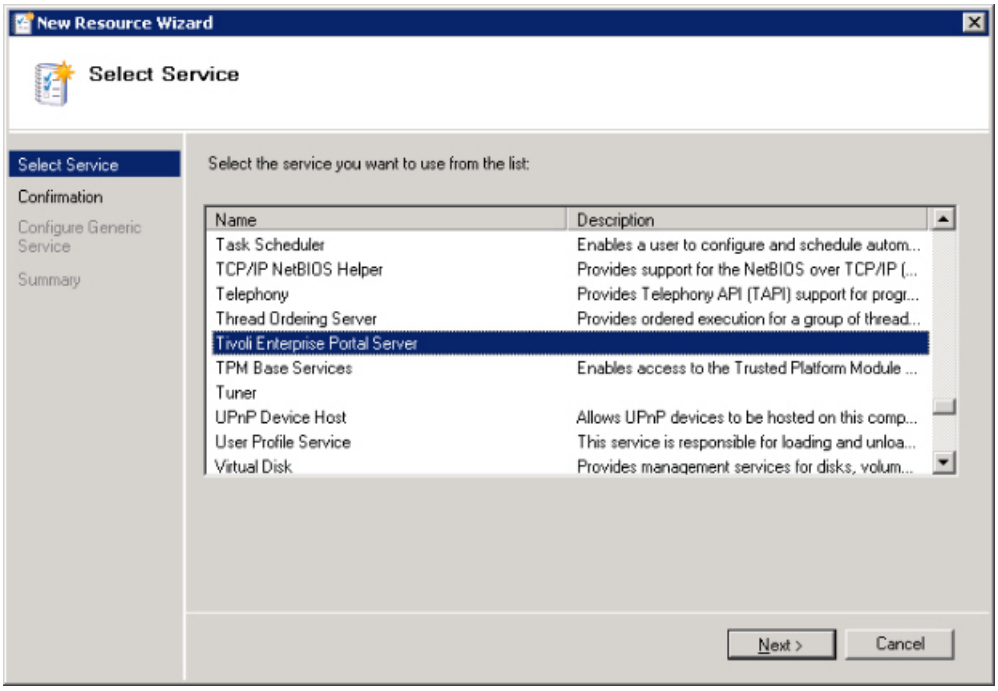

Figure 41. Entering Name and Resource type into New Resource window

3. In the **Possible Owners** list, include both nodes.

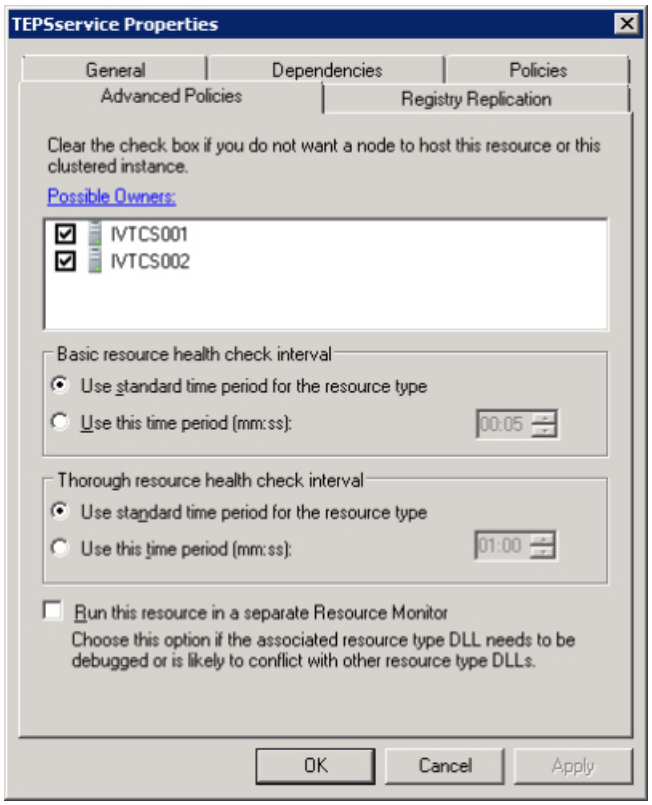

Figure 42. Both nodes appear in the Possible Owners window

4. As **Resource dependencies**, include shared disk, virtual IP, virtual host name, DB2, and HELPSVRservice.

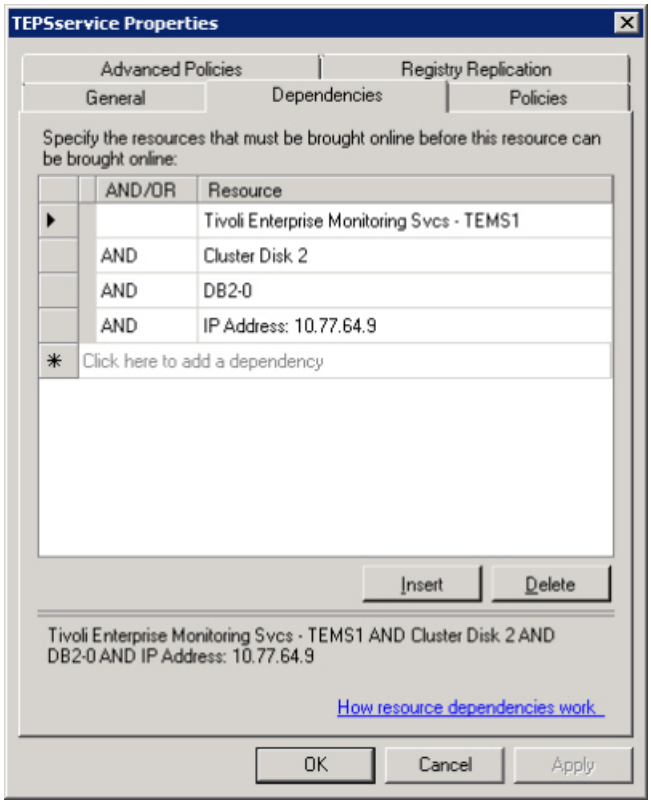

Figure 43. Add Shared disk, virtual IP, virtual host name and DB2

5. For **Generic Service Parameters**, the Service name must be *KFWSRV*.

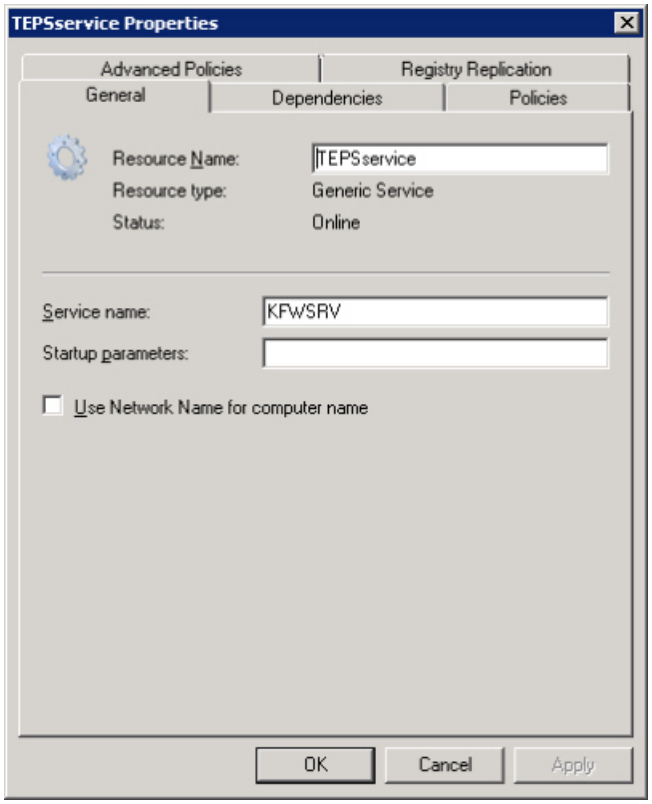

Figure 44. Entering KFWSRV into Generic Service Parameters

6. For Registry Replication, include the following entries:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Candle
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\KFWSRV
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IHSforTivoliEnterprisePortalServer
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment
   ```

   Note: The registry replication entries might not be identical to the ones in this document.

   As noted in the following window, only the part below `HKEY_LOCAL_MACHINE\` must be specified to replicate to all nodes in the cluster:

   If you need to be more specific on the registry replication for the `IBM` and `Environment` keys, see "Adding the portal server resource to the resource group" on page 120 and "Adding the Warehouse Proxy Agent resource to the resource group" on page 129.

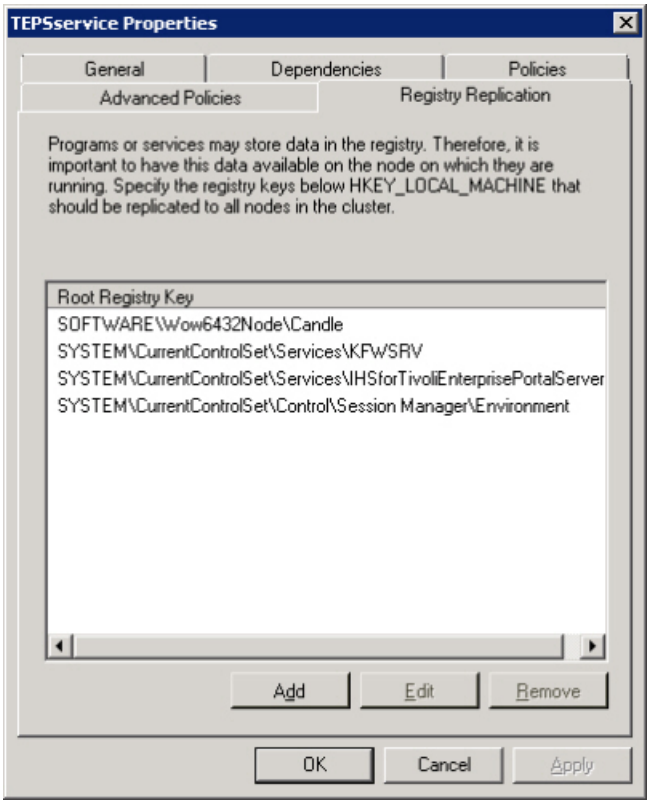

Figure 45. Registry Replication window showing necessary settings

Results

After the settings are finished, you can bring the portal server resource group online.

Now the Cluster Administrator window looks like this:

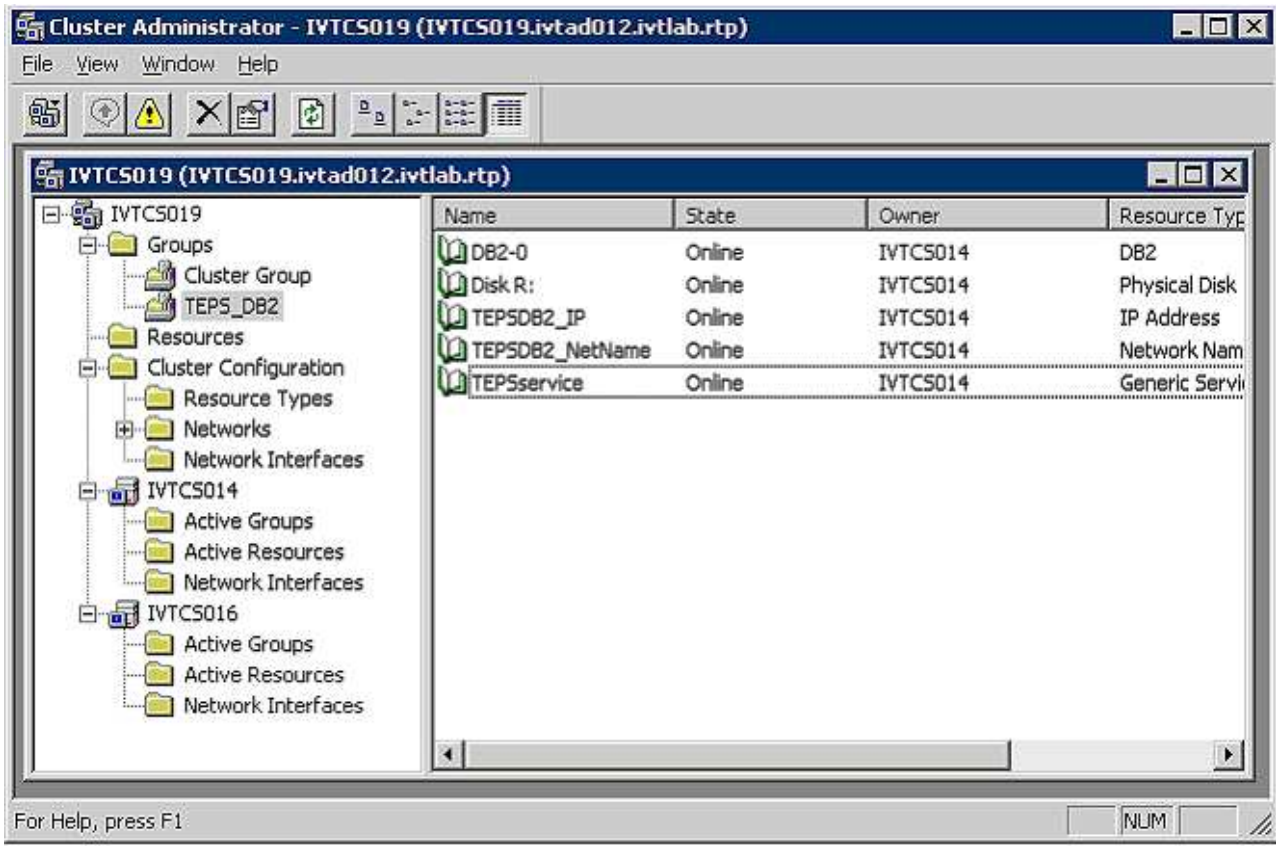

Figure 46. Cluster Administrator window after Registry Replication

Testing the portal server in the cluster

After you bring the resource of the portal server online, log on using the portal browser or portal client. Perform typical tasks, such as creating a situation and associating a situation.

Now, switch the portal server resource group to clusternode2. During the switch, you notice the typical error messages from the portal client about losing the connection. When the switch is finished, the portal client reconnects to the portal server, and you should be able to continue your activities on the client.

Note: The registry replication of services from one cluster node to the other might not work 100% correctly. Partial registry replication of services might cause the resource not to come up after switching to the second node.

In this case, verify that the service is visible in the services on the second computer.

If not, restart the computer.

Setting up Tivoli Data Warehouse components in a Microsoft Cluster Server

About this task

To have the Tivoli Data Warehouse and associated components (Warehouse Proxy Agent and Summarization and Pruning Agent) highly available in a Microsoft Server Cluster, verify that the following requirements are satisfied:

1. The basic cluster resources are set.
2. The DB software is installed and added to the resource group.
3. The Warehouse Proxy Agent and Summarization and Pruning Agent is installed on clusternode1 and configured on clusternode2.
4. The Warehouse Proxy Agent and Summarization and Pruning Agent services are added to the resource group.

The setup of the Warehouse Proxy Agent and Summarization and Pruning Agent as part of the cluster is optional. However, the procedures provided in this section are written for a cluster configuration where these agents are installed on the database server.

Setting up basic cluster resources

The Tivoli Data Warehouse components installation and setup on a Microsoft Cluster Server is similar to the portal server and requires the following five clustered resources:

- Shared disk *R*
- IP address
- Host name
- Tivoli Data Warehouse DB
- Warehouse component services with associated information (optional)

See "Setting up basic cluster resources" on page 93 for the setup of the basic cluster resources. In this testing environment, specify TDW_Cluster as the name for the Tivoli Data Warehouse resource group.

In addition, similar to the portal server, the Tivoli Data Warehouse components also require a DB resource in the cluster, as described in the following sections.

Installing and setting up DB2 on a Microsoft Cluster Server

See "Installing and setting up DB2 on a Microsoft Cluster Server" on page 108.

Installing and setting up Tivoli Data Warehouse components in the cluster

Installing the Tivoli Data Warehouse components in the cluster is very similar to the portal server installation. The installation also requires the database and the ODBC data source.

The ODBC data source must be set up and tested before Tivoli Data Warehouse components are installed.

Setting up the Tivoli Data Warehouse database and ODBC DSN in the cluster

About this task

Set up the Tivoli Data Warehouse DB by completing the following steps:

Make sure the Tivoli Data Warehouse resource group is running on clusternode1.

On clusternode1:

Procedure

1. Create a local user on clusternode1 (for the Tivoli Data Warehouse DB access, for instance, *ITMUser*).
2. Create the Tivoli Data Warehouse database under the clustered instance on the shared drive *R*, for instance, *Warehouse*.
3. Grant access to the Tivoli Data Warehouse DB to the user.
4. In the ODBC data sources on the computer, create a System DSN named *ITM Warehouse*, point to the previously created Tivoli Data Warehouse database, and test the connection.
5. Move the Tivoli Data Warehouse resource group to clusternode2.

On clusternode2:
Procedure

1. Create a local user (same as in clusternode1) on clusternode2 for the Tivoli Data Warehouse DB access.
2. Start the DB2 Control Center and make sure that the access has been granted to the user.
3. In the ODBC data sources, create a System DSN named *ITM Warehouse*, point to the Tivoli Data Warehouse database, and test the connection.
4. Move the Tivoli Data Warehouse resource group back to clusternode1.

Installing the Warehouse Proxy Agent and the Summarization and Pruning Agent in the cluster

About this task

This step is optional. Complete this step only if Warehouse Proxy Agent or Summarization and Pruning Agent will be part of the Tivoli Data Warehouse cluster.

Procedure

1. Start the installation with **setup.exe** command.

   See "Setting up basic cluster resources" on page 93 for specifying the installation drive and location.

   a. Start the setup and make sure that the installation directory is the one on the shared disk *R*.
   b. As the host name of the Warehouse Proxy Agent and Summarization and Pruning Agent, configure the virtual host name of the Tivoli Data Warehouse cluster.
   c. For the Tivoli Data Warehouse DB access, specify the DSN and the user that you configured when you set up the Tivoli Data Warehouse DB on clusternode1 and clusternode2.
   d. Finish the installation.
2. Configure the Warehouse Proxy Agent, as shown in the following window:

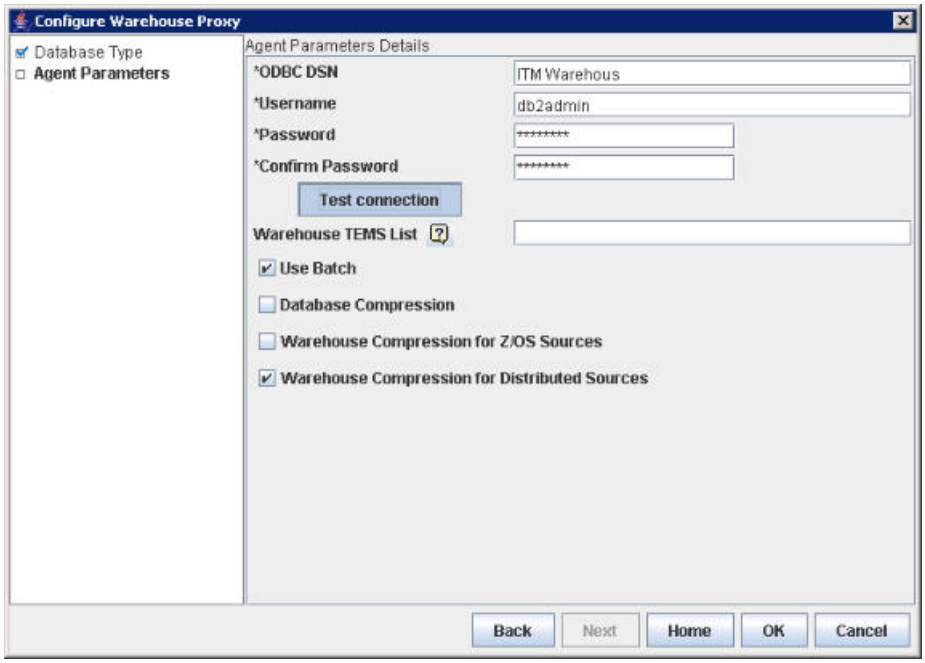

Figure 47. Configuring the Warehouse Proxy Agent

3. Configure the Summarization and Pruning Agent. However, for the portal server host, specify the virtual host name of the portal server (assuming the portal server is running in a cluster – in this case *IVTCS022*), as shown in the following window:

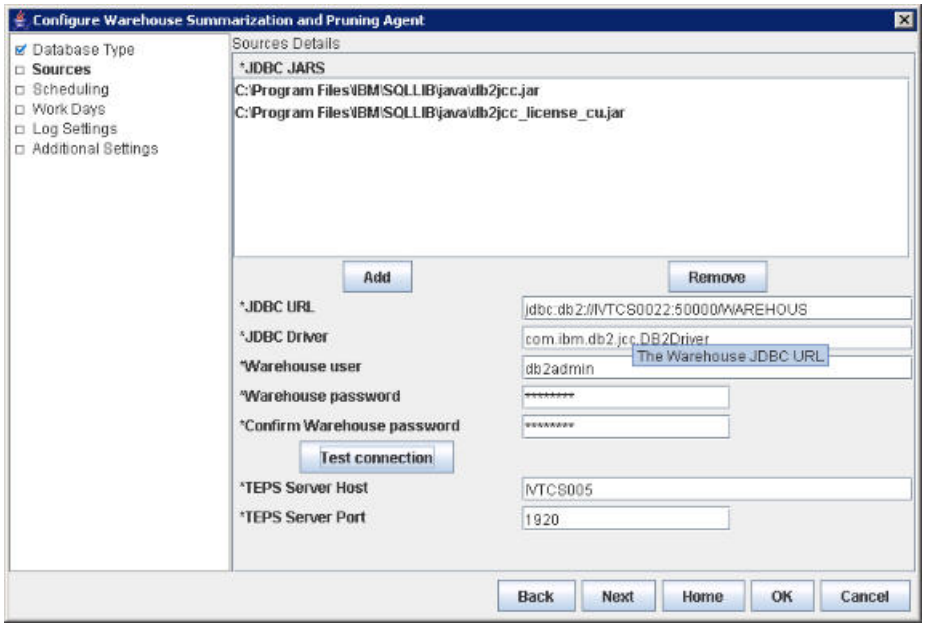

Figure 48. Specifying the virtual host name of the portal server

4. Open the Manage Tivoli Enterprise Services tool.
   a. Right-click the Warehouse Proxy Agent, choose **Advanced**, and then **Set Network Interface**.
   b. Set the network interface also to the Tivoli Data Warehouse virtual host name. See "Setting up basic cluster resources" on page 93 for more details about this setting.

This setting creates an environment variable for the Tivoli Data Warehouse Agents, so it does not take effect unless the computer is restarted. It also affects the Summarization and Pruning Agent running on the same computer.

5. Switch both agents to **manual** startup such that the cluster can control their startup and shutdown processes.

Adding the Warehouse Proxy Agent resource to the resource group

About this task

This step is optional. Complete this step only if warehouse proxy agent must be part of the Tivoli Data Warehouse cluster.

The following steps for adding the warehouse proxy agent resource to the resource group are the same as for the portal server; only the name for the service is different.

Open the Cluster Administrator and complete this procedure:

Procedure

1. Create a new resource under your resource group; the type has to be **Generic Service**. The name can be any user-defined name (in this case, use WHPservice). The **Service name** must be ***khdxprto*** as shown on the following window:

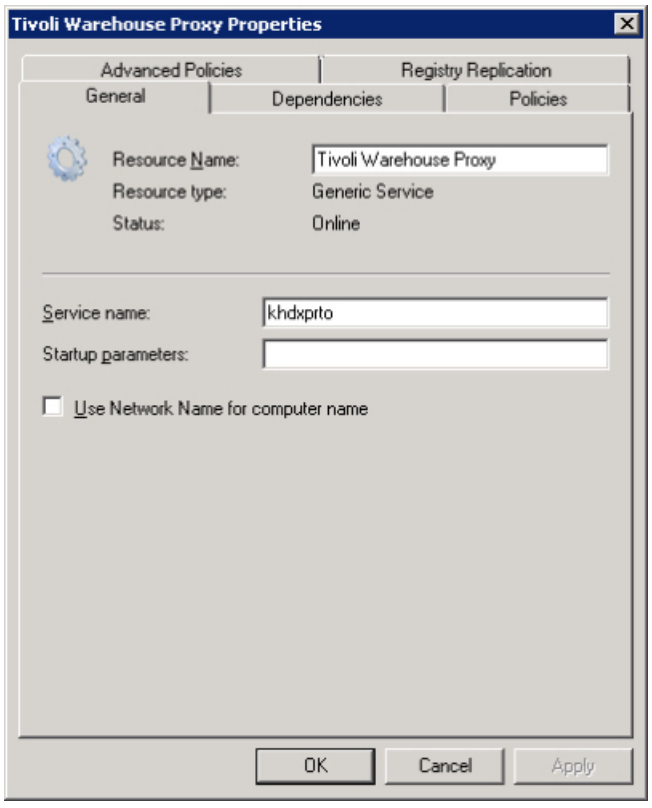

Figure 49. Entering the khdxprto service name into the Generic Service Parameters

2. In the **Possible owners** list, add both cluster nodes.
3. As **Resource dependencies**, add Shared disk, virtual IP, virtual host name, and DB2.
4. For Registry Replication, use the following entries:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\khdxprto
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment
```

```
HKEY_LOCAL_MACHINE\SOFTWARE\Candle
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Candle
```

Note: The registry replication entries might not be identical to the ones in this document.

If you need to be more specific on the registry replication for `IBM` and `Environment` keys, refer to "Adding the portal server resource to the resource group" on page 120 and "Adding the Warehouse Proxy Agent resource to the resource group" on page 129.

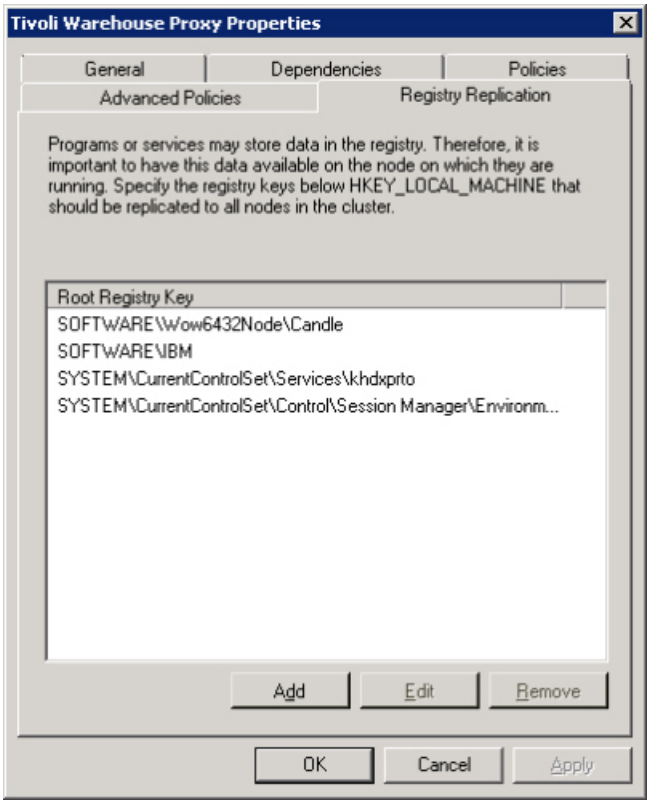

Figure 50. Registry Replication window showing required settings

5. Click **Finish**.
6. Set up CTIRA_HOSTNAME in the `khdcma.ini` file for the Warehouse Proxy Agent so that it shows up with the virtual host name on the Managed System Status table (no matter which node it is running on).
7. Bring this resource group online.

Adding the Summarization and Pruning Agent resource to the resource group

About this task

The following steps are optional and performed only if the Summarization and Pruning Agent will be part of the Tivoli Data Warehouse cluster.

Perform the same steps as in "Adding the Warehouse Proxy Agent resource to the resource group" on page 129. Substitute the appropriate names.

- On **Generic Service Parameters**, Service Name must be ***KSYSRV*** (internal name of Summarization and Pruning Agent).
- For Registry Replication, the entry for Summarization and Pruning Agent is `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\KSYSRV`.

- The other entries are the same as Warehouse Proxy Agent.

Set up the CTIRA_HOSTNAME variable (in this case IVTCS020) in the **ksycma.ini** file for the Summarization and Pruning Agent so that it shows up with the virtual host name on the Managed System Status table (no matter which node it is running on).

When finished, bring the resource group online.

Keep in mind that the first failover to the other node might fail due to the missing registry entries on this node. If this occurs, restart the computer.

If all of the Tivoli Monitoring components are installed and configured in one cluster, the following window is displayed:

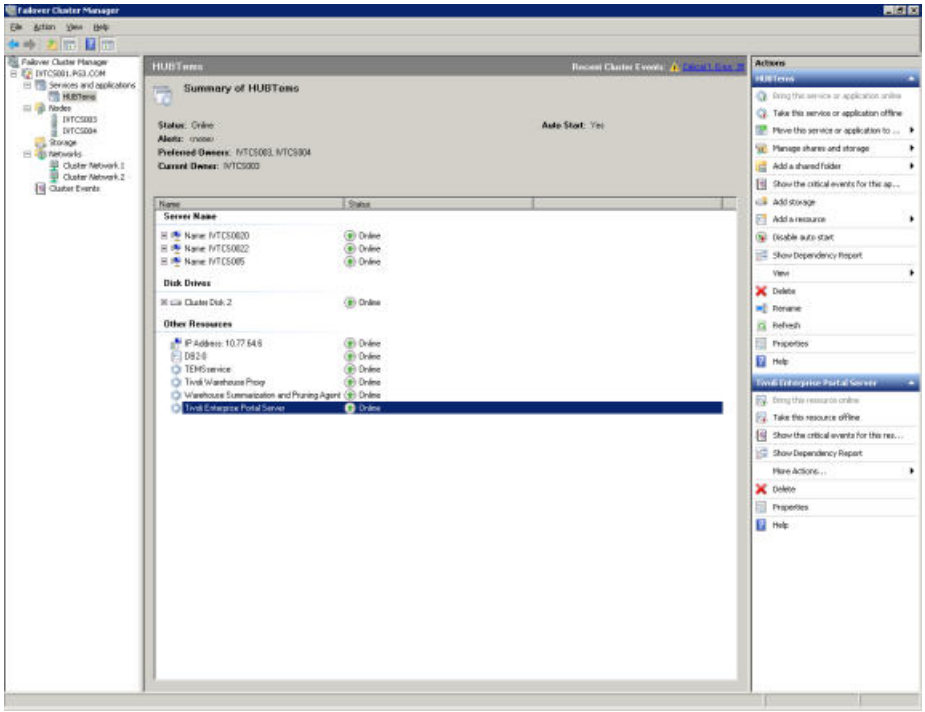

Figure 51. Failover Cluster Manager window after configuration

The cluster CLI `cluster res output` looks like this:

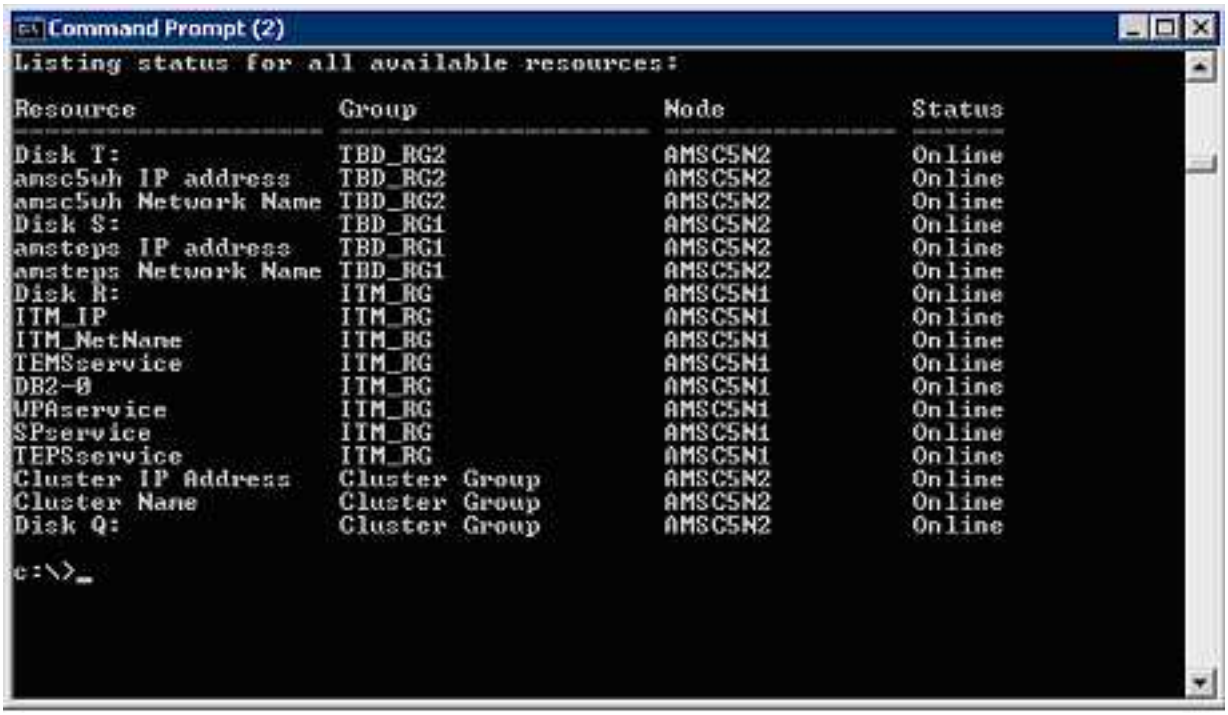

Figure 52. Correct Command Prompt window after configuration

Testing the Tivoli Data Warehouse components in the cluster

Verify that Warehouse Proxy Agent is correctly exporting data to Tivoli Data Warehouse, and Summarization and Pruning Agent is correctly summarizing and pruning. Move the resources from clusternode1 to clusternode2, and verify that Warehouse Proxy Agent and Summarization and Pruning Agent restart their processing.

Upgrading IBM Tivoli Monitoring in a Microsoft Cluster environment

Before upgrading IBM Tivoli Monitoring within a Microsoft Cluster, the following steps need to be completed to return IBM Tivoli Monitoring to a pre-cluster configuration. Failure to do these steps might cause the upgrade of IBM Tivoli Monitoring to hang or fail. After the upgrade completes, the resources should be checked to verify that they connect to the cluster resources and not to the installation node resources.

In the following procedure, *ITM_Home* refers to the location where IBM Tivoli Monitoring was installed. This path is usually `C:\IBM\ITM`. The path can be the same, but the drive letter is usually a Shared Disk seen by all members of the cluster. For the installation example, drive R: is used. So in the following steps `R:\IBM\ITM` replaces *ITM_Home*.

1. Ensure that the upgrade is done from the cluster node where IBM Tivoli Monitoring was originally installed.
2. Delete the *ITM_Home*`\CNPSJ\itmEwasRegistry.properties` file.
3. Delete the *ITM_Home*`\CNPS\profiles\ITMProfile` directory.
4. Clear the *ITM_Home*`\logs` directory (remove all files).
5. Modify the *ITM_Home*`\CNPSJ\properties\profileRegistry.xml` file to remove the data between the <profiles> and </profiles> statement.
6. Run the upgrade.

7. After the upgrade completes, change the host name for eWas from the installation cluster host name to the cluster resource host name. You must complete this change before the cluster can be switched from the installation computer another member of the cluster. Run the **updateTEPSEHostname** command in the *ITM_Home*\CNPSJ\scripts directory to perform the rename. The syntax is as follows:

   ```
   updateTEPSEHostname oldname newname
   ```

8. Verify the post installation information to validate that Tivoli Enterprise Monitoring Server and monitoring agents are configured correctly to connect to the cluster resource.
9. Test the cluster failover scenarios.

Tivoli Monitoring maintenance on the cluster

To maintain patches, fix packs, and release upgrades for Tivoli Monitoring components running on the Microsoft Cluster Server environment, you must first stop the cluster so that the control of the Tivoli Monitoring services are available to the installation program. Make sure you start the maintenance from the node where you installed Tivoli Monitoring (typically clusternode1). When the maintenance is complete, make sure the Tivoli Monitoring services controlled by the cluster are stopped and in manual startup mode. Then, restart the cluster.

Known problems and limitations

This section includes specific characteristics and constraints of the Tivoli Monitoring installation and setup on a Windows environment and their effects on the cluster setup.

During the certification test for Tivoli Monitoring clustering, issues encountered when setting up the clustered environment are formally reported as defects. These defects are typically related to the setup of Tivoli Monitoring in a non-default manner, instead of being specific to the cluster environment. These defects are handled as part of the Tivoli Monitoring service stream. See the following list of known problems and workarounds.

Installation-related issues:

- Tivoli Monitoring components cannot be installed in multiple installation directories in the same way that you can install them on UNIX or Linux systems. Under this limitation, every Tivoli Monitoring component must use the same cluster shared disk resource. Each component must be defined under the same cluster resource group. You cannot use more than one shared disk for the Tivoli Monitoring components.
- As a consequence of the preceding limitation, if you have a Monitoring Agent for Windows OS installed on the same computer as the clustered hub monitoring server, the OS agent must also be clustered and move together with the resource group.
- Tivoli Monitoring components on Windows operating systems are registered as services. You may have only one instance of each service on the computer. You cannot set up a clustered environment where two main hub monitoring servers use each other's computers as their failover hub (such as by having another instance of the hub installed and running as part of a second resource group) because you may have only one instance of the monitoring server service installed and running on a given computer.

Configuration-related issues:

- Tivoli Monitoring "Set Network Interface" capability issues:
  1. If you use the "Set Network Interface" function on the Tivoli Monitoring GUI to set a Network Interface for one agent (for instance, when running such an agent as part of the Tivoli Monitoring clustered resource group and assigning the virtual host name to it), the setting affects all other agents on the same node. *This is the current default behavior of Tivoli Monitoring.*
  2. Uninstalling Tivoli Monitoring might not remove the KDEB_INTERFACELIST entry in registry if this has been configured. To completely remove this setting, you need to remove the following registry entry:

HKEY_Local_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\
KDEB_INTERFACELIST

- If the clustered Tivoli Enterprise Portal Server does not start after failover or switching, complete the following steps:
  1. Make sure Tivoli Enterprise Portal Server starts successfully on the first computer in your cluster.
  2. Switch to the first cluster computer.
     a. For Windows systems:

        Change the current directory to `%CANDLE_HOME%\CNPSJ\profiles\ITMProfile\config\cells\ITMCell\nodes\ITMNode`
     b. For UNIX or Linux systems:

        Change the current directory to `$CANDLE_HOME/<architecture>/iw/profiles/ITMProfile/config/cells/ITMCell/nodes/ITMNode`
  3. Locate and open the `serverindex.xml` file.
  4. Check if all "hostName" and "host" properties match the virtual host name for your Tivoli Enterprise Portal Server cluster. Correct incorrect values.
  5. For each unique and invalid host name found in step 5, apply steps 2–5 accordingly.
- If the Eclipse Help Server will not start on a particular node, you might need to add a dependency to the Eclipse Service on the Tivoli Enterprise Portal Server. Add the dependency in the Cluster Manager, as shown in Figure 53.

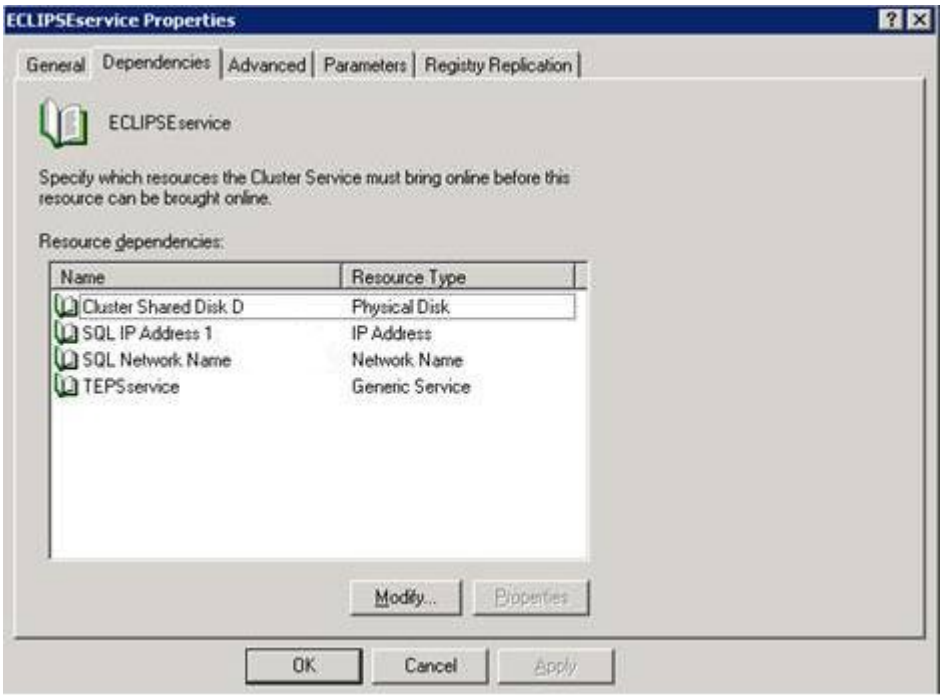

Figure 53. Adding a dependency to the Eclipse Service on the Tivoli Enterprise Portal Server

- If the DB2 resource fails to come online, review the registry entries for "DB2ADMNS" on both the nodes. The group should not be created under the local node names. Modify the value and create the group under the domain groups.

Maintenance-related issues:

- When using the "Modify" option in a Tivoli Monitoring installation to modify a running Tivoli Monitoring environment, such as adding additional agent support to the monitoring server or portal server, the processing might start Tivoli Monitoring services at the end of the change.

- When you are upgrading IBM Tivoli Monitoring, the monitoring server and the portal server might automatically start at the end of the upgrade, even when the startup of these services was set to Manual. After the upgrade, the monitoring server and portal server have to be manually stopped if they were started. Use Cluster Administrator to bring them online so that these clustered resources function in the cluster as expected.

Appendix A. Configuring the cluster creation

To set up Tivoli Monitoring with Tivoli System Automation for Multiplatforms, you will need two cluster nodes with at least one NIC (Network Interface Cards) in each computer. The first step is to prepare both cluster nodes to be Tivoli System Automation for Multiplatforms cluster ready.

All commands must run under root authority.

1. On *both* nodes run the following commands:

   ```
   export CT_MANAGEMENT_SCOPE=2
   preprpnode node1 node2
   ```

   where node1 and node2 are the names that resolve to the NICs, as prepared previously.
2. On one node, edit the **itm6/BuildScripts/clustervariables.sh** file, and supply the values that are described in Table 8 for the variables.

Table 8. Variables for cluster creation

Variable name	Type	Description	Required	Notes	Example
ITM6_POLICIES_DIRECTORY	General	The complete path to the `itm6` directory, as untarred from the `itm6.sam. policies-2.0.tar` file	YES	All nodes must use the same location	/usr/sbin/rsct/ sapolicie s/itm6
CANDLEHOME	General	The Tivoli Monitoring installation directory on the shared disk	YES	All nodes must use the same path	/shareddisk/ITM
HUBNAME	General	The name of the Tivoli Monitoring Hub	YES		HUB_ ITMSYSTEM
CLUSTER_NODE1	General	The name of the first cluster node	YES	This should be the same value used in the `preprpnode` command	NODE1
CLUSTER_NODE2	General	The name of the second cluster node	YES	This should be the same value used in the `preprpnode` command	NODE2
CLUSTER_DOMAIN_NAME	General	The name of the Tivoli System Automation for Multiplatforms domain to be created	YES		ITM

Table 8. Variables for cluster creation (continued)

Variable name	Type	Description	Required	Notes	Example
CLUSTER_RESOURCE_GROUP	General	The name of the Tivoli System Automation for Multiplatforms Resource Group to hold the Tivoli Monitoring resources	YES		ITMRG
DB2_POLICIES_DIRECTORY	DB2 cluster	The complete path to the directory that contains the DB2 Tivoli System Automation for Multiplatforms scripts (db2_start.ksh, db2_stop.ksh, db2_monitor.ksh, etc), as extracted from the db2salinux.tar.gz file	NO – only needed if a DB2 cluster is being used		/usr/sbin/rsct/ sapolicie s/db2
DB2_INSTANCE	DB2 cluster	The DB2 instance in the cluster	See above		DB2INST1
HAFS_MOUNTPOINT	Shared file system	The mount point of the shared disk	YES		/shareddisk
HAFS_DEVICENAME	Shared file system	The device name of the shared disk	YES		/dev/sdd1
HAFS_VFS	Shared file system	The type of shared disk	YES		reiserfs
AIXVG_VGNAMEG	Shared file system on AIX	The name of the volume group containing the shared file system's logical volume	NO – only needed if cluster is running on AIX		
AIXVG_LVNAME	Shared file system on AIX	The name of the logical volume containing the shares file system	NO – only needed if cluster is running on AIX		
HAIP_IPADDRESS	Virtual IP address information	The IP address of the virtual system	YES		9.12.13.14
HAIP_NETMASK	Virtual IP address information	The subnet mask of the virtual system	YES		255.255.252.0

Table 8. Variables for cluster creation (continued)

Variable name	Type	Description	Required	Notes	Example
HAIPNICS_ NODE1ETHER NETNUMBER	Ethernet information for nodes	The Ethernet interface number for node1	YES	Run the `ifconfig` command to determine the Ethernet numbers	eth3 (for Linux)
HAIPNICS_ NODE2ETHER NETNUMBER	Ethernet information for nodes	The Ethernet interface number for node2	YES	Run the `ifconfig` command to determine the Ethernet numbers	en5 (for AIX)
NETTB_ADDRESS	Network tiebreaker	The IP address of the network tiebreakers	NO – only needed if using a network tiebreaker	Must be pingable by both nodes	10.20.20.40
SCSITB_NODE1HOST	SCSI tiebreaker on Linux	The host number of the SCSI tiebreaker device on node1	NO - only needed if using a SCSI tiebreaker on Linux	Use the `dmesg | grep 'Attached scsi disk'` command or `cat /proc/scsi/scsi` command to determine these values	1
SCSITB_NODE1CHAN	SCSI tiebreaker on Linux	The channel number of the SCSI tiebreaker device on node1	See above	See above	0
SCSITB_NODE1ID	SCSI tiebreaker on Linux	The ID number of the SCSI tiebreaker device on node1	See above	See above	0
SCSITB_NODE1LUN	SCSI tiebreaker on Linux	The logical unit number of the SCSI tiebreaker device on node1	See above	See above	0
SCSITB_NODE2HOST	SCSI tiebreaker on Linux	The host number of the SCSI tiebreaker device on node2	See above	See above	1
SCSITB_NODE2CHAN	SCSI tiebreaker on Linux	The channel number of the SCSI tiebreaker device on node2	See above	See above	0
SCSITB_NODE2ID	SCSI tiebreaker on Linux	The ID number of the SCSI tiebreaker device on node2	See above	See above	0

Table 8. Variables for cluster creation (continued)

Variable name	Type	Description	Required	Notes	Example
SCSITB_NODE2LUN	SCSI tiebreaker on Linux	The logical unit number of the SCSI tiebreaker device on node2	See above	See above	0
DISKTB_DISKNAME	Disk tiebreaker on AIX	The disk name of the disk tiebreaker	NO - only needed if using a Disk tiebreaker on AIX		/dev/hdisk0
ECKDTB_DEVICENUM	ECKD tiebreaker on Linux on zSeries	The device number of the ECKD tiebreaker	NO - only needed if using a ECKD tiebreaker on Linux on zSeries	Use the `cat /proc/dasd/devices` command to get this information	099f

3. Change to the `itm6/BuildScripts` directory, and run the following command:
   `./generateclusterfiles.sh`
4. You might want to modify the **StartCommandTimeout** (the amount of time, in seconds, to wait for the resource to start) and **StopCommandTimeout** (the amount of time, in seconds, to wait for the resource to stop) values in the following files, based on how long these actions take on your system. For example, you can double the expected times.

   Hub TEMS: `itm6/DefFiles/TEMSSrv.def`
   Portal server: `itm6/DefFiles/TEPSSrv.def`
   Warehouse Proxy Agent: `itm6/DefFiles/TDWProxy.def`
   Summarization and Pruning Agent: `itm6/DefFiles/SPAgent.def`
5. Copy the contents of the `itm6` directory, and all of its subdirectories, from the node where the `generateclusterfiles.sh` command was run, to the same location on the other node.

   After copying, verify that the execution flag and owner are correct, for the files in `itm6/BuildScripts` and `itm6/ControlScripts`.

Appendix B. Autonomous mode and autonomous agents

This section provides information on both switching from a secondary monitoring server back to the primary hub monitoring server in autonomous mode and achieving high availability when using the autonomous agent.

Autonomous mode ensures event information is not lost when communications are lost between an agent and its monitoring server. Agent autonomy comprises the following two running modes:

Managed mode
Agents come in either of the following two flavors: a fully connected agent to the Tivoli Enterprise Monitoring Server. While in this mode, the agent behaves as agents traditionally do. But, a partially connected agent runs in a semi-autonomous mode; that is, it is disconnected for some time from the monitoring server.

Unmanaged mode
Such agents are fully autonomous and need not be connected to a monitoring server at all.

Achieving High-Availability with the autonomous agent

To achieve high-availability, you must specify two SNMP based event destinations when you are configuring the autonomous agent. If you are using SNMP V1 or V2, the trap might not make it to the destination. However, if you are using SNMP V3, a response will be sent to the agent. When specifying a trap destination with SNMP V3, you can use the retries and timeout parameters to assist with high availability. If a notification cannot be properly sent, it will retry for the number of retries specified in the retries parameter. Each retry is attempted after the timeout value is reached. The default number of retries is 3, and the default timeout value is 2 seconds. You can configure these parameters.

Note: If the autonomous agent loses network connectivity, the traps are lost and will not be received by the agent. If you are using hysteresis mode, the pure event traps won't be sent on the next true interval. In the hysteresis mode, the trap is emitted on the first TRUE encountered, and the clearing trap is emitted when reset. The pure events will only be sent on the next true interval if you are in the default rising continuous mode, whereby an alert is sent on each TRUE evaluation of the situation.

Autonomous mode agent switch from a secondary monitoring server back to the primary hub monitoring server

When an agent running in autonomous mode detects that the primary Tivoli Enterprise Monitoring Server has come back online, it needs to switch from the secondary back to the primary.

Agent configuration parameters

The following agent configuration parameters can control agent to Tivoli Enterprise Monitoring Server connectivity and behavior:

CT_CMSLIST
Required variable that specifies the primary or secondary Tivoli Enterprise Monitoring Server the agent must connect with. Takes a semicolon-delimited list of monitoring servers of the form network *protocol:hostname* or *protocol:address*.

CTIRA_RECONNECT_WAIT
This parameter specifies, in seconds, the frequency with which the agent will attempt to connect with the monitoring server. The default is 600 (10 minutes).

CTIRA_MAX_RECONNECT_TRIES

This parameter is being deprecated. Use it to specify the number of consecutive times without success the agent attempts to connect to a monitoring server before giving up and exiting. The default value of 0 means that the agent will remain started regardless of its connection status with the monitoring server.

Before the release of Tivoli Monitoring V6.2.2, the default value was 720. Along with the CTIRA_RECONNECT_WAIT default setting of 600, the agent tries to connect to the monitoring server for 432000 seconds (5 days) before giving up and exiting. If you prefer to have the agent shut down when the reconnect limit is reached, specify the number of retries. You must also disable the agent watchdog function (disarmWatchdog), which is described in the Agent User's guide.

CTIRA_PRIMARY_FALLBACK_INTERVAL

Forces the agent to switch from the primary Tivoli Enterprise Monitoring Server to one of the defined secondary monitoring servers, because the primary monitoring server is offline or due to network-connectivity issues. It is desirable for the agent to switch back to the primary monitoring server as soon as possible when it becomes available. This parameter controls the frequency with which the agent performs lookup of the primary monitoring server. If the primary monitoring server is found, the agent disconnects from the secondary monitoring server and reconnects to the primary monitoring server. The minimum value must be 2.5 times **CTIRA_RECONNECT_WAIT** value. Default value is 4500 seconds, or 75 minutes.

A value of zero disables this feature. Always set **CTIRA_PRIMARY_FALLBACK_INTERVAL=0** for agents that are directly connected to the hub monitoring server. If Hot Standby is configured, the agent will switch based on the Hot Standby failover operation and not based on the primary fallback feature.

Note: In an hot standby configuration, an agent switches to the acting hub monitoring server after receiving a SWITCH command from the standby hub monitoring server, even if CTIRA_PRIMARY_FALLBACK_INTERVAL=0, because the standby hub monitoring server reroutes agents to the acting hub monitoring server. If the agent requires redundant access to a monitoring server, the minimum configuration is two hub monitoring servers in a Hot Standby configuration and two remote monitoring servers. The agent must be configured to only the remote monitoring servers. That configuration provides the agent two monitoring servers to connect to at any time. If redundancy is not required, then do not configure Hot Standby.

See the *IBM Tivoli Monitoring: Installation and Setup Guide* for a complete list of common agent environment variables and the *IBM Tivoli Monitoring: Administrator's Guide* for more information about autonomous agent behavior.

Switchback processing

When an agent loses its connection with its Tivoli Enterprise Monitoring Server, it switches to a secondary server if one has been defined with the CT_CMSLIST environment variable. The agent must switch back to the primary monitoring server as soon as possible and without restarting.

When an agent running in autonomous mode detects that the primary hub has come back online, it needs to switch from the secondary hub monitoring server back to the primary. The following steps show how the switchback proceeds:

1. The agent detects that it has connected to a secondary monitoring server.
2. The switch back to the primary server occurs when the following conditions are true:
   - There are multiple monitoring servers defined by the CT_CMSLIST parameter.
   - An agent is actively connected to a secondary monitoring server.
   - That agent is not in shutdown mode.
3. The agent starts a background periodic lookup task.

4. The agent calculates the periodic lookup interval from the CTIRA_PRIMARY_FALLBACK_INTERVAL customized value and enforces the minimum value limit.
5. When a periodic lookup interval expires, the lookup task performs Local Location Broker lookup of the primary server. This periodic lookup continues until the primary Tivoli Enterprise Monitoring Server is found or the agent shuts down.
6. On a successful lookup of the primary monitoring server, the agent initiates server-switch processing:
   a. Notifies the acting monitoring server that the agent is now disconnecting.
   b. Performs the agent shutdown procedure as if it has lost monitoring server connectivity.
   c. Sets the primary monitoring server as the preferred switch-to target.
   d. Begins the server connection procedure.
   e. Exits periodic lookup unconditionally: If the reconnection is successful, there is no need for the periodic lookup task. Otherwise, the standard agent reconnection procedure takes effect.

Appendix C. Predefined scripts

The following scripts are located in the `itm6/BuildScripts` directory, in the `itm6.sam.policies-2.0.tar` file.

The `mkitmcluster.sh` script calls the scripts listed in Table 9. The `mkitmcluster.sh` script creates the following resources:

- Cluster domain
- Resource group
- Tiebreaker
- Shared file system and virtual IP resources
- DB2 resource (optional)

```
mkitmcluster.sh net|scsi|disk|eckd [db2]
```

Where the first parameter is the tiebreaker type. If the second parameter is specified, then the DB2 resources are also created.

In certain cases, you might already have these resources defined, or you might want to define them separately. Table 9 includes a description of each script. If you need to define any of the resources differently, you can create your own scripts based on the provided ones.

The scripts depend on the variables defined in `itm6/BuildScripts/clustervariables.sh` , so be sure to edit this file, and run `itm6/BuildScripts/generateclusterfiles.sh`. If a command-line parameter is specified, it overrides the variable's value from `itm6/BuildScripts/clustervariables.sh`.

After running the scripts to create the resources, start the resource group with the following commands:

```
export CT_MANGEMENT_SCOPE=2
chrg -o online rg_name
```

Table 9. Predefined scripts

Script	Command line	Prereqs	Variables used

mkitmdomain.sh
 	
Creates the cluster domain
	`mkitmdomain.sh node1 node2 domain_name`
	`mkitmddomain.sh node1 node2`
	`mkitmdomain.sh node1`
	`mkitmddomain.sh`
		`preprpnode` must be run on both nodes before running this command	$CLUSTER_NODE1
			$CLUSTER_NODE2
			$CLUSTER _DOMAIN _ NAME

mkitmrg.sh
 	
Creates the cluster resource group
	`mkitmrg.sh rg_name domain_name`
	`mkitmrg.sh rg_name`
	`mkitmrg.sh`
		The domain must already be created and started before running this command	$CLUSTER _RESOURCE _ GROUP
			$CLUSTER _DOMAIN _ NAME

Table 9. Predefined scripts (continued)

Script	Command line	Prereqs	Variables used

mkitmtb.sh
 	
Creates the cluster tiebreaker, and sets it as the active tiebreaker
	`mkitmtb.sh net|scsi|disk|eckd`
	`domain_name`
	`mkitmtb.sh net|scsi|disk|eckd`
		The domain must already be created and started before running this command	$CLUSTER_DOMAIN_ NAME

mkitmbasicresources.sh
 	
Creates the file system and virtual IP resources, and adds them to the resource group
	`mkitmbasicresources.sh rg_name`
	`domain_name`
	`mkitmbasicresources.sh rg_name`
	`mkitmbasicresources.sh`
		The domain must already be created and started, and the resource group must be created, before running this command	$CLUSTER_RESOURCE_ GROUP
			
			$CLUSTER_DOMAIN_ NAME

mkitmdb2resources.sh
 	
Creates the DB2 resource, and adds it to the resource group
	`mkitmdb2resources.sh rg_name domain_name`
	`mkitmdb2resources.sh rg_name`
	`mkitmdb2resources.sh`
		The domain must already be created and started, and the resource group must be created, before running this command	

When the resources have been created by the scripts, or by some other means, you must then install the Tivoli Monitoring components, and add their resources to the resource group. Use the commands that are described in Table 10 to add the Tivoli Monitoring resources.

Again, you might want to modify them to fit your environment.

Before running these scripts, take the resource group offline with the following command:

```
chrg -o offline rg_name
```

After running the scripts to create the resources, start the resource group with the following command:

```
chrg -o online rg_name
```

Table 10. Commands

Script	Command line	Prerequisites	Variables Used
`temsrscbuild.sh` Creates the Tivoli Monitoring hub resource, and adds it to the resource group	`temsrscbuild.sh`	The domain must be created and started, and the resource group must be created and must be offline. The ITMIP resource must exist, which is created in the `mkitmbasicresources.sh` script.	$CLUSTER_RESOURCE_ GROUP

Table 10. Commands (continued)

Script	Command line	Prerequisites	Variables Used
`tepsrscbuild.sh` Creates the Tivoli Monitoring portal server resource, and adds it to the resource group	`tepsrscbuild.sh`	The domain must be created and started, and the resource group must be created and must be offline. The ITMDB2 resource must exist, which is created in the `mkitmdb2resources.sh` script.	$CLUSTER_RESOURCE_ GROUP
`spagentrscbuild.sh` Creates the Tivoli Monitoring Summarization and Pruning Agent resource, and adds it to the resource group	`spagentrscbuild.sh`	The domain must be created and started, and the resource group must be created and must be offline. The ITMDB2 resource must exist, which is created in the `mkitmdb2resources.sh` script.	$CLUSTER_RESOURCE_ GROUP
`tdwproxyrscbuild.sh` Creates the Tivoli Monitoring Tivoli Data Warehouse Proxy Agent resource, and adds it to the resource group	`tdwproxyrscbuild.sh`	The domain must be created and started, and the resource group must be created and must be offline. The ITMDB2 resource must exist, which is created in the `mkitmdb2resources.sh` script.	$CLUSTER_RESOURCE_ GROUP
`allrscbuild.sh` Calls the above four scripts to create all of the resources, and adds them to the resource group	`allrscbuild.sh`	See above descriptions.	See above descriptions.

Appendix D. EIF Information

The following sections describe procedures to assist in creating a high-availability setup by using the EIF (Event Integration Facility).

EIF event synchronization with hot standby

To set up event synchronization from the Tivoli Enterprise Console or OMNIbus for a hub monitoring server with hot standby, you must configure the event synchronization to send updates to the secondary monitoring server equipped with hot standby. This setup can be done during the initial installation of event synchronization, or by entering a command at the command-line interface after installation.

For event synchronization with Netcool/OMNIbus, enter the `sitconfuser.cmd` (for Windows systems) or `sitconfuser.sh` (for UNIX systems) command in the command line interface. The following example displays the command to add the itm17 monitoring server on a UNIX system:

```
sitconfuser.sh add serverid=itm17.ibm.com userid=admin password=acc3ssing
pathc=/opt/IBM/SitForwarder/etc type=OMNIBUS
```

For more information about the `sitconfuser.sh` command, see Chapter 4 of the *IBM Tivoli Monitoring Command Reference*.

For event synchronization with the Tivoli Enterprise Console, enter the `sitconfsvruser.sh` command in the command line interface. The following example displays the command to add the itm17 monitoring server:

```
sitconfsvruser.sh add serverid=itm17.ibm.com userid=admin password=acc3ssing
```

For more information about the `sitconfsvruser.sh` command, see Chapter 3 of the *IBM Tivoli Monitoring Command Reference*.

EIF event synchronization with a clustered monitoring server

There should not be a need for configuration changes in the EIF Event Synchronization setup from the Tivoli Enterprise Console and OMNIbus if you have a clustered monitoring server.

In a Hot Standby environment, the user ID and password for SUF (Situation Update Forwarder) must be defined to both the acting hub and the standby hub.

EIF event forwarding with multiple event destinations

A user can set up multiple event destinations from the command line interface, and can forward events to multiple event destinations by using the `tacmd createeventdest` command to designate additional default event destinations.

The event destination specified during the initial installation and configuration is automatically designated as the default event destination and given `id=0`, so a different number must be chosen for the `id` property of any subsequent default event destinations. After a user has specified multiple event destinations, any events that do not have designated event destinations specified (using the `TDST=` parameter in the `SITINFO` column of the situation definition) will be forwarded to all designated default event destinations.

The following example displays the code used to define an additional default event destination with an ID of "2".

```
tacmd createeventdest -i  2 -p NAME=DEST2 HOST1=dest2.ibm.com:5529  DEFAULT=Y
```

For more information on the `tacmd createeventdest` command, see to the *IBM Tivoli Monitoring Command Reference* at http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.2/itm_cmdref.pdf.

Scenarios for setting up multiple EIF destinations

If you are managing a cluster environment, see "EIF event forwarding with multiple event destinations" on page 149 to set up multiple event destinations using the `tacmd createeventdest` command. For a Hot Standby environment, the event destination table is not currently automatically replicated, so the `tacmd createeventdest` command must be used to define the same event destinations for both the acting hub monitoring server and the standby hub monitoring server.

Documentation library

Various publications are relevant to the use of IBM Tivoli Monitoring and to the commonly shared components of Tivoli Management Services.

These publications are listed in the following categories:

- IBM Tivoli Monitoring library
- Related publications

Documentation is delivered in the IBM Tivoli Monitoring and OMEGAMON® XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp and also in the **Files** section of the Application Performance Management community.

For information about accessing and using the publications, select IBM Tivoli Monitoring → **Using the publications** in the **Contents** pane of the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp.

To find a list of new and changed publications, click the **New in this release** topic on the IBM Tivoli Monitoring welcome page. To find publications from the previous version of a product, click **Previous versions** under the name of the product in the **Contents** pane.

IBM Tivoli Monitoring library

The IBM Tivoli Monitoring library provides information about the commonly shared components of Tivoli Management Services.

- *Quick Start Guide*

  Introduces the components of IBM Tivoli Monitoring.
- *Installation and Setup Guide*, SC22-5445

  Provides instructions for installing and configuring IBM Tivoli Monitoring components on Windows, Linux, and UNIX systems.
- *Installation Roadmap* available on Service Management Connect

  Provides a roadmap that covers the installation of IBM Tivoli Monitoring.
- *High Availability Guide for Distributed Systems*, SC22-5455

  Gives instructions for several methods of ensuring the availability of the IBM Tivoli Monitoring components.
- *Program Directory for IBM Tivoli Management Services on z/OS*, GI11-4105

  Gives instructions for the SMP/E installation of the Tivoli Management Services components on z/OS.
- *Administrator's Guide*, SC22-5446

  Describes the support tasks and functions required for the Tivoli Enterprise Portal Server and clients, including Tivoli Enterprise Portal user administration.
- *Command Reference* available on Service Management Connect

  Provides detailed syntax and parameter information, as well as examples, for the commands you can use in IBM Tivoli Monitoring.
- *Messages* available on Service Management Connect

  Lists and explains messages generated by all IBM Tivoli Monitoring components and by z/OS-based Tivoli Management Services components (such as Tivoli Enterprise Monitoring Server on z/OS and TMS:Engine).
- *Troubleshooting Guide* available on Service Management Connect

Provides information to help you troubleshoot problems with the software.

- *Tivoli Enterprise Portal User's Guide* available on Service Management Connect

  Complements the Tivoli Enterprise Portal online help. The guide provides hands-on lessons and detailed instructions for all Tivoli Enterprise Portal features.

- Tivoli Enterprise Portal online help

  Provides context-sensitive reference information about all features and customization options of the Tivoli Enterprise Portal. Also gives instructions for using and administering the Tivoli Enterprise Portal.

Documentation for the base agents

If you purchased IBM Tivoli Monitoring as a product, you received a set of base monitoring agents as part of the product. If you purchased a monitoring agent product (for example, an OMEGAMON XE product) that includes the commonly shared components of Tivoli Management Services, you did not receive the base agents.

The following publications provide information about using the base agents.

- Agentless operating system monitors
  - *Agentless Monitoring for Windows Operating Systems User's Guide*, SC23-9765
  - *Agentless Monitoring for AIX Operating Systems User's Guide*, SC23-9761
  - *Agentless Monitoring for HP-UX Operating Systems User's Guide*, SC23-9763
  - *Agentless Monitoring for Solaris Operating Systems User's Guide*, SC23-9764
  - *Agentless Monitoring for Linux Operating Systems User's Guide*, SC23-9762
- OS agent documentation is delivered in the following locations:

  Agent Installation and Configuration Guide
  Available in the Information Center:
  - *IBM i OS Agent Installation and Configuration Guide*, SC27-5653
  - *Linux OS Agent Installation and Configuration Guide*, SC27-5652
  - *UNIX OS Agent Installation and Configuration Guide*, SC27-5651
  - *Windows OS Agent Installation and Configuration Guide*, SC27-5650

  Agent Reference
  Available on Service Management Connect

  Agent Troubleshooting Guide
  Available on Service Management Connect

  Infrastructure Management Dashboards for Servers Reference
  Available on Service Management Connect
- Warehouse agent documentation is delivered in the following locations:

  Agent Installation and Configuration Guide
  Available in the Information Center:
  - *Warehouse Proxy Agent Installation and Configuration Guide*, SC27-5655
  - *Warehouse Summarization and Pruning Agent Installation and Configuration Guide*, SC27-5654

  Agent Reference
  Available on Service Management Connect

  Agent Troubleshooting Guide
  Available on Service Management Connect
- System P agents
  - *AIX Premium Agent User's Guide*, SA23-2237
  - *CEC Base Agent User's Guide*, SC23-5239

  - *HMC Base Agent User's Guide*, SA23-2239
  - *VIOS Premium Agent User's Guide*, SA23-2238
- Other base agents
  - *Agent Builder User's Guide*, SC32-1921
  - *Performance Analyzer User's Guide*, SC27-4004
  - *Systems Director base Agent User's Guide*, SC27-2872
  - *Tivoli Log File Agent User's Guide*, SC14-7484
  - *Tivoli zEnterprise Monitoring Agent User's Guide*, SC14-7359 and the *Tivoli zEnterprise Monitoring Agent Installation and Configuration Guide*, SC14-7358

Related publications

For information about related products and publications select **OMEGAMON XE shared publications** or other entries in the **Contents** pane of the IBM Tivoli Monitoring and OMEGAMON XE Information Center.

You can access the IBM Tivoli Monitoring and OMEGAMON XE Information Center at http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/index.jsp .

You can also access other information centers at IBM Tivoli Documentation Central (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central).

Tivoli Monitoring community on Service Management Connect

Connect, learn, and share with Service Management professionals: product support technical experts who provide their perspectives and expertise.

For information about Tivoli products, see the Application Performance Management community on SMC at IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm).

For introductory information, see IBM Service Management Connect (http://www.ibm.com/developerworks/servicemanagement).

Use Service Management Connect in the following ways:

- Become involved with transparent development, an ongoing, open engagement between other users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and the (enter your community name here) community.
- Read blogs to benefit from the expertise and experience of others.
- Use wikis and forums to collaborate with the broader user community.

Other sources of documentation

You can obtain additional technical documentation about monitoring products from other sources.

- Tivoli wikis

  IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm) includes a list of relevant Tivoli wikis that offer best practices and scenarios for using Tivoli products, white papers contributed by IBM employees, and content created by customers and business partners.

Two of these wikis are of particular relevance to IBM Tivoli Monitoring:

- The IBM Tivoli Monitoring Wiki (https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Monitoring) provides information about IBM Tivoli Monitoring and related distributed products, including IBM Tivoli Composite Application Management products.
- The Tivoli System z® Monitoring and Application Management Wiki provides information about the OMEGAMON XE products, NetView® for z/OS, Tivoli Monitoring Agent for z/TPF, and other System z monitoring and application management products.

- IBM Integrated Service Management Library

  http://www.ibm.com/software/brandcatalog/ismlibrary/

  IBM Integrated Service Management Library is an online catalog that contains integration documentation and other downloadable product extensions.
- Redbooks®

  http://www.redbooks.ibm.com/

  IBM Redbooks and Redpapers include information about products from platform and solution perspectives.
- Technotes

  Technotes provide the latest information about known product limitations and workarounds. You can find Technotes through the IBM Software Support Web site at http://www.ibm.com/software/support/.

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides ways for you to obtain the support you need.

Online

The following sites contain troubleshooting information:

- Go to the IBM Support Portal (http://www.ibm.com/support/entry/portal/software) and follow the instructions.
- Go to IBM Service Management Connect > Application Performance Management (http://www.ibm.com/developerworks/servicemanagement/apm) and select the appropriate wiki.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to IBM Support Assistant (http://www-01.ibm.com/software/support/isa).

Troubleshooting Guide

For more information about resolving problems, see the product's Troubleshooting Guide.

Using IBM Support Assistant

The IBM Support Assistant is a free, stand-alone application that you can install on any workstation. You can then enhance the application by installing product-specific plug-in modules for the IBM products you use.

The IBM Support Assistant saves you the time it takes to search the product, support, and educational resources. The IBM Support Assistant helps you gather support information when you need to open a problem management record (PMR), which you can then use to track the problem.

The product-specific plug-in modules provide you with the following resources:

- Support links
- Education links
- Ability to submit problem management reports

For more information, and to download the IBM Support Assistant, see http://www.ibm.com/software/support/isa. After you download and install the IBM Support Assistant, follow these steps to install the plug-in for your Tivoli product:

1. Start the IBM Support Assistant application.
2. Select **Updater** on the Welcome page.
3. Select **New Properties and Tools** or select the **New Plug-ins** tab (depending on the version of IBM Support Assistant installed).
4. Under **Tivoli**, select your product, and then click **Install**. Be sure to read the license and description.

   If your product is not included on the list under **Tivoli**, no plug-in is available yet for the product.
5. Read the license and description, and click **I agree**.
6. Restart the IBM Support Assistant.

Obtaining fixes

A product fix might be available to resolve your problem. To determine which fixes are available for your Tivoli software product, follow these steps:

1. Go to the IBM Software Support website at http://www.ibm.com/software/support.
2. Under **Select a brand and/or product**, select **Tivoli**.

   If you click **Go**, the **Search within all of Tivoli support** section is displayed. If you don't click **Go**, you see the **Select a product** section.
3. Select your product and click **Go**.
4. Under **Download**, click the name of a fix to read its description and, optionally, to download it.

   If there is no **Download** heading for your product, supply a search term, error code, or APAR number in the field provided under **Search Support (this product)**, and click **Search**.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support website at http://www.ibm.com/software/support.
2. Click **My support** in the far upper-right corner of the page under **Personalized support**.
3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. The **Edit profile** tab is displayed.
5. In the first list under **Products**, select **Software**. In the second list, select a product category (for example, **Systems and Asset Management**). In the third list, select a product sub-category (for example, **Application Performance & Availability** or **Systems Performance**). A list of applicable products is displayed.
6. Select the products for which you want to receive updates.
7. Click **Add products**.
8. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
9. In the **Documents** list, select **Software**.
10. Select **Please send these documents by weekly email**.
11. Update your e-mail address as needed.
12. Select the types of documents you want to receive.
13. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

Online
: Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

By phone
: Call 1-800-IBM-4You (1-800-426-4968).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects. The easiest way to obtain that assistance is to open a PMR or ETR directly from the IBM Support Assistant.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online

  Go to the Passport Advantage website at http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm .

  By telephone

  For the telephone number to call in your country, go to the IBM Software Support website at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request website at https://techsupport.services.ibm.com/ssr/login.
- For customers with Linux, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line website at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage website at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the web at http://www14.software.ibm.com/webapp/set2/sas/f/handbook/home.html and click the name of your geographic region for telephone numbers of people who provide support for your location.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: `© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2013. All rights reserved.`

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Printed in USA

SC22-5455-01