

IBM Workload Scheduler



# High Availability Cluster Environments

*Version 9 Release 4*



IBM Workload Scheduler



# High Availability Cluster Environments

*Version 9 Release 4*

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 57.

---

# Contents

**Figures . . . . . v**

**Tables . . . . . vii**

**About this publication . . . . . ix**

What is new in this release . . . . . ix

Who should read this publication . . . . . ix

Accessibility . . . . . ix

Technical training . . . . . ix

Support information. . . . . x

**Chapter 1. Types of high availability . . . 1**

Agent availability versus job availability . . . . . 1

HACMP for AIX scenario - Backup domain manager . . . . . 1

**Chapter 2. Tivoli System Automation for Multiplatforms high availability . . . . . 3**

**Chapter 3. IBM Workload Scheduler with Windows cluster . . . . . 5**

Windows cluster overview. . . . . 5

IBM Workload Scheduler with Microsoft Windows cluster environments. . . . . 5

Prerequisite knowledge. . . . . 5

Design limitations . . . . . 5

Supported operating systems. . . . . 6

Compatibility, upgrade, and coexistence . . . . . 6

Security and Authentication . . . . . 7

Enabling IBM Workload Scheduler to work in a

Windows Cluster environment . . . . . 7

IBM Workload Scheduler Windows Cluster

Enabler components. . . . . 7

Installation and configuration. . . . . 9

Upgrading agents to a new version and available fix packs . . . . . 14

twClusterAdm command with examples of usage . . . . . 16

Operating IBM Workload Scheduler in Windows cluster environment . . . . . 29

IBM Workload Scheduler Cluster Administrator extension . . . . . 31

Uninstalling IBM Workload Scheduler . . . . . 32

Troubleshooting . . . . . 33

Traces . . . . . 33

Error 1314 taking online the resource and the Workstation does not link . . . . . 34

IBM Workload Scheduler resource instance reports fail status or IBM Workload Scheduler

user jobs go in the abend state . . . . . 34

Windows Report panel with Jobmon.exe. . . . . 35

Cluster: IP validation error on Netman stdlist . . . . . 35

**Chapter 4. IBM Workload Scheduler with HACMP . . . . . 37**

High-Availability Cluster Multi-Processing . . . . . 37

Benefits. . . . . 37

Physical components of an HACMP cluster. . . . . 38

UNIX cluster overview . . . . . 40

Prerequisite knowledge . . . . . 41

Standby and takeover configurations . . . . . 41

Design limitations . . . . . 44

Supported configurations. . . . . 44

Upgrading from previous versions of the product . . . . . 48

**Appendix. Resolving desktop heap size problems on workstations with more than three agents . . . . . 51**

Problem description . . . . . 51

Solutions . . . . . 52

Modify the shared heap buffer sizes . . . . . 52

Configure the IBM Workload Scheduler Windows service to start as a local system account . . . . . 53

Customize the desktop name so that it is reused . . . . . 53

Implementing the solutions . . . . . 53

Modify configuration of Windows service . . . . . 54

Modify the Windows registry entries that determine the heap size . . . . . 54

Modify localopts to supply a shared desktop name . . . . . 54

**Notices . . . . . 57**

Trademarks . . . . . 59

Terms and conditions for product documentation. . . . . 59

**Index . . . . . 61**



---

## Figures

1. Main components of the IBM Workload Scheduler Cluster Enabler . . . . .	8	6. New Properties Parameters tab . . . . .	32
2. Installing in a cluster . . . . .	10	7. Shared disk with mirror . . . . .	39
3. Clusters in an IBM Workload Scheduler network. . . . .	10	8. Active-Passive configuration in normal operation . . . . .	42
4. File Server example on Windows Server 2008	14	9. Failover on Active-Passive configuration	42
5. Resource Dependencies tab (Windows Server 2008). . . . .	29	10. Logical file system volumes . . . . .	43
		11. Failover scenario. . . . .	43





---

## Tables

1. Monitoring options in your cluster environment. . . . . 20



---

## About this publication

This publication describes how Windows, UNIX, Linux, and HACMP for AIX clusters fit into the topology of IBM Workload Scheduler. This publication also describes the enhancements to IBM Workload Scheduler to support the clustering and high-availability environment based on Microsoft Windows.

---

## What is new in this release

Learn what is new in this release.

For information about the new or changed functions in this release, see *IBM Workload Automation: Overview*, section *Summary of enhancements*.

For information about the APARs that this release addresses, see the IBM Workload Scheduler Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27048863> and the Dynamic Workload Console Release Notes at <http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27048864>.

=  
=  
=

New or changed content is marked with revision bars. For the PDF format, new or changed V9.4 content is marked in the left margin with a pipe (|) character and new or changed V9.4FP1 content is marked with an equal sign (=).

---

## Who should read this publication

Learn the audience of this publication.

This publication is intended for the following audience:

- Users that run IBM Workload Scheduler in a Windows cluster environment.
- Users that run IBM Workload Scheduler in a UNIX and Linux cluster environment.
- IT Administrators or IBM Workload Scheduler IT administrators who plan the layout of the IBM Workload Scheduler network.
- Users that install the various software packages on the computers that make up the IBM Workload Scheduler network.

---

## Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully.

With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For full information, see the Accessibility Appendix in the *IBM Workload Scheduler User's Guide and Reference*.

---

## Technical training

Cloud & Smarter Infrastructure provides technical training.

For Cloud & Smarter Infrastructure technical training information, see:  
<http://www.ibm.com/software/tivoli/education>

---

## Support information

IBM provides several ways for you to obtain support when you encounter a problem.

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see the appendix about support information in *IBM Workload Scheduler: Troubleshooting Guide*.

---

## Chapter 1. Types of high availability

Server clusters are designed to keep resources (such as applications, disks, and file shares) available. Availability is a measure of the ability of clients to connect with and use a resource. If a resource is not available, clients cannot use it.

It is possible to contrast high-availability with fault-tolerance, as different benchmarks for measuring availability:

### **Fault-tolerance**

Fault-tolerance is defined as 100% availability all of the time. Fault-tolerant systems are designed to guarantee resource availability.

### **High-availability**

A high-availability system maximizes resource availability. A highly available resource is available a high percentage of the time that might approach 100% availability, but a small percentage of down time is acceptable and expected.

In this way, high-availability can be defined as a highly available resource that is almost always operational and accessible to clients.

The section explains the following type of high availability: “HACMP for AIX scenario - Backup domain manager”

---

## Agent availability versus job availability

Having IBM Workload Scheduler working in a Windows cluster and HACMP for AIX environments does not mean that the jobs the scheduler launches are automatically aware of the cluster. It is not the responsibility of IBM Workload Scheduler to roll back any actions that a job might have performed during the time it was running. It is the responsibility of the user creating the script or command to allow for a roll back or recovery action in case of failover.

For a failover, the IBM Workload Scheduler agent reports any job running at that moment in the **ABEND** state with return code RC=0. This prevents any further dependencies being released. Only a recovery (or rerun) of the failing jobs is possible.

In general, IBM Workload Scheduler does not manage job and job stream interruption. Extra logic needs to be added by the user to recover job and job stream interruptions (see sections 1.4.2 and 2.3.4 of the Redbook *High Availability Scenarios with IBM® Workload Scheduler and IBM Tivoli® Framework*).

---

## HACMP for AIX scenario - Backup domain manager

IBM Workload Scheduler provides a degree of high-availability through its backup domain manager feature, which can also be implemented as a backup master domain manager.

The backup domain manager duplicates changes to the production plan of the domain manager. When a failure is detected, the **switchmgr** command is issued to all workstations in the domain of the domain manager server, causing the workstations to recognize the backup domain manager.

However there are cases where a cluster environment represents a suitable alternative:

- Difficulty in implementing the automatic domain responsibility switch
- Difficulty in switching jobs that should run on the domain manager to the backup domain manager
- The need to notify the switch of a domain manager to the IBM Workload Scheduler network
- A high-availability product addresses many of the coding issues that surround detecting hardware failures
- Implementing high-availability for fault-tolerant agents cannot be accomplished using the backup domain manager feature

---

## Chapter 2. Tivoli System Automation for Multiplatforms high availability

Integration of IBM Workload Scheduler and Tivoli System Automation for Multiplatforms to implement a highly available scheduling environment.

IBM Workload Scheduler consists of multiple built-in high-availability features that you use to switch components to other workstations in the IBM Workload Scheduler network. When combined with Tivoli System Automation for Multiplatforms, the recovery operations:

- Are regulated by a third-party, which has more insight into the state of all critical components in the IBM Workload Scheduler network.
- Are issued in an automatic fashion, while maintaining dependencies and relationships between all components.
- Require less time, compared with a manual failover of all components.

For more information about the requirements and configuration steps to implement this integration, see *Integrating with Tivoli System Automation for Multiplatforms*.





---

## Chapter 3. IBM Workload Scheduler with Windows cluster

This section contains information on the following topics:

- “Windows cluster overview”
- “Enabling IBM Workload Scheduler to work in a Windows Cluster environment” on page 7
- “Troubleshooting” on page 33

---

### Windows cluster overview

This section describes how Windows clusters fit into the topology of IBM Workload Scheduler. It is divided into the following subsections:

- “IBM Workload Scheduler with Microsoft Windows cluster environments”
- “Prerequisite knowledge”
- “Design limitations”
- “Supported operating systems” on page 6
- “Compatibility, upgrade, and coexistence” on page 6
- “Security and Authentication” on page 7

### IBM Workload Scheduler with Microsoft Windows cluster environments

IBM Workload Scheduler can be integrated into the Windows cluster environments using Microsoft generic cluster resources. This document describes how this is achieved.

To help you perform this integration, the product provides:

- A utility that remotely configures IBM Workload Scheduler on all the nodes of the cluster without reinstalling IBM Workload Scheduler on each node. The utility implements the logic to define and install the IBM Workload Scheduler custom resource within a cluster group.
- A new custom resource DLL specifically for IBM Workload Scheduler.

### Prerequisite knowledge

To understand this document, you must be knowledgeable about IBM Workload Scheduler and Microsoft Windows clusters:

#### IBM Workload Scheduler

For an overview of IBM Workload Scheduler, see *Planning and Installation Guide*.

#### Microsoft Windows clusters

For a Quick Start Guide for Server Clusters, and information about Windows Clustering Services, see the Microsoft Windows Server TechNet website.

### Design limitations

The following design limitations apply:

- “The master domain manager” on page 6
- “IBM Workload Scheduler commands” on page 6

- “Use with multiple agents”

### **The master domain manager**

The IBM Workload Scheduler master domain manager is not supported as a cluster resource for the following reasons:

- The master domain manager runs the **JnextPlan** critical job stream. The responsibility of the job stream is to create a new plan for the current production day. This process cannot be interrupted. An interruption might cause malfunctions and scheduling service interruptions. Only manual steps can be used to recover from such malfunctions or service interruptions. Because failover of the cluster group that contains the IBM Workload Scheduler resource stops the agent on the current node and starts it on a different node, if failover happens during running of **JnextPlan** it could be destructive.
- The IBM Workload Scheduler command-line utilities (conman, composer, and so on) are not aware of the cluster and if they are interrupted (through a failover of the cluster group that contains the IBM Workload Scheduler resource) they might corrupt some vital information for IBM Workload Scheduler.

### **IBM Workload Scheduler commands**

Any IBM Workload Scheduler command that is running during a failover is not automatically taken offline (unlike the main processes netman, mailman, batchman, and jobman) by the IBM Workload Scheduler cluster resource.

This could be particularly problematical if the failover happens during an ad-hoc submission. The job submitted could remain in the **ADDING** state forever.

### **Use with multiple agents**

If you plan to use multiple agents on the same Windows server, you must take steps to reconfigure the Windows desktop heap memory so that the multiple agents processes share more desktop heap memory. These steps are described in “Resolving desktop heap size problems on workstations with more than three agents,” on page 51.

## **Supported operating systems**

The IBM Workload Scheduler Windows Cluster Enabler is available for 64-bit Windows systems.

To obtain the latest information about supported Windows versions, see Windows supported versions.

## **Compatibility, upgrade, and coexistence**

The IBM Workload Scheduler agent configured to work in a cluster environment does not impact compatibility with previous IBM Workload Scheduler versions and does not require configuration or data migration.

An IBM Workload Scheduler agent configured to work in a Windows cluster environment can be connected to both the distributed and the end-to-end network configurations.

The DLL that extends the Windows Cluster Administration program is sometimes updated in fix packs and new releases of IBM Workload Scheduler. For this reason, the program that installs the Windows Cluster Enabler has an update option that you use to update the DLL with a new version, minor (fix pack) or major (new release of IBM Workload Scheduler).

## Security and Authentication

The usual IBM Workload Scheduler security authentication and authorization mechanism applies.

---

## Enabling IBM Workload Scheduler to work in a Windows Cluster environment

### About this task

This section describes the implementation of the Windows Cluster Enabler. It consists of the following subsections:

- “IBM Workload Scheduler Windows Cluster Enabler components”
- “Installation and configuration” on page 9
- “Upgrading cluster nodes manually” on page 15
- “twscClusterAdm command with examples of usage” on page 16
- “Operating IBM Workload Scheduler in Windows cluster environment” on page 29
- “Uninstalling IBM Workload Scheduler” on page 32

## IBM Workload Scheduler Windows Cluster Enabler components

### About this task

The IBM Workload Scheduler Windows Cluster Enabler consists of the following elements:

- A utility to:
  - Install and remotely configure IBM Workload Scheduler on all the other nodes of the cluster
  - Install and configure the IBM Workload Scheduler cluster resource type for a given virtual server
- An IBM Workload Scheduler Manager Custom Resource type to manage cluster events for IBM Workload Scheduler instances (new DLLs)
- An IBM Workload Scheduler extension DLL to extend the Windows Cluster Administration program

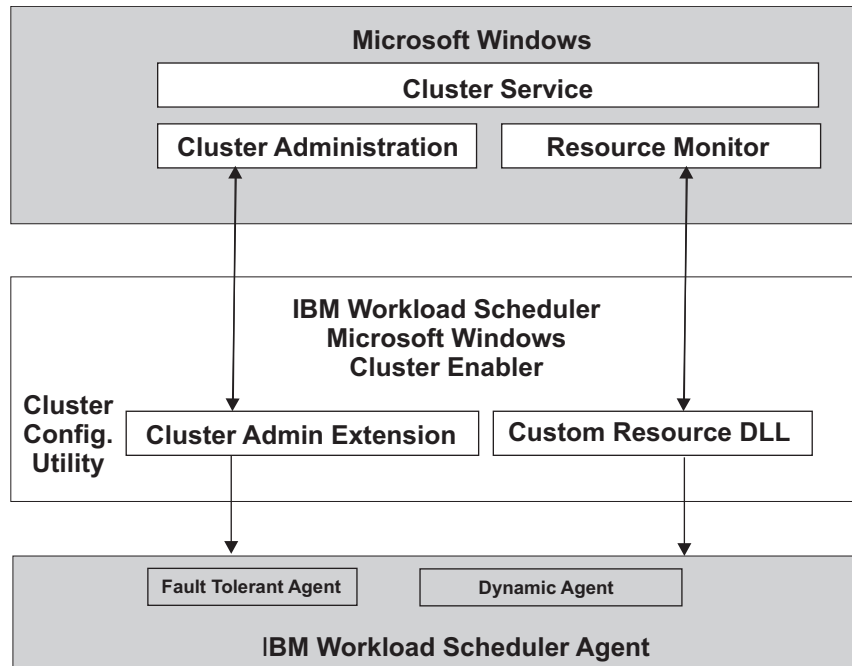


Figure 1. Main components of the IBM Workload Scheduler Cluster Enabler

The main component is the custom resource DLL. It has the following characteristics:

- It can be brought online and taken offline
- It can be managed in a cluster
- It can be hosted (owned) by only one node at a time

As shown in Figure 1, the Cluster service communicates with the custom resource DLL through the resource monitor to manage resources. In response to a Cluster service request, the resource monitor calls the appropriate entry-point function in the custom resource DLL to check and control the resource state (possibly the IBM Workload Scheduler agent).

The custom resource DLL either performs the operation, signals the resource monitor to apply default processing (if any), or both. The custom resource DLL is responsible for providing entry-point implementations that serve the needs of the IBM Workload Scheduler resources.

The IBM Workload Scheduler Manager custom resource DLL provides the following entry-points (or services):

**IsAlive**

Determines if the IBM Workload Scheduler agent is currently active.

**Offline**

Performs a graceful shutdown of the IBM Workload Scheduler agent.

**Online**

Starts the IBM Workload Scheduler agent, links the agent to the network, and makes the resource available to the cluster.

**Terminate**

Performs an immediate shutdown of the resource.

The IBM Workload Scheduler Manager custom resource DLL is a bridge between the resource monitor (part of the Windows cluster service) and the IBM Workload Scheduler agent. The most important objective of the custom resource DLL is to understand the agent state and to bring it online or offline using the correct sequence of commands.

## Installation and configuration

### About this task

This section describes the installation and configuration of the Windows Cluster Enabler. It is divided into the following subsections:

- “Windows Cluster Enabler”
- “Installing in a cluster”
- “Prerequisites” on page 11
- “Install and configure a new IBM Workload Scheduler agent” on page 12

The IBM Workload Scheduler Windows Cluster Enabler is installed automatically when you install IBM Workload Scheduler. A new folder, named `cluster`, is created within the IBM Workload Scheduler installation directory.

### Windows Cluster Enabler

#### About this task

To enable the IBM Workload Scheduler to work in a windows cluster environment, the installation process provides the following files:

#### **ITWSWorkstationEx.dll**

The IBM Workload Scheduler Cluster Administrator extension. It adds a new property sheet and wizard pages for the IBM Workload Scheduler resource type to the Cluster Administrator console. See “IBM Workload Scheduler Cluster Administrator extension” on page 31 for more details.

#### **twscClusterAdm.exe**

Used to install and configure IBM Workload Scheduler.

#### **ITWSResources.dll**

The dynamic-link library containing the implementation of the Resource API for the IBM Workload Scheduler **ITWSWorkstation** resource type. It implements the logic that enables the **Resource Monitor** to monitor and manage the IBM Workload Scheduler agent

#### **ITWSExInst.cmd**

The sample script that registers the IBM Workload Scheduler Cluster Administrator extension.

### Installing in a cluster

#### About this task

A minimal cluster configuration is composed of two nodes. Complying with the used disk technology, a Windows cluster can have from 2 to 36 nodes.

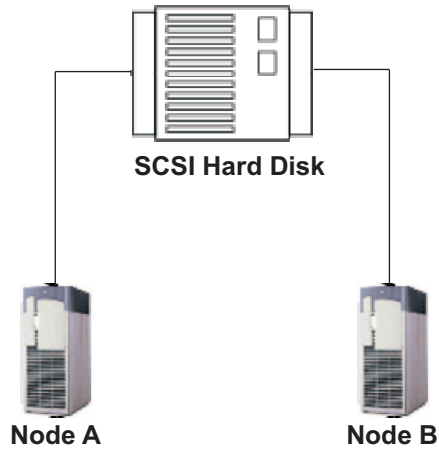


Figure 2. Installing in a cluster

On each node run zero, one, or more cluster resource groups. In case of failure, for example, of node **A** all the cluster resource groups associated to the failing node failover to node **B**. In this way node **B** runs all the cluster-aware applications that were running on node **A**.

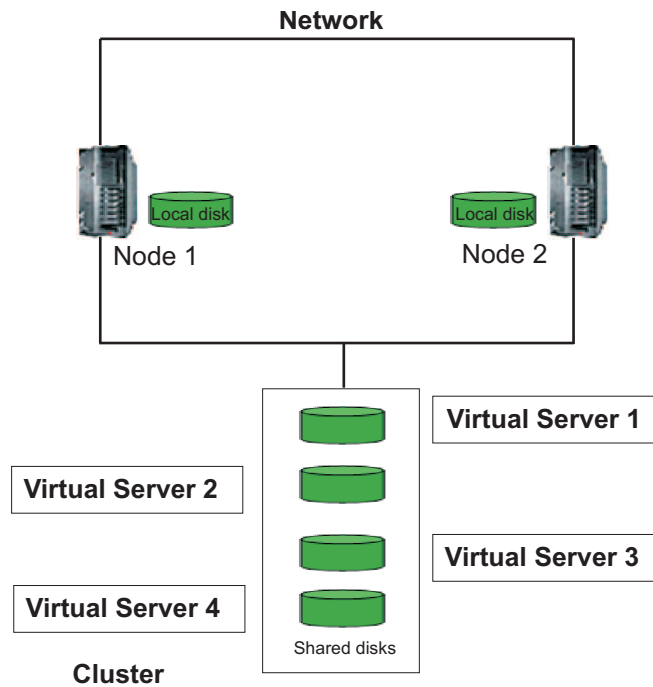


Figure 3. Clusters in an IBM Workload Scheduler network

To have IBM Workload Scheduler working in a cluster environment you can:

- Install the IBM Workload Scheduler agent locally on the hard disk of one of the nodes if you need to schedule on that cluster node only (as a single computer). This works like a normal installation. No cluster awareness is required.
- Install the IBM Workload Scheduler agent on one or more virtual servers if you need to schedule jobs on that virtual server. Cluster awareness is required.

To configure IBM Workload Scheduler to work in Windows cluster environment you are required to create a virtual server, adding to it a physical disk resource type and installing IBM Workload Scheduler on that disk.

A virtual server is a group containing a network name resource, an IP address resource, and additional resources necessary to run one or more applications or services. Clients can use the network name to access the resources in the group, analogous to using a computer name to access the services on a physical server. However, because a virtual server is a group, it can be failed over to another node without affecting the underlying name or address.

The new cluster resource type created to manage an IBM Workload Scheduler agent will perform a graceful shutdown and start up the agent during a failover.

## **Prerequisites**

### **About this task**

The following are prerequisites for the correct setup of IBM Workload Scheduler on the cluster:

#### **Windows Cluster Server**

A fully configured, up and running Windows Cluster Server must be ready.

#### **A configured File Server Service for Windows 2008**

The File Server Service is a service containing at least the virtual IP address resource, the network name resource, and the physical disk resource. The Cluster File Server resource can contain other application resources, not only IBM Workload Scheduler ones.

To create the File Server Service, you can use the Failover Cluster Manager console. You must do the following:

- Create the File Server
- Associate the static IP address to the File Server Resource
- Add the Shared Disk to the File Server Resource created
- The system automatically adds to the File Server Resource the Network Name Resource with the same name

See the Windows documentation for more details.

#### **A configured Role for Windows 2012**

A Role, formerly called File Server Service, contains at least the virtual IP address resource, the network name resource, and the physical disk resource. The Role can contain other application resources, not only IBM Workload Scheduler ones.

To create a File Server Role with type General Use, you can use the Failover Cluster Manager console. You must do the following:

- Create the File Server Role
- Associate the static IP address to the File Server Role
- Add the Shared Disk to the File Server Role created
- The system automatically adds to the File Server Role the Network Name Resource with the same name

See the Windows documentation for more details.

### **A Domain Administrator user**

A Domain Administrator user ready to use (the user should belong to the Administrator group of all the nodes of the cluster) and password.

### **A domain user**

Specify a domain user as an IBM Workload Scheduler user during the installation. If a valid domain is not specified, a local user is created by default.

### **Grant access rights to Cluster Administrator**

Verify that the cluster administrator account has the following right:

**Replace a process level token.** To add this right to the Cluster Administrator account open **Control Panel** → **Administrative Tools** → **Local Security Policy** → **Local Policies** → **User Rights Assignment** and add the Cluster Administrator user account to the **Replace a process level token** security policy list. This right is required to enable the Cluster Administrator to act as the IBM Workload Scheduler user. In this way the IBM Workload Scheduler custom resource that runs with the rights of the Cluster Administrator user, is able to stop, start, and link IBM Workload Scheduler. Restart the cluster nodes to have this change take effect. This operation is required only the first time you configure IBM Workload Scheduler to work in the Windows cluster environments.

### **Install Microsoft Visual C++ 2005 Redistributable Package (x86 or x64) on other cluster nodes**

All nodes in the cluster must be able to support the use of C++. This is achieved on a given node by installing the *Microsoft Visual C++ 2005 Redistributable Package (x86 or x64)*. The installation of the IBM Workload Scheduler cluster enabler installs this package on the node where the enabler is installed, but to allow you to switch to the other nodes in the cluster, the package must be installed on them, too.

Follow this procedure:

1. Either download the Visual C++ 2005 Redistributable Package (x86) or the Visual C++ 2005 Redistributable Package (x64) from Microsoft Download Center website or go to <http://www.microsoft.com> and search for the package by name. Download the package file (vcredist\_x86.exe or vcredist\_x64.exe)
2. Copy the package to each node in the Cluster Virtual Server Group:
3. On each node in the group (other than the one where you install the cluster enabler), do the following:
  - a. Log on as Domain Administrator
  - b. Run vcredist\_x86.exe or vcredist\_x64.exe

### **Install and configure a new IBM Workload Scheduler agent** **About this task**

To install IBM Workload Scheduler in a cluster-aware configuration, use the following procedure:

1. Install the IBM Workload Scheduler agent:
  - a. Select one node of the cluster. This node must be used for any subsequent operations (such as fix pack installations).
  - b. Log on to the node by using a user with Domain Administrator privileges.
  - c. Choose the Microsoft Virtual Server (for Windows 2008) or Clustered Role (for Windows 2012) where you want to install the IBM Workload Scheduler agent.



- d. Install IBM Workload Scheduler:
  - Specify a domain user as user for which you want to install IBM Workload Scheduler
  - Specify the shared disk that is associated to that Virtual Server (for Windows 2008) or Clustered Role (for Windows 2012) as destination directory
- e. Install IBM Workload Scheduler.
2. Make the IBM Workload Scheduler cluster aware:
  - a. Run the Windows Command Prompt.
  - b. Move into the IBM Workload Scheduler home directory (on the shared disk).
  - c. Run `twc_env.cmd` to load the IBM Workload Scheduler environment variables.
  - d. Run `Shutdown.cmd` to stop IBM Workload Scheduler.
  - e. Move into the cluster directory.
  - f. Run the utility `twcClusterAdm.exe` to configure IBM Workload Scheduler remotely on all nodes of the cluster and to install the IBM Workload Scheduler Cluster Resource. See “Example 1: First installation of IBM Workload Scheduler in a Windows cluster environment” on page 23 for an installation example.

**Note:** **ITWSWorkstation** is the name of the IBM Workload Scheduler cluster resource type. By default, when it is created by using the **twcClusterAdm** command line, the instance name is `ITWSWorkstation_<domain_name>_<user_name>`.

**Note:** During IBM Workload Scheduler 9.4 Cluster installation, the following parameters must not be specified in double-byte character set (DBCS) characters:

- User
- Domain

3. Define a new workstation object on the master domain manager either by using **composer** or the Dynamic Workload Console. Verify that the **node** name specified is resolved by the DNS and the IP address can be pinged from the master domain manager. If you are using end-to-end network configuration, you must specify the IP address that you specified for the NET/IP Cluster Resource Value.
4. Start the **ITWSWorkstation** resource instance that you created in step 2f:
  - a. Locate the **ITWSWorkstation** resource instance:
 

**On Windows 2008 cluster:**  
From the Failover Cluster Manager console, find the File Server where the **ITWSWorkstation** resource instance is saved.

In the Figure 4 on page 14 example, the `ITWSWorkstation_CLUSTER_aia86` resource instance is saved in the `twcFS` File Server.

**On Windows 2012 cluster:**  
From the Failover Cluster Manager console, find the Role where the **ITWSWorkstation** resource instance is saved.
  - b. Right-click on the selected **ITWSWorkstation** resource instance and select **Bring Online**.

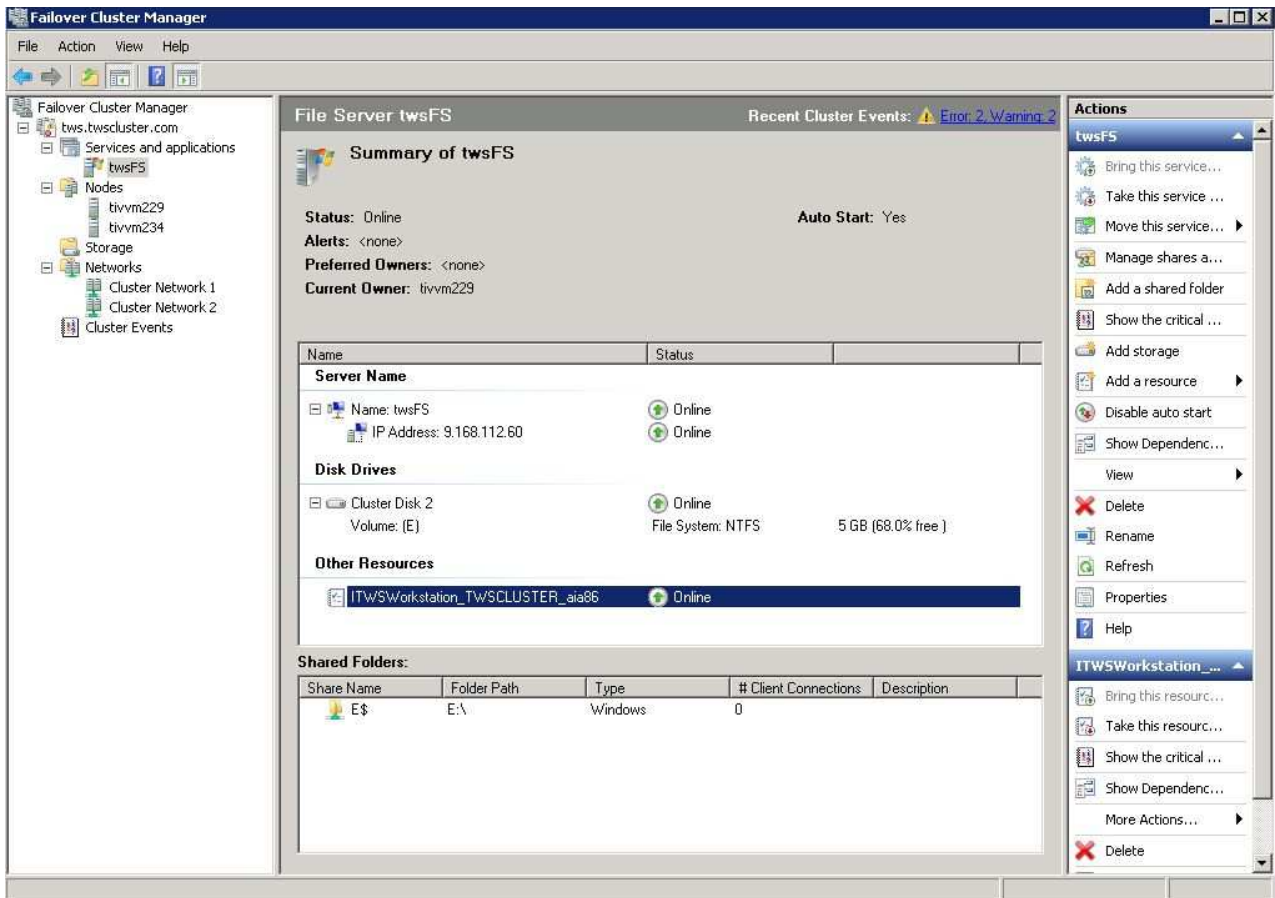


Figure 4. File Server example on Windows Server 2008

5. Wait until the final job stream runs or generate a new plan to add the new workstation to the plan.

## Upgrading agents to a new version and available fix packs

You can upgrade cluster nodes to the latest version of the product either automatically, by running **twClusterUpg** script, or manually.

You can upgrade from:

- V9.1 or later

### Upgrading cluster nodes automatically by using twClusterUpg

Use the **twClusterUpg** command to upgrade automatically IBM Workload Scheduler agents in a cluster-aware configuration, from V9.1 or later, to the general availability (GA) version of the product.

### About this task

To perform the upgrade, run the following procedure:

1. Ensure that your environment meets the prerequisites that are listed in “Prerequisites” on page 11.
2. Move to the directory where you downloaded the images of the general availability (GA) version.

3. On all the nodes you want to upgrade, run the **twscClusterUpg** script with Cluster Administrator and Domain Administrator rights by using the following syntax:

```
cscript.exe twscClusterUpg.vbs
-groups <group1, ..., groupn>
-passwords <pwd1, ..., pwn>
```

Where:

**group1, ..., groupn**

Specify the groups that you want to upgrade.

**pwd1, ..., pwn**

Specify the password of the IBM Workload Scheduler user you had used to install the product in the group. Specify the passwords in the same order that you specify the groups.

The script pauses all the nodes where IBM Workload Scheduler is installed and proceeds to upgrade the product group by group. If errors occur, analyze the `%temp%\TWA\TWSClusterMainLog.txt` file.

Where `%temp%` is the temporary directory of the user that is running the **twscClusterUpg** command.

The **twscClusterUpg** command runs the **twscinst.vbs**, the **twscpatch.vbs**, and the **twscClusterAdm.exe** commands. If the `TWSClusterMainLog.txt` indicates that one of these commands fails, analyze the corresponding log for further details.

4. Resume all the nodes of the cluster you upgraded, by running the following command against each node:

```
cluster.exe node <node_name> /Resume
```

Where `node_name` is the name of the node you want to resume.

5. Bring online the IBM Workload Scheduler resources on all the nodes, by running the following command against each resource:

```
cluster.exe res <res_name> /Online
```

Where `<res_name>` is the name of the IBM Workload Scheduler resource.

## Upgrading cluster nodes manually

You can upgrade IBM Workload Scheduler agents in a cluster-aware configuration, from V9.1 or later, to the general availability (GA) version of the product, by running the following manual procedure.

### About this task

1. Ensure your environment meets the prerequisites listed in “Prerequisites” on page 11.
2. Set all the nodes of the cluster to the **Pause** state. You must pause all the nodes in which you defined at least one cluster resource. In this way the IBM Workload Scheduler cluster resources cannot be moved from one node to another. With this operation you fix the resources on the node where they are running. Perform this action by running the following command against each node:

```
cluster.exe node <node_name> /Pause
```

Where `<node_name>` is the name of the node to pause.

3. Set offline all the IBM Workload Scheduler resources belonging to the nodes of the cluster you paused to prevent IBM Workload Scheduler from upgrading the

resources DLL with a cluster service that is still loading the DLL. Perform this action by running the following command against each resource:

```
cluster.exe res <res_name> /Offline
```

Where *<res\_name>* is the name of the resource to set offline.

4. Move to the directory where you downloaded the images of the general availability (GA) version.
5. For all the nodes in the cluster, upgrade all the resources running on the nodes, by performing the following steps:

- a. Generate the installation registries for one of the cluster groups belonging to the node on which you are upgrading the resources, and upgrade the instance you are working on, using one of the following script:

```
cscript.exe twsinst.vbs -update -uname <user_name>  
-password <TWS_user_password>  
-acceptlicense yes  
-inst_dir TWS_install_dir -recovInstReg true
```

- b. Move to the *inst\_dir* directory where you upgraded the cluster node and update the remote Windows Services and the resource DLL, by running the following command:

```
twsClusterAdm.exe -update resource=<res_name> ask=yes  
-twsupd pwd <TWS_user_password>
```

Where *<res\_name>* is the name of the resource you are upgrading and *<TWS\_user\_password>* is the Windows password of the IBM Workload Scheduler user.

**Note:** If you added a node to your IBM Workload Scheduler cluster after installing IBM Workload Scheduler, run the following command:

```
twsClusterAdm.exe -update hosts=<hostname1>,<hostname2>,...  
resource=<res_name> ask=yes -twsupd pwd <TWS_user_password>
```

Where *<hostname1>,<hostname2>,...* are the host names of the cluster nodes that you added after the installation and you want to upgrade.

- c. Repeat steps 5a and 5b for all the resources present on this node.
6. Resume all the nodes of the cluster you paused in the Step 1 by running the following command against each node:

```
cluster.exe node <node_name> /Resume
```

Where *<node\_name>* is the name of the node you want to resume.

7. Bring online the IBM Workload Scheduler resources on all the nodes, by running the following command against each resource:

```
cluster.exe res <res_name> /Online
```

Where *<res\_name>* is the name of the IBM Workload Scheduler resource.

See “Example 10: Upgrading agents in a cluster-aware configuration” on page 26 for an upgrade example.

## **twsClusterAdm command with examples of usage**

### **About this task**

The **twsClusterAdm** command configures your IBM Workload Scheduler agent type installations in the Microsoft Windows cluster environment.

You can configure the following IBM Workload Scheduler agent type installations:

- Fault-tolerant agent only.
- Both fault-tolerant agent and dynamic agent.
- Dynamic agent only.

Both fault-tolerant agents and dynamic agents are monitored and can be involved in failover. For details, see the **ftaOff** and **lwaOn** monitoring options of **twClusterAdm** parameter **opts**, (described in “Syntax”).

If you want to install a domain manager on a cluster environment, you must specify the *link* option using the **twClusterAdm** parameter **opts** (described in “Syntax”). See “Example 9: First installation of domain manager in Windows cluster environment, specifying generic options” on page 26 for a worked example.

The **twClusterAdm** command:

- Configures IBM Workload Scheduler on all the nodes of the cluster or on a new joining cluster node.
- Installs the new IBM Workload Scheduler Cluster resource type for a first time installation. The name of this new cluster resource type is `ITWSWorkstation`.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within a cluster group.
- Removes IBM Workload Scheduler configuration from one or more nodes of the cluster.
- Upgrades the IBM Workload Scheduler Cluster resource type if a new version is available.

It works in several steps to complete a cluster-aware installation:

- Determines if setup is running in a cluster environment.
- Copies the IBM Workload Scheduler resource DLL to the cluster nodes.
- Updates IBM Workload Scheduler services startup mode from automatic to manual.
- Installs IBM Workload Scheduler services and registry key on the other nodes (because the current services and registry key was installed by the normal product installation).
- Registers IBM Workload Scheduler Resource Types.
- Creates a new instance of the IBM Workload Scheduler resource within a given cluster group (virtual server) and updates `localopts` and `JobManager.ini` configuration files.
- Installs Common Inventory Technology on the other nodes

## Syntax

### About this task

**Note:** The order and the position of the **twClusterAdm** command parameters must be respected.

```
twClusterAdm.exe -new  
  domain=<Windows_domain>  
  user=<TWS_user>  
  pwd=<TWS_user_password>  
  [hosts=<hostname1,hostname2...>]  
  [twshome=<TWS_home_dir>]
```

```

[ -res
  group=<cluster_group_name>
  ip=<IP_resource_name>
  net=<network_resource_name>
  disk=<disk_resource_name>
  [resname=<resource_instance_name>]
  [check_interval=<TWS_polling_interval>]
  [failover=yes|no]
  [looksalive=<lookalive_interval>]
  [isalive=<isalive_interval>]
  [tcpport=<tcp_port>]
  [opts=<generic_options>]
]
[-dll
  [path=<DLL_resource_path>]
]
[-force]
[-sharedDesktop [name=<desktop_name>]]

```

```

twscClusterAdm.exe -uninst
  domain=<Windows_domain>
  user=<TWS_user_name>
  [hosts=< hostname1,hostname2...>]

```

```

twscClusterAdm.exe -update
  resource=<resource_instance_name>
  [hosts=< hostname1,hostname2...>]
  [ask={yes|no}]
  [-force]
  [-twsupd [pwd=<TWS_user_password>]]

```

```

twscClusterAdm.exe -changeResName
  "<resource_instance_name>"
  "<new_resource_instance_name>"

```

## Parameters and arguments

### About this task

**-new** The **-new** parameter configures IBM Workload Scheduler on all the nodes of the cluster or on a new cluster node. It takes the following arguments:

**Domain**=<Windows\_domain>

The Windows User Domain of the IBM Workload Scheduler User. This parameter is mandatory if **-new** or **-uninst** is specified. This parameter must not be specified in double-byte character set (DBCS) characters.

**user**=<TWS\_user>

The Windows User Name of the IBM Workload Scheduler User. This parameter is mandatory if **-new** or **-uninst** is specified. This parameter must not be specified in double-byte character set (DBCS) characters.

**pwd**=<TWS\_user\_password>

The Windows password of the IBM Workload Scheduler user. This parameter is mandatory if **-new** is specified.

**hosts**=<hostname1,hostname2...>

The host names of the cluster nodes where you want to configure

IBM Workload Scheduler. Host names must be separated by commas. This parameter is optional. It can be used to configure a new joining node of the cluster.

**twshome**=<TWS\_home\_directory>

The directory where IBM Workload Scheduler is installed. This parameter is optional. If you do not specify this directory, the command discovers the installation directory.

**-res** The **-res** parameter adds a new instance of the IBM Workload Scheduler resource type to an existing cluster group. It takes the following arguments:

**group**=<cluster\_group\_name>

The name of the group (Virtual Server) where IBM Workload Scheduler is configured as the cluster resource. This parameter is mandatory.

**ip**=<IP\_resource\_name>

The name of the cluster IP resource type that the IBM Workload Scheduler resource depends on. This parameter is mandatory.

**net**=<network\_resource\_name>

The name of the network resource type that the IBM Workload Scheduler resource depends on. This parameter is mandatory.

**disk**=<disk\_resource\_name>

The name of the disk resource type that the IBM Workload Scheduler resource depends on. This parameter is mandatory.

**resname**=<resource\_instance\_name>

The name of the resource instance, as it appears in the Cluster Administrator (see Figure 4 on page 14). If this parameter is not supplied, the default value of ITWSWorkstation\_<domain\_name>\_<user\_name> is used.

**failover**=yes|no

If you specify **yes**, IBM Workload Scheduler can cause the failover of the virtual server group. If you do not specify this option IBM Workload Scheduler will not cause the failover of the virtual server group. This parameter is optional. Note that you can modify this setting directly from the Cluster Administrator console. Modifying the threshold and period values from the resource property tab you can enable or disable the automatic failover in case of resource failure. See the *Windows Cluster Guide* for more information.

**check\_interval**=<TWS\_polling\_interval>

The interval in milliseconds that the IBM Workload Scheduler resource waits between two health checks. This parameter is optional. Use values greater than 60000. The default value is 100000. You can change this value from the Cluster Administrator console: right-click the resource and select **Properties** → **Parameters**.

**lookalive**=<lookalive\_interval>

The interval in milliseconds at which the Cluster service polls the resource to determine if it appears operational. This parameter is optional. Use values greater than 10000. The default value is 10000. You can change this value from the Cluster Administrator console: right-click the resource and select **Properties** → **Advanced**.

**Isalive**=<isalive\_interval>

The interval in milliseconds at which the Cluster service polls the resource to determine if it is operational. This parameter is optional. Use values greater than 10000. The default value is 60000. You can change this value from the Cluster Administrator console: right-click the resource and select **Properties** → **Advanced**.

**tcpport**=tcp\_port

This parameter is reserved for future use.

**opts**=generic\_options

The **opts** parameter is used to specify a set of options. Each option is separated by a semicolon ";". The **opts** parameter accepts the following options:

- **Monitoring options:**

- **ftaOff**

- Monitoring option: use this option to disable monitoring (and failover) of fault-tolerant agents.

- **lwaOn**

- Monitoring option: use this option to enable monitoring (and failover) of dynamic agents.

For details about the monitoring options in your cluster environment, see Table 1.

Table 1. Monitoring options in your cluster environment.

Types of IBM Workload Scheduler agent installation in your cluster environment	Monitoring options for your agent installation	Result
• Agent installation composed of both fault-tolerant agent and dynamic agent • Agent installation composed of fault-tolerant agent only	No options are specified	Only fault-tolerant agent is monitored (default).
	opts=lwaOn;ftaOff	Only dynamic agent is monitored.
	opts=lwaOn	Both fault-tolerant agent and dynamic agent are monitored.
	opts=ftaOff	Neither fault-tolerant agent nor dynamic agent are monitored.
• Agent installation composed of dynamic agent only	No options are specified	Dynamic agent is monitored (default).
	opts=lwaOn;ftaOff	Dynamic agent is monitored.
	opts=lwaOn	Dynamic agent is monitored.
	opts=ftaOff	Dynamic agent is monitored.

- **killjob**

- Use this option to kill any job (and job child) running at the moment of the resource failure.

- **link**=<parent\_domain>!<parent\_domain\_manager>

- Use this option if you are configuring a cluster-aware domain manager. Specify the parent domain and manager of the agent you are configuring.

- For example if you are configuring a domain manager which is a child of the master domain manager (named *MyMasterWS* in the domain *MASTERDOM*), the value to specify is `link=MASTERDOM!MyMasterWS`.



The **kill** and the **link** options can be used together (for example, `opts=killjob;link=MASTERDM!MASTERWS;`). You can change this value from the Cluster Administrator console: right-click the resource and select **Properties** → **Parameters**. Change the values in the **genericOpts** field.

**-dll** The **-dll** parameter specifies that the `ITWSResources.dll` that implements the new IBM Workload Scheduler resource type needs to be installed. This parameter is mandatory the first time you configure IBM Workload Scheduler on the cluster or if a new node is joining the cluster. This parameter takes one optional argument:

**[path=<DLL\_resource\_path>]**

The path where the `ITWSResources.dll` must be installed. This parameter is optional. If you do not specify the path, the default value, `%systemRoot%\cluster`, is used. **Do not** specify the drive letter for the path. The path specified must exist and must be accessible on each node of the cluster.

**-force** The **-force** parameter optionally forces the installation of the IBM Workload Scheduler resource DLL (`ITWSResources.dll`) without checking the version. The parameter is ignored if you did not specify the **-dll** parameter.

**-sharedDesktop**

The **-sharedDesktop** parameter optionally specifies that **Jobmon** uses a shared desktop name to manage desktop heap memory allocation where multiple agents are installed on one computer (see “Resolving desktop heap size problems on workstations with more than three agents,” on page 51 for details). Use the same name for at least two agents on this computer to make the option effective.

**name=<desktop\_name>**

The optional desktop name. If you supply a name, it must be in single-byte characters (English alphabet), with no special characters allowed, except spaces, in which case you must surround it by double quotes. The default name (by not supplying the **name=<desktop\_name>** argument), is `TWS_JOBS_WINSTA`.

**-uninst**

The **-uninst** parameter uninstalls the cluster resource instance, and accepts the following arguments:

**Domain=<Windows\_domain>**

The Windows User Domain of the IBM Workload Scheduler User. This parameter is mandatory if you specify **-new** or **-uninst**. This parameter must not be specified in double-byte character set (DBCS) characters.

**user=<TWS\_user>**

The Windows User Name of the IBM Workload Scheduler User. This parameter is mandatory if you specify **-new** or **-uninst**. This parameter must not be specified in double-byte character set (DBCS) characters.

**hosts=<hostname1,hostname2...>**

The host names of the cluster nodes where you want to uninstall IBM Workload Scheduler. Host names have to be separated by commas. This parameter is optional. If you do not specify this

parameter, IBM Workload Scheduler is uninstalled from all nodes in the cluster except for the current node.

### **-update**

The **-update** parameter updates the IBM Workload Scheduler resource DLL of an existing instance, and accepts the following arguments:

**resource**=<resource\_instance\_name>

The IBM Workload Scheduler Resource Instance name as it appears within the cluster group. The default name is ITWSWorkstation\_<domain>\_<user>. This parameter is mandatory.

**ask**={yes | no}

Define if the resource DLL upgrade can be performed automatically. Supply yes to ask that the resource upgrade is performed automatically, if the upgrade version is greater than the actual version or - **force** parameter is provided. Supply no to ask that the resource is not upgraded. This parameter is optional. If it is not specified, the installation stops and, in case the upgrade version is greater than the actual version or - **force** parameter is provided, asks the operator to confirm the resource DLL upgrade and the restart of the related cluster resource.

**-force** Force the installation of the IBM Workload Scheduler resource DLL (ITWSResources.dll) without checking the version. This parameter is optional.

**-twsupd**

Define whether to update the Windows service registry after updating the resource DLL. Use this parameter only when updating the DLL after upgrading IBM Workload Scheduler to a new major version, such as 9.4. This parameter is optional.

**hosts**=<hostname1,hostname2...>

The host names of the cluster nodes on which you want to update IBM Workload Scheduler. Host names have to be separated by commas. This parameter is optional. If you do not specify this parameter, IBM Workload Scheduler is updated on all nodes in the cluster.

### **-changeResName**

The **-changeResName** parameter changes the cluster instance resource name, and accepts the following arguments:

"<resource\_instance\_name>"

The IBM Workload Scheduler Resource Instance name as it appears within the cluster group. The default name is ITWSWorkstation\_<domain>\_<user>. This argument is mandatory for **-changeResName**.

"<new\_resource\_instance\_name>"

The new name you want to use for the IBM Workload Scheduler resource instance. This argument is also mandatory for **-changeResName**.

## **Examples**

### **About this task**

For all the examples described below it is assumed that IBM Workload Scheduler 9.4 has been installed.

In all the examples described below the following definitions are used:

*mydom* Is the Windows User Domain of the IBM Workload Scheduler user.

*mytwsuser*

Is the IBM Workload Scheduler user name.

*mytwspwd*

Is the password for the *MYDOM\mytwsuser* domain user.

*myresgroup*

Is the name of the cluster resource group selected.

*myip*

Is the name of the IP Address resource type within the *myresgroup* resource group.

*mynetname*

Is the name of the Network Name resource type within the *myresgroup* resource group.

*mydisk*

Is the name of the Physical Disk resource type within the *myresgroup* resource group.

*my shared desktop*

Is the name of the shared desktop that all instances of **jobmon** will use.

*myResName*

Is the customized name of the resource instance.

The examples are as follows:

- “Example 1: First installation of IBM Workload Scheduler in a Windows cluster environment”
- “Example 2: Install and configure the new custom resource for an existing installation of IBM Workload Scheduler” on page 24
- “Example 3: Add a new agent in a cluster environment with IBM Workload Scheduler already installed” on page 24
- “Example 4: Add a custom resource type instance to an existing cluster group” on page 25
- “Example 5: Configure IBM Workload Scheduler in a new joining node of the cluster” on page 25
- “Example 6: Deregister IBM Workload Scheduler on all nodes of the cluster except for the current node” on page 25
- “Example 7: Install a new version of the cluster resource DLL into the cluster” on page 25
- “Example 8: Force the upgrading/downgrading of the cluster resource DLL into the cluster” on page 26
- “Example 9: First installation of domain manager in Windows cluster environment, specifying generic options” on page 26
- “Example 10: Upgrading agents in a cluster-aware configuration” on page 26
- “Example 11: First installation of IBM Workload Scheduler in a Windows cluster environment, defining shared desktop” on page 27
- “Example 12: First installation of IBM Workload Scheduler in Windows cluster environment, using customized resource instance name” on page 27
- “Example 13: Changing the resource instance name” on page 28

**Example 1: First installation of IBM Workload Scheduler in a Windows cluster environment:**

## About this task

First time installation of IBM Workload Scheduler in a Windows cluster environment.

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd  
-res group=myresgroup ip=myip net=mynetname disk=mydisk opts=lwaOn -dll
```

The command:

- Configures IBM Workload Scheduler on all the nodes of the cluster.
- Installs the new IBM Workload Scheduler Cluster resource type (named ITWSWorkstation) on all the nodes of the cluster.
- Copies the ITWSResources.dll to the %systemRoot%\cluster folder.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group.
- Adds a dependency from *myip*, *mynetname*, and *mydisk* to the resource.
- Enable monitoring of dynamic agents (lwaOn).

### Example 2: Install and configure the new custom resource for an existing installation of IBM Workload Scheduler:

#### About this task

Install and configure the new IBM Workload Scheduler custom resource for an existing installation of IBM Workload Scheduler.

**Note:** This example is applicable only if the IBM Workload Scheduler agent has already been installed for the same instance on all the nodes of the cluster and all the IBM Workload Scheduler services startup types are set to **Manual**.

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd  
-res group=myresgroup ip=myip net=mynetname disk=mydisk opts=lwaOn -dll
```

The command:

- Installs the new IBM Workload Scheduler Cluster resource type (named ITWSWorkstation) on all the nodes of the cluster.
- Copies the ITWSResources.dll file to the %systemRoot%\cluster folder.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group.
- Adds a dependency from *myip*, *mynetname*, and *mydisk* to the resource.
- Enable monitoring of dynamic agents (lwaOn).

### Example 3: Add a new agent in a cluster environment with IBM Workload Scheduler already installed:

#### About this task

Add a new IBM Workload Scheduler agent in a cluster environment where an agent of IBM Workload Scheduler has been installed and configured in a different Virtual Server.

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd  
-res group=myresgroup ip=myip net=mynetname disk=mydisk opts=lwaOn
```

The command:

- Configures IBM Workload Scheduler on all the nodes of the cluster.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group

- Adds a dependency from *myip*, *mynetname*, and *mydisk* to the resource.
- Enable monitoring of dynamic agents (lwaOn).

**Example 4: Add a custom resource type instance to an existing cluster group:  
About this task**

Add an instance of the IBM Workload Scheduler custom resource type to an existing cluster group.

**Note:** This example is applicable only if the IBM Workload Scheduler agent has been installed and configured, and the IBM Workload Scheduler custom resource type has been installed and registered.

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd
-res group=myresgroup ip=myip net=mynetname disk=mydisk
```

The command:

- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group
- Adds a dependency from *myip*, *mynetname*, and *mydisk* to the resource.

**Example 5: Configure IBM Workload Scheduler in a new joining node of the cluster:  
About this task**

Configure IBM Workload Scheduler in a new joining node of the cluster.

**Note:** This example is applicable only if the IBM Workload Scheduler agent has been installed and configured in the cluster environment. Possibly you have been using IBM Workload Scheduler for a long time, you have bought a new node for this cluster, and you want IBM Workload Scheduler to be able to move there in case of failure.

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd
hosts=my_new_joining_host_name -dll
```

The command configures IBM Workload Scheduler and installs the IBM Workload Scheduler cluster resource DLL on the *my\_new\_joining\_host\_name* node.

**Example 6: Deregister IBM Workload Scheduler on all nodes of the cluster except for the current node:  
About this task**

Deregister IBM Workload Scheduler on all the nodes of the cluster except for the current node. See the section relative to uninstall procedure for more details.

```
twClusterAdm.exe -uninst domain=MYDOM user=mytwsuser
```

The command removes IBM Workload Scheduler configuration from all the nodes of the cluster except for the current node.

To uninstall IBM Workload Scheduler from the current node you must use the normal uninstallation procedure described in *Planning and Installation Guide*.

**Example 7: Install a new version of the cluster resource DLL into the cluster:**

### About this task

Install a new version of the IBM Workload Scheduler cluster resource DLL into the cluster.

```
twClusterAdm.exe -update resource=<resource_name>
```

The command upgrades the IBM Workload Scheduler Cluster resource type if a new version is available.

### Example 8: Force the upgrading/downgrading of the cluster resource DLL into the cluster:

#### About this task

Force the upgrading/downgrading of the IBM Workload Scheduler cluster resource DLL into the cluster.

```
twClusterAdm.exe -update resource=<resource name> -force
```

The command upgrades the IBM Workload Scheduler cluster resource without verifying if the version is greater than the version of the installed version.

### Example 9: First installation of domain manager in Windows cluster environment, specifying generic options:

#### About this task

First time installation of an IBM Workload Scheduler domain manager in a Windows cluster environment, specifying the `kill` and `link` generic options

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd -res  
group=myresgroup ip=myip net=mynetname disk=mydisk  
opts=killjob;link=MASTERDM!MASTER; -dll
```

The command:

- Configures IBM Workload Scheduler on all the nodes of the cluster.
- Installs the new IBM Workload Scheduler Cluster resource type (named ITWSWorkstation) on all the nodes of the cluster.
- Copies the ITWSResources.dll to the %systemRoot%\cluster folder.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group.
- Adds a dependency from `myip`, `mynetname`, and `mydisk` to the resource.
- Sets the generic options `kill` and `link`.

The `link` option specifies the parent domain of the domain manager, and the parent domain manager workstation, so that the installation process can correctly manage the unlinking and relinking required. If the `link` option is omitted, or supplied with incorrect values, the configuration cannot complete correctly. However, you do not need to repeat the installation to resolve the problem. Instead, go to the IBM Workload Scheduler resource instance property panel, and under the **Parameters** tab add the link in the **genericOpts** field. When you activate the cluster the information in the `link` option is used to complete the configuration.

### Example 10: Upgrading agents in a cluster-aware configuration:

#### About this task

Upgrade agents in a cluster-aware configuration as follows:

1. Set all the nodes of the cluster to the **Pause** state. Perform this action by running the following command against each node:
 

```
cluster.exe node NODE1 /Pause
```
2. Set offline all the IBM Workload Scheduler resources belonging to the nodes of the cluster you paused:
 

```
cluster.exe res RES1 /Offline
```
3. For all the nodes in the cluster, upgrade all the resources running on the nodes, by performing the following steps:
  - a. Generate the installation registries for one of the cluster groups belonging to the node on which you are upgrading the resources, and upgrade the instance you are working on, using the following script:
 

```
cscript.exe twsinst.vbs -update -uname user1
          -password password1
          -inst_dir "C:\Program Files\IBM\TWA" -recovInstReg true
```
  - b. Update the remote Windows Services and the resource DLL, by running the following command:
 

```
twsClusterAdm.exe -update resource=RES1 ask=yes -twsupd pwd us1pass1
```
  - c. Repeat steps 3a and 3b for all the resources present on this node.
4. Resume all the nodes of the cluster you paused in step 1 by running the following command against each node:
 

```
cluster.exe node NODE1 /Resume
```
5. Bring online the IBM Workload Scheduler resources on all the nodes, by running the following command against each resource:
 

```
cluster.exe res RES1 /Online
```

**Example 11: First installation of IBM Workload Scheduler in a Windows cluster environment, defining shared desktop:**

**About this task**

First time installation of IBM Workload Scheduler in a Windows cluster environment, defining a shared desktop to be used by **Jobmon** (this is like example 1, but with the addition of the shared desktop):

```
twsClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd
-res group=myresgroup ip=myip net=mynetname disk=mydisk opts=lwaOn -dll -sharedesktop
```

The command:

- Configures IBM Workload Scheduler on all the nodes of the cluster.
- Installs the new IBM Workload Scheduler Cluster resource type (named ITWSWorkstation) on all the nodes of the cluster.
- Copies the ITWSResources.dll to the %systemRoot%\cluster folder.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group.
- Adds a dependency from *myip*, *mynetname*, and *mydisk* to the resource.
- Enable monitoring of dynamic agents (lwaOn).
- Defines that **jobmon** uses the default shared desktop name

**Example 12: First installation of IBM Workload Scheduler in Windows cluster environment, using customized resource instance name:**

## About this task

First time installation of IBM Workload Scheduler in a Windows cluster environment, using a customized resource instance name (this is like example 1, but with the addition of the customized resource instance name):

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd  
-res group=myresgroup ip=myip net=mynetname disk=mydisk resname=myResName opts=lwa0n -d11
```

The command:

- Configures IBM Workload Scheduler on all the nodes of the cluster.
- Installs the new IBM Workload Scheduler Cluster resource type (named ITWSWorkstation) on all the nodes of the cluster.
- Copies the ITWSResources.dll to the %systemRoot%\cluster folder.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group.
- Adds a dependency from *myip*, *mynetname*, and *mydisk* to the resource.
- Defines that the resource instance name is *myResName*.
- Enable monitoring of dynamic agents (lwa0n).

### Example 13: Changing the resource instance name:

#### About this task

Changing the name of an existing resource instance:

```
twClusterAdm.exe -changeResName "ITWSWorkstation_CLUSTER_SA_DM1"  
"myResName"
```

The command changes the resource instance name from *ITWSWorkstation\_CLUSTER\_SA\_DM1* to *myResName*.

### Example 14: First installation of domain manager in Windows cluster environment, specifying monitoring options of dynamic scheduling:

#### About this task

First time installation of an IBM Workload Scheduler domain manager in a Windows cluster environment, specifying the lwa0n generic option.

```
twClusterAdm.exe -new domain=MYDOM user=mytwsuser pwd=mytwspwd -res  
group=myresgroup ip=myip net=mynetname disk=mydisk  
opts=lwa0n
```

The command:

- Configures IBM Workload Scheduler on all the nodes of the cluster.
- Installs the new IBM Workload Scheduler Cluster resource type (named ITWSWorkstation) on all the nodes of the cluster.
- Copies the ITWSResources.dll to the %systemRoot%\cluster folder.
- Creates an instance of the IBM Workload Scheduler Cluster resource type within the specified cluster group.
- Adds a dependency from *myip*, *mynetname*, and *mydisk* to the resource.
- Sets the generic option lwa0n to enable monitoring of the dynamic scheduling agent.



# Operating IBM Workload Scheduler in Windows cluster environment

## About this task

This section describes how to operate IBM Workload Scheduler in the Windows cluster environments. It is divided into the following subsections:

- “Cluster resource dependencies”
- “Start up and shut down IBM Workload Scheduler” on page 30
- “The new “cluster instance name” local option” on page 30

## Cluster resource dependencies

One of the most important steps when running IBM Workload Scheduler in the Windows cluster environments is to verify that the dependencies have been set correctly.

To ensure IBM Workload Scheduler works correctly, the IBM Workload Scheduler cluster resource instance has to depend on the following resource types:

- IP Address
- Physical Disk
- Network Name

as shown in the following example on Windows Server 2008

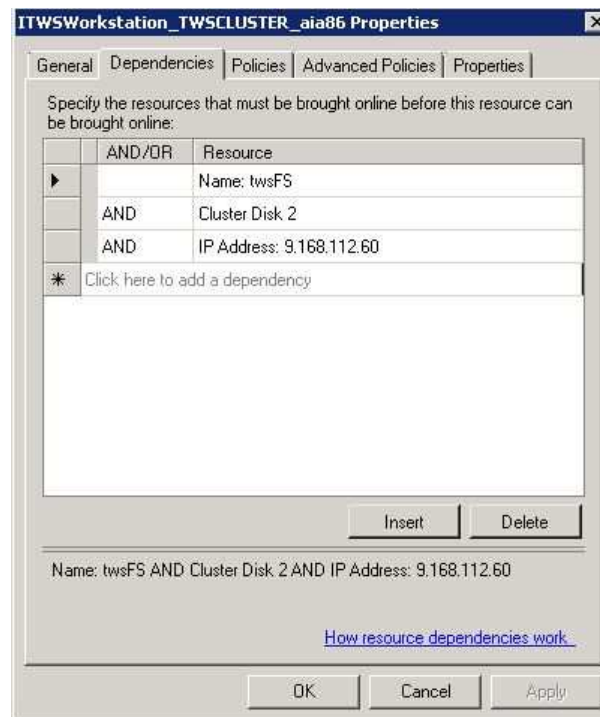


Figure 5. Resource Dependencies tab (Windows Server 2008)

You can decide to add more dependencies to ensure IBM Workload Scheduler launches jobs only after a given service is available. This happens when IBM Workload Scheduler schedules jobs that prerequisite other cluster aware applications. For example, to ensure the SQL job is launched only after the cluster aware relational database is available, add a dependency from the relational database cluster resource to the IBM Workload Scheduler cluster resource.

## Start up and shut down IBM Workload Scheduler

### About this task

The following methods can no longer be used to stop IBM Workload Scheduler because they will cause a failure of the IBM Workload Scheduler cluster resource:

- Conman shut
- Shutdown.cmd
- StartUp.cmd
- Conman start if the ITWSWorkstation resource is offline.
- StartupLwa.cmd

Use the following scripts to stop and start IBM Workload Scheduler (you can rename them if required):

- ShutDown\_clu.cmd
- StartUp\_clu.cmd
- ShutDownLwa.cmd

The above scripts will be automatically created under the IBM Workload Scheduler installation directory by the twsClusterAdm.exe program.

If you do not use these scripts, you must run the following commands to stop and start IBM Workload Scheduler services.

### Stop:

#### About this task

```
cluster res <TWSresource instance name> /offline
```

### Start:

#### About this task

```
cluster res <TWS resource instance name>/online
```

### Examples:

#### About this task

If ITWSWorkstation\_DOMAIN\_MST\_UserR is the name of the TWS resource instance, to shut down IBM Workload Scheduler you have to use:

```
cluster res ITWSWorkstation_DOMAIN_MST_UserR /offline
```

To start IBM Workload Scheduler services you have to use:

```
cluster res ITWSWorkstation_DOMAIN_MST_UserR /online
```

where cluster is the Windows command to administer the cluster (run from the Windows Command prompt).

## The new “cluster instance name” local option

### About this task

One of the steps of the twsClusterAdmin utility is instance name registration of the IBM Workload Scheduler cluster resource within the local option localopts file.

The IBM Workload Scheduler agent uses the value of this new local option to signal to the IBM Workload Scheduler cluster resource that the agent has received a stop command.

It is important to change the value of the `cluster instance name` local option every time the IBM Workload Scheduler resource instance name is changed. If the `clusterinstancename` local option does not point to the right name, the IBM Workload Scheduler resource will be set to failure state from the cluster resource monitor. Do not specify this name in double-byte character set (DBCS) characters.

To change the IBM Workload Scheduler resource instance name use the following procedure:

1. Take the IBM Workload Scheduler resource instance offline using the Cluster Administrator console. To verify if IBM Workload Scheduler stopped correctly you can use the Cluster Administrator console and check the status of the IBM Workload Scheduler resource instance. If the resource has failed to stop you can check in the cluster and IBM Workload Scheduler logs for the reason. See “Traces” on page 33 for more details on the log files.
2. From the Failover Cluster manager console, select the resource and modify the IBM Workload Scheduler cluster resource name from `cluster_resource_name` to `cluster_resource_name_new`.

**Note:** Do not specify this name in double-byte character set (DBCS) characters.

3. Open the `localopts` file using Notepad. Modify the value of the `clusterinstancename` local option. Check that the name is the same you specified for the IBM Workload Scheduler cluster resource in Step 2.
4. Modify the cluster instance name in the `StartUpclu.cmd` and `ShutDownclu.cmd` scripts.
5. Bring the IBM Workload Scheduler resource instance online in the Failover Cluster manager console.

## IBM Workload Scheduler Cluster Administrator extension

This section describes the Cluster Administrator extension. It is divided into the following subsections:

- “Cluster Administrator extension overview”
- “Installing the Cluster Administrator extension” on page 32

### Cluster Administrator extension overview

The Cluster Administrator is a system utility with a graphical user interface that allows administrators to manage cluster objects, handle maintenance, and monitor cluster activity.

The IBM Workload Scheduler Cluster Administrator extension is a dynamic-link library that, when installed, extends the Cluster Administrator console with a new property sheet and a wizard page that allows you to view and edit the IBM Workload Scheduler resource properties.

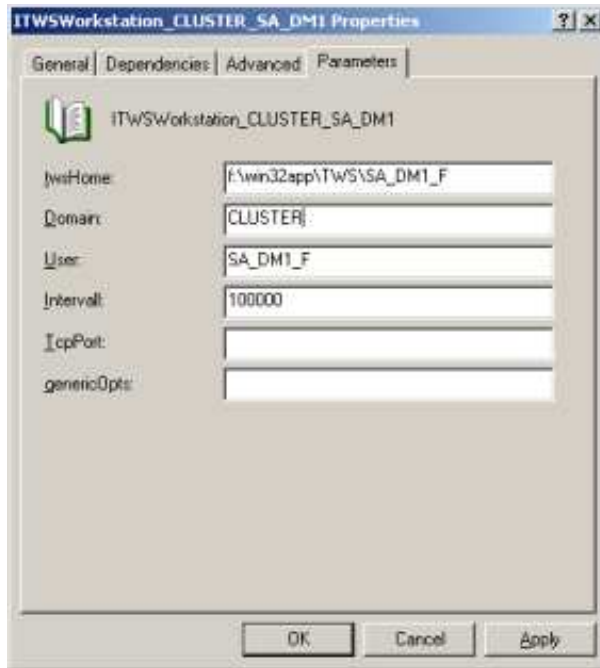


Figure 6. New Properties Parameters tab

The graphic shows the new properties page **Parameters** tab that allows you to modify the ITWSWorkstation cluster resource parameters.

## Installing the Cluster Administrator extension

### About this task

Install this component only if you want to edit the properties directly from Cluster Administrator console.

If you do not install the Cluster Administration extension, the **Parameters** tab (see Figure 6). is not available. To modify ITWSWorkstation cluster resource parameters, you will have to change these properties using the `cluster.exe` system utility.

Install the Cluster Administrator extension on any computer where the Cluster Administrator console will be used.

Use the following procedure to install a new Cluster Administrator extension:

1. Copy the `ITWSWorkstationEx.dll` and the `ITWSExInst.cmd` files from the `<TWS_HOME>\cluster` directory into the directory where you want to install the Administrator Extension. You can use the default directory for the cluster: `%systemRoot%\cluster`.
2. Double click on `ITWSExInst.cmd`, or run it from a command shell to install the Administrator Extension.

## Uninstalling IBM Workload Scheduler

### About this task

The steps to remove the product must be launched from the same node that was used to install IBM Workload Scheduler and subsequent fix packs.

Complete the following procedure:

1. Run the utility `TwsClusterAdm -uninst`. This utility removes the IBM Workload Scheduler services and registry keys from cluster nodes other than the current node. Optionally, use `-host hostname1, hostname2` arguments to specify the remote nodes from which the IBM Workload Scheduler service must be removed. Do not specify the primary node where the IBM Workload Scheduler agent was installed, otherwise you will remove the original service and make the IBM Workload Scheduler agent unusable.
2. Manually remove the IBM Workload Scheduler custom resource instance from the cluster group. You can use the Cluster Administrator console to do this.
3. Optionally, deregister the resource type by using the command:  

```
cluster restype ITWSWorkstation /delete
```

**Note:** Do not deregister the resource if instances are present in other cluster groups.

4. Optionally, delete the DLL `ITWSResources.dll` from the installation directory (the default directory is `%systemRoot%\cluster`).

To remove IBM Workload Scheduler from the current node, you can run the normal uninstallation program, see *Planning and Installation Guide*.

---

## Troubleshooting

This part of the guide gives troubleshooting information about IBM Workload Scheduler in a Windows cluster environment. The information here applies to the IBM Workload Scheduler engine and its installation for this environment. For more troubleshooting information about IBM Workload Scheduler, see *Troubleshooting Guide*.

For more information, see the following sections:

- “Traces”
- “Error 1314 taking online the resource and the Workstation does not link” on page 34
- “IBM Workload Scheduler resource instance reports fail status or IBM Workload Scheduler user jobs go in the abend state” on page 34
- “Windows Report panel with Jobmon.exe” on page 35
- “Cluster: IP validation error on Netman stdlist” on page 35

## Traces

IBM Workload Scheduler maintains logs for different activities in different places. For more information, see *Troubleshooting Guide*.

The new cluster enablement pack introduces two trace files in the `TWSInstallation Directory\stdlist\traces` directory:

### **clu\_offline.log**

When the IBM Workload Scheduler custom resource is taken offline (each time a failover happens), the IBM Workload Scheduler custom resource launches the **conman** command line to stop and unlink the instance. In this log you can find the output of the command.

### **clu\_online.log**

When the IBM Workload Scheduler custom resource is brought online (each time a failover happens), the IBM Workload Scheduler custom

resource launches the **conman** command line to link the workstation to its domain manager. In this log you can find the output of the command **conman link @!@;noask**.

Any action the IBM Workload Scheduler custom resource follows is logged within the system cluster log file. This is a file named `cluster.log` located under the `%systemRoot%\cluster` folder.

## Error 1314 taking online the resource and the Workstation does not link

The problem could be related to the rights of the cluster administrator. To check this:

1. Open the cluster log file (`cluster.log`) located in the `%systemRoot%\cluster` folder.
2. Look for the strings containing `ITWSWorkstation`. These are the messages logged by IBM Workload Scheduler custom resource.
3. If you see a message like:

```
<time> ERR ITWSWorkstation <resource instance name>: SubmitTwsCommand:
CreateProcessWithLogonW failed <TWS_HOME>\conman.exe < start;noask>
'1314'
```

It means that the system error 1314, A required privilege is not held by the client, occurred launching the **conman** command.

4. To solve the problem, you must give the cluster user sufficient privileges to allow custom resource instance to submit IBM Workload Scheduler command link.

To solve this problem, add the **Replace a process level token** right to the cluster administrator account (this is the name of the user you chose when you configured the cluster). To add this right to the Cluster Administrator account open **Control Panel → Administrative Tools → Local Security Policy → Local Policies → User Rights Assignment** and add the Cluster Administrator user account to the **Replace a process level token** security policy list. This right is required in order to enable the Cluster Administrator to act as the IBM Workload Scheduler user. In this way the IBM Workload Scheduler custom resource, that runs with the rights of the Cluster Administrator user, is able to stop, start, and link IBM Workload Scheduler. Reboot the cluster nodes to have this change take effect. This operation is required only the first time you configure IBM Workload Scheduler to work in the Windows cluster environment.

**You must reboot the cluster nodes for this change to take effect.**

## IBM Workload Scheduler resource instance reports fail status or IBM Workload Scheduler user jobs go in the abend state

**Problem:** If you run more than three instances of IBM Workload Scheduler on the same node with jobs running it is possible to have the following behavior:

- The IBM Workload Scheduler cluster resource instance is in fail status.
- IBM Workload Scheduler user jobs go in the **abend** or **fail** state.
- In this case you can find the following error message in `<TWS_HOME>\stdlist\date\TWSUSERNAME:`

```
+++++
+ AWSBIJ139E An internal error has occurred. Jobmon was unable to create a
+ new desktop on the window station associated with the calling process.
```

```

+ The error occurred in the following source code file:
+ ../../src/jobmon/monutil.c at line: 2454. The error mess
+ ++++++
AWSBIJ140E An internal error has occurred. Jobmon was unable to create the
Windows process environment to launch jobs. The error occurred in the
following source code file: ../../src/jobmon/monutil.c at line: 830.

The following error message is in the <TWS_HOME>\stdlist\logs\
date_TWSMERGE.log file,e: 06:00:28 19.05.2006|BATCHMAN:* AWSBHT061E Batchman
has received a mailbox record indicating that the following job has
terminated unexpectedly: The system has run out of desktop heap.

06:00:28 19.05.2006|BATCHMAN:*
AWSBHT061E Batchman as received a mailbox record indicating that the following
job has terminated unexpectedly:
The system has run out of desktop heap.

```

**Solution:** The solution to this problem has a number of different options, and is described in “Resolving desktop heap size problems on workstations with more than three agents,” on page 51

## Windows Report panel with Jobmon.exe

**Problem:** After failover from node A to node B, sometimes Jobmon cause a core dump with a segmentation violation error to occur on node A. You can see the segmentation after node A is rebooted, or when logging on with IBM Workload Scheduler user. This does not cause a problem because IBM Workload Scheduler on node B works correctly after a second failover, and IBM Workload Scheduler also works on node A.

## Cluster: IP validation error on Netman stdlist

This problem occurs when the node field in a workstation definition is defined as a real IP address instead of a cluster network name resource

The problem is that the IP validation is performed using the IP address of the node when IBM Workload Scheduler is starting and not the IP address of the resources in the cluster.

You could see this warning when the parent/child agent is installed on a cluster and there is a mismatch between the real IP address that has been detected from the TCP/IP channel and the IP address declared in the definition of the workstation (property node). If the property node is a host name, this will be resolved first (querying the DNS).





---

## Chapter 4. IBM Workload Scheduler with HACMP

For detailed information, see the following sections:

- “High-Availability Cluster Multi-Processing”
- “UNIX cluster overview” on page 40

---

### High-Availability Cluster Multi-Processing

The High-Availability Cluster Multi-Processing (HACMP) tool builds UNIX-based, mission-critical computing operating systems. HACMP ensures that critical resources, such as applications, are available for processing. HACMP has two major components: high availability (HA) and cluster multi-processing (CMP).

The primary reason to create HACMP clusters is to provide a highly available environment for mission-critical applications. For example, an HACMP cluster might run a database server program to service client applications. Clients send queries to the server program, which responds to their requests by accessing a database stored on a shared external disk.

In an HACMP cluster, to ensure the availability of these applications, the applications are put under HACMP control. HACMP ensures that the applications remain available to client processes even if a component in a cluster fails. To ensure availability, in case of a component failure, HACMP moves the application (together with resources needed to access the application) to another node in the cluster.

You can find more details on the following topics:

- “Benefits”
- “Physical components of an HACMP cluster” on page 38

### Benefits

HACMP™ provides the following benefits:

- The HACMP planning process and documentation include tips and advice about the best practices for installing and maintaining a highly available HACMP cluster.
- When the cluster is operational, HACMP provides automated monitoring and recovery of all the resources that the application needs.
- HACMP provides a full set of tools for maintaining the cluster and ensures that the application is available to clients.

Use HACMP to:

- Set up an HACMP environment using online planning worksheets that simplify initial planning and setup.
- Ensure high availability of applications by eliminating single points of failure in an HACMP environment.
- Use high-availability features available in AIX®.
- Manage how a cluster handles component failures.
- Secure cluster communications.

- Set up fast disk takeover for volume groups managed by the Logical Volume Manager (LVM).
- Manage event processing for an HACMP environment.
- Monitor HACMP components and diagnose problems that might occur.

## Physical components of an HACMP cluster

HACMP provides a highly-available environment by identifying a set of resources that are essential to uninterrupted processing, and by defining a protocol that nodes use to collaborate to ensure that these resources are available. HACMP extends the clustering model by defining relationships among cooperating processors where one processor provides the service offered by a peer, when the peer is unable to do so.

An HACMP Cluster is made up of the following physical components:

- “Nodes” on page 39
- “Shared external disk devices” on page 40
- “Networks” on page 40
- “Clients” on page 40

HACMP allows you to combine physical components into a wide range of cluster configurations, providing you with flexibility in building a cluster that meets your processing requirements. Figure 7 on page 39 shows an example of an HACMP cluster. Other HACMP clusters can look very different, depending on the number of processors, the choice of networking and disk technologies, and so on.

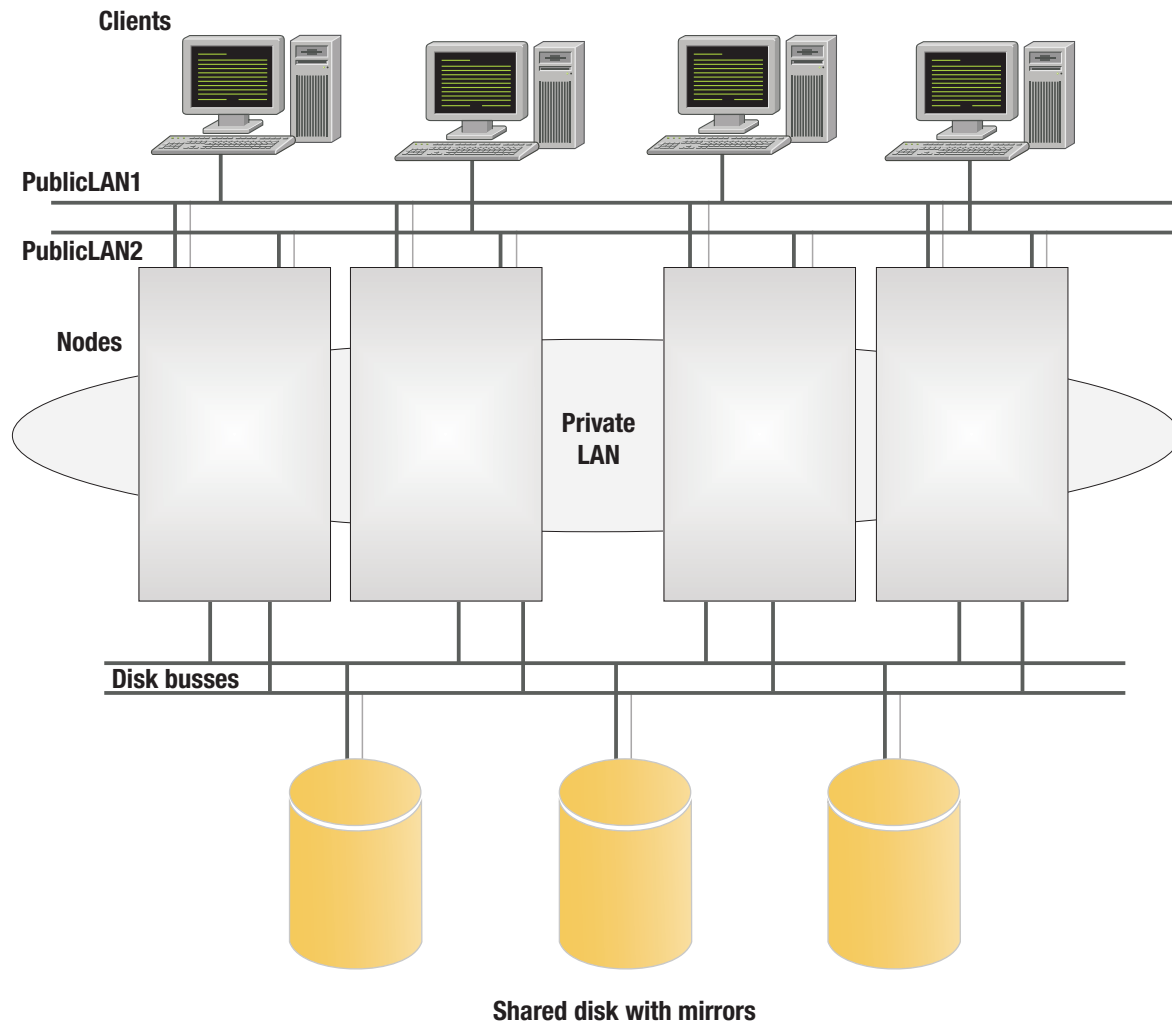


Figure 7. Shared disk with mirror

## Nodes

Nodes form the core of an HACMP cluster. A node is a processor that runs both AIX and HACMP. HACMP supports pSeries uniprocessor and symmetric multiprocessor (SMP) systems, and the Scalable POWERParallel processor (SP) systems as cluster nodes. The HACMP, an SMP system looks just like a uniprocessor. SMP systems provide a cost-effective way to increase cluster throughput. Each node in the cluster can be a large SMP machine, extending an HACMP cluster beyond the limits of a single system and allowing thousands of clients to connect to a single database.

In an HACMP Cluster, up to 32 computers or nodes cooperate to provide a set of services or resources to other remote clients. Clustering these servers to back up critical applications is a cost-effective high availability option. A business can use more of its computing power, to ensure that its critical applications resume running after a short interruption caused by a hardware or software failure.

In an HACMP cluster, each node is identified by a unique name. A node might own a set of resources (disks, volume groups, filesystems, networks, network

addresses, and applications). Typically, a node runs a server or a “back-end” application that accesses data on the shared external disks.

HACMP supports from 2 to 32 nodes in a cluster, depending on the disk technology used for the shared external disks. A node in an HACMP cluster has several layers of software components.

### **Shared external disk devices**

Each node must have access to one or more shared external disk devices. A shared external disk device is a disk physically connected to multiple nodes. The shared disk stores mission-critical data, typically mirrored or RAID-configured for data redundancy. A node in an HACMP cluster must also have internal disks that store the operating system and application binaries, but these disks are not shared.

Depending on the type of disk used, HACMP supports two types of access to shared external disk devices: non-concurrent and concurrent access.

- In non-concurrent access environments, only one connection is active at any time, and the node with the active connection owns the disk. When a node fails, disk takeover occurs when the node that currently owns the disk leaves the cluster and a surviving node assumes ownership of the shared disk.
- In concurrent access environments, the shared disks are actively connected to more than one node simultaneously. Therefore, when a node fails, disk takeover is not required.

### **Networks**

As an independent, layered component of AIX, HACMP is designed to work with any TCP/IP-based network. Nodes in an HACMP cluster use the network to allow clients to access the cluster nodes, enable cluster nodes to exchange heartbeat messages, and, in concurrent access environments, serialize access to data.

HACMP defines two types of communication networks, characterized by whether these networks use communication interfaces based on the TCP/IP subsystem (TCP/IP-based), or communication devices based on non-TCP/IP subsystems (device-based).

### **Clients**

A client is a processor that can access the nodes in a cluster over a local area network. Clients each run a front-end or client application that queries the server application running on the cluster node.

HACMP provides a highly-available environment for critical data and applications on cluster nodes. Note that HACMP does not make the clients themselves highly available. AIX clients can use the Client Information (Cinfo) services to receive notification of cluster events. Cinfo provides an API that displays cluster status information. The `/usr/es/sbin/cluster/clstat` utility, a Cinfo client provided with HACMP, provides information about all cluster service interfaces.

---

## **UNIX cluster overview**

This section describes the procedure for granting high availability using High-Availability Cluster Multi-processing (HACMP), on the AIX, UNIX, and Linux for IBM operating systems. It is divided into the following subsections:

- “Prerequisite knowledge” on page 41
- “Standby and takeover configurations” on page 41
- “Design limitations” on page 44

- “Supported configurations” on page 44

## Prerequisite knowledge

To understand the topics in this section, you must be familiar with IBM Workload Scheduler and HACMP clusters:

### IBM Workload Scheduler

For an overview of IBM Workload Scheduler, see *IBM Workload Automation: Overview*.

### HACMP clusters

For a Quick Start Guide for HACMP clusters, see High Availability Cluster Multi-Processing for AIX Version 7.3 at

<http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=/com.ibm.aix.doc/doc/base/aixinformation.htm>

## Standby and takeover configurations

There are two basic types of cluster configuration:

### Standby

This is the traditional redundant hardware configuration. One or more standby nodes are set aside idling, waiting for a primary server in the cluster to fail. This is also known as hot standby. From now on, we refer to an active/passive configuration to mean a two-node cluster with a hot standby configuration.

### Takeover

In this configuration, all cluster nodes process part of the cluster's workload. No nodes are set aside as standby nodes. When a primary node fails, one of the other nodes assumes the workload of the failed node in addition to its existing primary workload. This is also known as mutual takeover.

Typically, implementations of both configurations will involve shared resources. Disks or mass storage such as a Storage Area Network (SAN) are most frequently configured as a shared resource.

As shown in Figure 8 on page 42, Node A is the primary node, and Node B is the standby node currently idling. Although Node B has a connection to the shared mass storage resource, it is not active during normal operation.

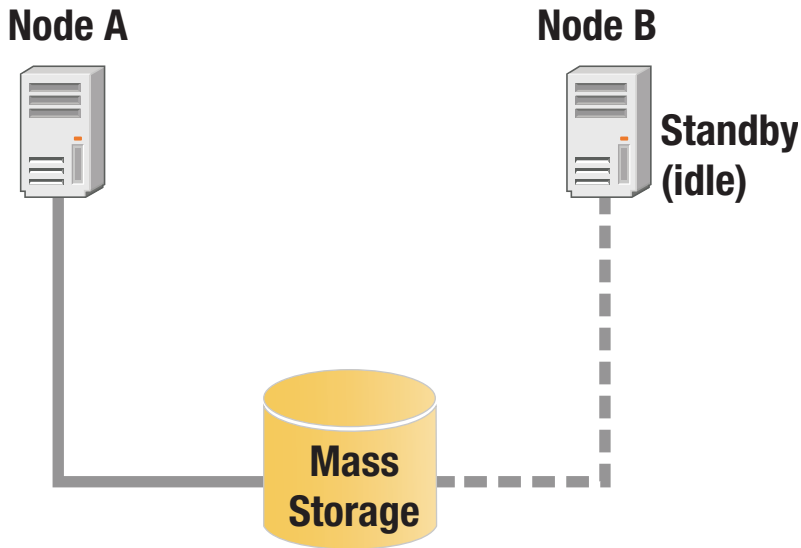


Figure 8. Active-Passive configuration in normal operation

After Node A fail over to Node B, the connection to the mass storage resource from Node B will be activated, and because Node A is unavailable, its connection to the mass storage resource is inactive. This is shown in Figure 9.

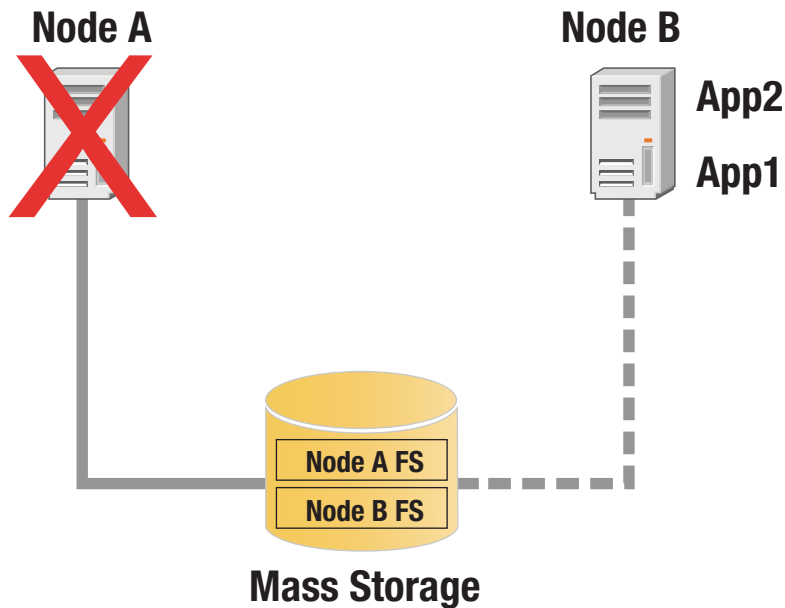


Figure 9. Failover on Active-Passive configuration

By contrast, in the following a takeover configuration, both Node A and Node B access the shared disk resource simultaneously. For IBM Workload Scheduler high-availability configurations, this usually means that the shared disk resource has separate, logical file system volumes, each accessed by a different node. This is illustrated in Figure 10 on page 43.

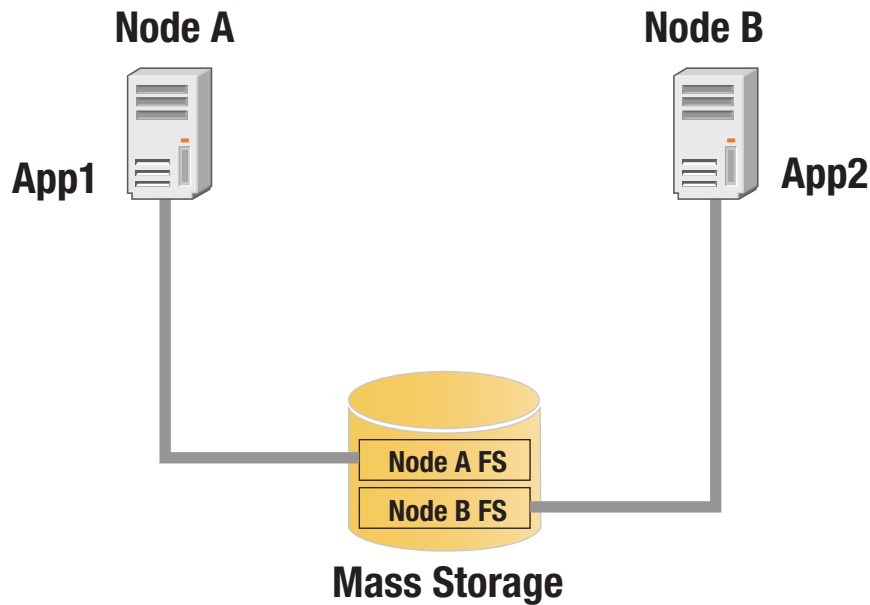


Figure 10. Logical file system volumes

During normal operation of this two-node highly available cluster in a takeover configuration, the filesystem Node A FS is accessed by App 1 on Node A, and the filesystem Node B FS is accessed by App 2 on Node B. If either node fails, the other node takes on the workload of the failed node. For example, if Node A fails, App 1 is restarted on Node B, and Node B opens a connection to filesystem Node A FS. This is illustrated in Figure 11.

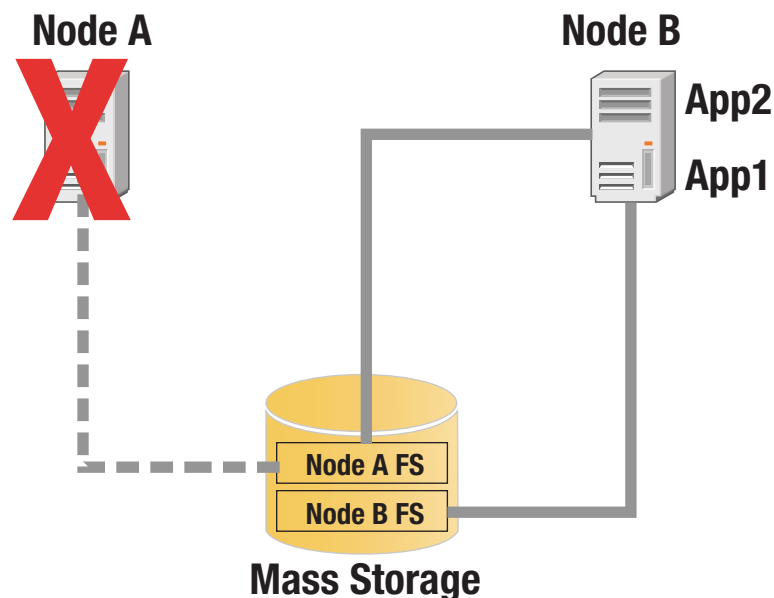


Figure 11. Failover scenario

Takeover configurations are more efficient than standby configurations with hardware resources because there are no idle nodes. Performance can degrade after a node failure, however, because the overall load on the remaining nodes increases.

## Design limitations

The following design limitations apply:

- “The master domain manager”
- “IBM Workload Scheduler commands”
- “Final status on running jobs”

### The master domain manager

The IBM Workload Scheduler master domain manager is supported on the Cluster Virtual Server, but has two important limitations:

- The master domain manager runs the Final job stream which creates a new plan for the current production day. This process cannot be interrupted. An interruption might cause malfunctions and scheduling service interruptions. Only manual steps can be used to recover from such malfunctions or service interruptions. Because failover of the cluster group that contains the IBM Workload Scheduler resource stops the agent on the current node and starts it on a different node, if failover happens when the Final job stream runs, could be destructive.
- The IBM Workload Scheduler command-line utilities (conman, composer, and so on) are unaware of the cluster and if they are interrupted (through a failover of the cluster group that contains the IBM Workload Scheduler resource) they might corrupt some vital IBM Workload Scheduler information.

### IBM Workload Scheduler commands

Any IBM Workload Scheduler command that is running during a failover is not automatically taken offline (unlike the main processes netman, mailman, batchman, and jobman) by the IBM Workload Scheduler cluster resource.

This is particularly problematical if the failover happens during an ad-hoc submission. The job submitted might remain in the **ADDING** state forever.

When Browse job log command via **conman** command line is active, the manual failover command is not working correctly and you must close all windows when the command is up and running.

### Final status on running jobs

If a job is running during failover, its final state is ABEND with return code zero. Because the Jobman process is unable to retrieve the true final state of the job.

## Supported configurations

This section describes the HACMP architecture that was set up for the test environment followed by in-depth scenario descriptions.

For the scenarios, the following resources were defined:

- 2 nodes
- 3 shared disks
- 1 volume group
- 1 application server
- 1 service IP address
- 1 resource group

The Heartbeat on was also configured on Disk.



The Application Server contains the definition of the **start\_tws.sh** and **stop\_tws.sh** scripts that are described in detail in each section and that must be created on both nodes.

The **start\_tws.sh** and **stop\_tws.sh** scripts are located in *TWA\_home/TWS/config* and you must customize them by setting the *DB2\_INST\_USER* parameter. After customizing the scripts, move them to another directory because any later release or fix pack overwrites them.

The Ethernet configuration we implemented is the IP Replacement, to have 1 boot address for each node and the Service IP address *replaces* the active one. In this configuration, the boot address of the active node can no longer be reached so, to avoid problems during the IBM Workload Scheduler installation, we configured an alias on the Ethernet network interface with the value of the boot address itself. Using the IP Aliasing configuration this additional step is unnecessary.

The following HACMP scenarios are supported with IBM Workload Scheduler:

- “Scenario: Shared disk, passive–active failover on a master domain manager”
- “Shared Disk, Passive – Active Failovers on Fault-Tolerant Agent” on page 47
- “Switching Domain Managers” on page 47

As an additional scenario we can also consider the possibility to have on the Master Domain Manager a local or a remote DB2® instance.

### **Scenario: Shared disk, passive–active failover on a master domain manager**

This scenario describes how to configure IBM Workload Scheduler and a remote or local DB2 database so that a HACMP cluster is able to manage the failover of the active master domain manager.

### **Configuring IBM Workload Scheduler and a remote DB2 database**

The following procedure explains how to configure IBM Workload Scheduler and a remote DB2 database so that a passive, idle node in the cluster can take over from an active master domain manager that has failed. The prerequisite for this procedure is that you have already configured HACMP.

Ensure that Installation Manager is installed in folders mounted on a shared disk and shared among all the cluster nodes, . If you do not change them during the installation, the default installation directories are:

```
/var/ibm/InstallationManager  
/opt/IBM/InstallationManager  
/opt/IBM/IMShared
```

Install IBM Workload Scheduler using one of the installation methods described in *Planning and Installation Guide*.

Install IBM Workload Scheduler using a shared Installation Manager instance, so that any patching, installing, uninstalling, or upgrading activity can be performed on any IBM Workload Scheduler node.

During the installation, perform the followings configuration steps:

1. Create the same *TWS administrator* user and group on all the nodes of the cluster. Ensure that the user has the same ID on all the nodes and points to the same home directory on the shared disk where you are going to install IBM Workload Scheduler.

**Example:** You want to create the group named *twsadm* for all IBM Workload Scheduler administrators and the TWS Administrator user named *twsusr* with user ID 518 and home */cluster/home/twsusr* on the shared disk:

```
mkgroup id=518 twsadm
mkuser id=518 pgrp=twsadm home=/cluster/home/twsusr twsusr
passwd twsusr
```

To install IBM Workload Scheduler in a directory other than the user home on the shared disk, ensure that the directory structure is the same on all nodes and that the *useropts* file is available to all nodes. Ensure also that the user has the same ID on all the nodes of the cluster.

2. Start the node that you want to use to run the installation of IBM Workload Scheduler and set the parameters so that HACMP mounts the shared disk automatically.
3. Install the DB2 administrative client on both nodes or on a shared disk configuring it for failover as described in DB2 documentation.
4. Create the *db2inst1* instance on the active node to create a direct link between IBM Workload Scheduler and the remote DB2 server.
5. Proceed with the IBM Workload Scheduler installation, using *twsuser* as the home directory and the local *db2inst1* instance.

After you installed IBM Workload Scheduler, run the cluster collector tool to automatically collect files from the active master domain manager. These files include the registry files, the Software Distribution catalog, and the IBM Workload Scheduler external libraries. The cluster collector tool creates a *.tar* file containing the collected files. To copy these files on the passive nodes, you must extract this *.tar* file on them.

To configure IBM Workload Scheduler for HACMP, perform the following steps:

1. Run the cluster collector tool.
2. From *TWA\_home/TWS/bin*, run `./twsClusterCollector.sh -collect -tarFileName tarFileName` where *tarFileName* is the complete path where the archive is stored.
3. Copy *tws\_user\_home/useropts\_twsuser* from the active node to the passive master domain manager from both the root and user home directories, to the other nodes.
4. Replace the node hostname with the service IP address for the master domain manager definitions, the WebSphere Application Server, the Dynamic workload broker and the agent, as described in the *Administration Guide*, section Changing the workstation host name or IP address.
5. Copy the **start\_tws.sh** and **stop\_tws.sh** scripts from *TWA\_home/TWS/config* to the *TWA\_home* directory.
6. Customize the **start\_tws.sh** and **stop\_tws.sh** scripts by setting the *DB2\_INST\_USER* parameter that is used to run the start and stop of the DB2 instance during the “failover” phase.
7. Try the **start\_tws.sh** and **stop\_tws.sh** scripts to verify IBM Workload Scheduler starts and stops correctly.

8. Move the shared volume on the second cluster node (if you have already defined the cluster group, you can move it by using the **c1RGmove HACMP** command).
9. Run the collector tool to extract IBM Workload Scheduler libraries. From the *TWA\_home/TWS/bin* directory, run:
 

```
./twsClusterCollector.sh -deploy -tarFileName tarFileName
```

where *tarFileName* is the complete path where the archive is stored.

10. Configure a new Application Controller resource on HACMP using the customized **start\_tws.sh** and **stop\_tws.sh** scripts.

When invoked by the HACMP during the failover, the scripts automatically start or stop the WebSphere Application Server and IBM Workload Scheduler, and link or unlink all the workstations.

## Local DB2

This scenario includes all of the steps described in “Configuring IBM Workload Scheduler and a remote DB2 database” on page 45 but, you must also perform the following additional steps:

1. Install the DB2 locally on both the nodes or on the shared disk, without creating a new instance.
2. Create a new instance on the shared disk, define all the DB2 users also on the second node, and modify the following two files:
  - */etc/hosts.equiv*  
Add a new line with just the Service IP address value.
  - *<db2-instance-home>/sql1lib/db2nodes.cfg*  
Add a new line similar to the following line:  
0 *<Service IP address>* 0
3. To stop the monman process used for Event Driven Workload Automation, add "conman startmon" and "conman stopman" to the *start\_tws.sh* and *stop\_tws.sh* scripts respectively.

## Shared Disk, Passive – Active Failovers on Fault-Tolerant Agent

This scenario is almost the same as “Scenario: Shared disk, passive–active failover on a master domain manager” on page 45, but there are no additional steps to perform on the DB2 and WebSphere® Application Server and the *start\_tws.sh* and *top\_tws.sh* scripts run just the link/unlink and start or stop commands.

## Switching Domain Managers

In this scenario, the DB2 database is installed on a remote server and the DB2 administration client is installed on both nodes. The configuration is based on a Master installation on the first node and a Backup Master on the second one. Both nodes are connected to the same DB2 remote server.

No additional post-installation steps are required. You can left the **stop\_tws.sh** script empty and create **start\_tws.sh** from *TWA\_home/TWS/config/switch\_tws.sh*. In **switch\_tws.sh**, you must set the *DB2\_INST\_USER*, *DB2\_INST\_PASSWD*, and *TWS\_DB* variables.

The **start\_tws.sh** script runs the switch manager command and, as an additional step, changes the workstation definition in DB2 in order to support more conveniently a switch that lasts longer than a production day.

## Upgrading from previous versions of the product

You can upgrade cluster nodes to the latest version of the product.

You can upgrade from:

- V9.1 or later

Perform the following actions:

- Read the system requirements.
- Read the *IBM Workload Scheduler: Planning and Installation*, chapter Upgrading.
- If you are upgrading a master domain manager or a backup master domain manager, perform a backup of the database.
- If you are upgrading a critical production node, perform a backup of the entire installation directory.
- Ensure that all the product processes have been shut down.
- Consider the following information that applies to the master domain manager, backup master domain manager, the domain manager, the backup domain manager, the dynamic agents, and the fault-tolerant agents:

### Using new features:

- The default authentication mechanism configured for the WebSphere Application Server included in the product is based on the Federated User Registry which supports the simultaneous use of more than one user registry. In earlier releases, the stand-alone user registry was used that can be local operating system, PAM, or LDAP. During a direct upgrade of a master domain manager or a backup master domain manager, the installation wizard attempts to reconfigure your authentication mechanism to use the Federated User Registry. If the reconfiguration fails, but all the other upgrade steps complete successfully, the upgraded master domain manager is configured to use a stand-alone user registry. This action is a temporary measure that allows you to access the master domain manager. Follow the instructions in the *IBM Workload Scheduler: Administration Guide*, section Configuring authentication to configure your authentication mechanism to use the Federated User Registry.
- The dynamic agent, if installed and enabled, listen for incoming requests from the master domain manager on a TCP/IP port different from the one used by netman. Ensure that this port is accessible and reachable after you upgrade. The new version of the product requires the Common Inventory Technology (CIT). This component is installed automatically during the upgrade. If you are configuring an hot standby HACMP cluster ensure that it is installed also on the passive node. To install CIT, use the command shown in **Files and components installed on the local disk**.

Note that, the dynamic agent runs jobs based on resource availability. In a cluster, for example, you can configure the workload to run on the node that is holding the resource at that time. In this case, to run the job where the service is located, you no longer need to configure the agent to fail over (to follow the service). You can design your workload according to a resource requirement, ensuring the logical resource that the jobs depend on is associated to the correct node every time the service fails over.

### Files and components installed on the local disk

- The product binary files depend on some files that are installed on the local file system in the following directories:

- /etc/TWS
- /etc/TWA
- /.swdis

This directory is the default Software Distribution directory. The product changes the directory specified in the **product\_dir** property in the /etc/Tivoli/swdis.ini file.

- /usr/Tivoli/TWS

These files must be replicated on the passive node. The product provides the `twClusterCollector.sh` utility to create a tar file with all external dependencies. Create a tar file and extract it on the passive node by using the `twClusterCollector.sh` utility.

## After upgrading

Instructions to follow after upgrading.

After you upgraded, perform the following actions:

- New features added to the product are enabled giving specific permissions to the users through the security file. During an upgrade, the product does not change your existing (customized) security file. You must modify the security file to include new security statements. See the sections about upgrading a master domain manager instance or its backup and upgrading a master domain manager or backup master domain manager instance in *IBM Workload Scheduler: Planning and Installation*.
- To configure the product to work with the HACMP service IP address, follow the instructions in the *IBM Workload Scheduler: Administration Guide*, section Changing the workstation host name or IP address. By default the installation wizard uses the node host name.
- To start and stop the product, by using the **start\_tws.sh** and **stop\_tws.sh** scripts in the directory `TWA_home/TWS/config`. Customize these scripts to satisfy your environment requirements.



---

## Appendix. Resolving desktop heap size problems on workstations with more than three agents

This appendix describes how to resolve the problem where the Windows desktop heap memory limitations cause processes to fail if there are more than three instances of IBM Workload Scheduler installed on a workstation in a Windows cluster environment.

Use this description whether you want to prevent the problem occurring (before installing the fourth agent instance) or if a problem has occurred caused by this limitation.

This section has the following topics

- “Problem description”
- “Solutions” on page 52
- “Implementing the solutions” on page 53

---

### Problem description

The problem occurs because of the way Windows handles its desktop heap memory, and the way IBM Workload Scheduler creates *desktops*. In the security context, a desktop is used to encapsulate Windows processes, preventing the process from performing unauthorized activities.

The total amount of memory available for the creation of desktops is determined by a Windows registry entry called:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager  
\Memory Management\SessionViewSize
```

The default value is 20MB.

The share of that buffer for each desktop is determined by a Windows registry entry called:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager  
\SubSystems\Windows
```

For example, the value of this entry might be:

```
%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=  
1024,3072,512 Windows=0n SubSystemType=Windows ServerDll=basesrv,  
1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:  
ConServerDllInitialization,2 ProfileControl=0ff MaxRequestThreads=16
```

In this entry, after the keyword *SharedSection*, there are three comma-separated memory entries (in KBs):

#### **Common memory (first entry)**

Defines the shared heap size common to all desktops (1024 in the example).

#### **Interactive desktop memory (second entry)**

Defines the extra desktop heap memory assigned to each interactive

process (3072 in the example). For example, the process which is in *foreground* at the moment. There are normally three interactive processes running at any one time.

**Non-interactive desktop memory (third entry)**

Defines the extra desktop memory assigned to non-interactive processes (512 in the example). For example, any process running in background.

IBM Workload Scheduler processes make the following use of desktops:

*IBM Workload Scheduler Netman* **Windows service**

Creates a non-interactive desktop shared between all agents running on the physical computer.

*IBM Workload Scheduler Token Service* **Windows service**

Creates a non-interactive desktop for the TWSUser for each agent.

*IBM Workload Scheduler* **Windows service**

Creates a non-interactive desktop for the TWSUser for each agent.

**Job manager (jobmon.exe)**

Creates a non-interactive desktop for all jobs launched by each agent.

Thus, for each extra agent, three non-interactive desktops are created. The problem occurs when Windows uses up all the memory for creating desktops.

---

## Solutions

### About this task

To reduce the risk that an IBM Workload Scheduler process cannot find sufficient memory to create a desktop, do one, or more, of the following:

- “Modify the shared heap buffer sizes”
- “Configure the IBM Workload Scheduler Windows service to start as a local system account” on page 53
- “Customize the desktop name so that it is reused” on page 53

## Modify the shared heap buffer sizes

### About this task

If you reduce the size of the common or interactive memory, you leave more memory available for non-interactive desktops. However, reducing the sizes of either of these might cause performance problems. Microsoft sets these values by default because their tests show that these are the required values. You are not recommended to change these values.

Reducing the memory used for a non-interactive desktop will allow more desktops to be created. Individual processes that require more memory might be impacted, but most processes will run successfully. If your default non-interactive desktop memory (third entry) is 512, try reducing it to 256. See “Modify the Windows registry entries that determine the heap size” on page 54 for how to do it.



## Configure the IBM Workload Scheduler Windows service to start as a local system account

### About this task

By default, the *IBM Workload Scheduler Windows* service is configured for the TWSUser of each agent. By changing it to start as a local system account, only one desktop instance is created on the computer, not one per agent. The solution is implemented as follows:

- For agents installed with version 9.4, or later, this change is achieved by using the optional installation parameter `-sharedDesktop`
- For agents being installed at earlier versions, or agents already installed, make the change manually. See “Modify configuration of Windows service” on page 54 for how to do it.

## Customize the desktop name so that it is reused

### About this task

When **Jobmon** opens a desktop, it allocates a unique name to the desktop, ensuring that a different desktop is created for each agent. However, if it creates the desktop using the name of a desktop already open, that process will open inside the existing desktop. To avoid this, you need to customize the name that Jobmon uses when it creates its desktop. By using the same name for all agents, each instance of **Jobmon** opens in the same desktop.

To ensure that this option is effective, the supplied name must be the same for at least two of the agents installed. The more agents that are run using the same shared desktop, the more memory will be available for desktop creation. However, if too many agents use the same shared desktop, there might be an impact on the ability of Windows to manage the jobs running in the shared desktop correctly. In this case, you might want to make a compromise. For example, if you had four agents installed on the same computer, you could choose to have pairs of agents share the same desktop.

The solution is implemented as follows:

- For agents installed with version 9.4, or later, this change is achieved by using the optional installation parameter `-sharedDesktop`. If you add this option without an argument, the installation applies the default name of `TWS_JOBS_WINSTA`. Otherwise supply your own name, for example, `-sharedDesktop name="my windows desktop name"`. See “twscClusterAdm command with examples of usage” on page 16 for how to do it.
- For agents being installed at earlier versions, or agents already installed, make the change manually. See “Modify localopts to supply a shared desktop name” on page 54 for how to do it.

---

## Implementing the solutions

### About this task

There are several possible solutions. Choose the one that is best for your circumstances:

- “Modify configuration of Windows service” on page 54
- “Modify the Windows registry entries that determine the heap size” on page 54

- “Modify localopts to supply a shared desktop name”

## Modify configuration of Windows service

### About this task

To modify the *IBM Workload Scheduler* Windows service to open as a local account, do the following:

1. From the **Start** button, select the **Services** panel (for example, select **Programs → Administrative Tools → Services**)
2. Select the **IBM Workload Scheduler** service and double-click it to edit it
3. Select the **Log on** tab
4. Click **Local System account** and then **Apply**
5. Right-click the service and select **Stop**
6. When the service has stopped, right-click it again and select **Start**
7. Check that the service has started correctly and close the Services window.

## Modify the Windows registry entries that determine the heap size

### About this task

To modify the Windows registry entries that determine the heap size, run **regedit.exe** and modify the key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager
\SubSystems\Windows
```

The default data for this registry value will look something like the following (all on one line):

```
%SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=
1024,3072,512 Windows=0n SubSystemType=Windows ServerDll=basesrv,
1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:
ConServerDllInitialization,2 ProfileControl=Off MaxRequestThreads=16
```

The numeric values following `SharedSection=` control how the desktop heap is allocated. These `SharedSection` values are specified in kilobytes. See “Problem description” on page 51 for a description of the values.

The third `SharedSection` value (512 in the above example) is the size of the desktop heap for each non-interactive desktop. Decrease the value to 256 kilobyte.

**Note:** Decreasing any of the `SharedSection` values will increase the number of desktops that can be created in the corresponding window stations. Smaller values will limit the number of hooks, menus, strings, and windows that can be created within a desktop. On the other hand, increasing the `SharedSection` values will decrease the number of desktops that can be created, but will increase the number of hooks, menus, strings, and windows that can be created within a desktop. This change will only take effect after you reboot the cluster nodes.

## Modify localopts to supply a shared desktop name

### About this task

To use a shared desktop name for an agent already installed, do the following:

1. Open the localopts file for the agent in question. For the location of this file, see *Planning and Installation Guide*.
2. Add the key jm windows station name = *<my\_name>*. Ensure that *<my\_name>* is the same name as used in another agent to save desktop memory.
3. Save the file.
4. Stop and restart IBM Workload Scheduler to make the change effective.



---

## Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

For license inquiries regarding double-byte character set (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*

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide 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 Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

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.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

© (your company name) (year).  
Portions of this code are derived from IBM Corp. Sample Programs.  
© Copyright IBM Corp. \_enter the year or years\_.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.



Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

ITIL is a Registered Trade Mark of AXELOS Limited.

UNIX is a registered trademark of The Open Group in the United States and other countries.

---

## Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

## **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

## **Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

## **Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## **Rights**

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



---

# Index

## Special characters

-ask, twsClusterAdm.exe argument 22  
-changeResName, twsClusterAdm.exe parameter 22  
-dll, twsClusterAdm.exe parameter 21  
-force, twsClusterAdm.exe argument 22  
-force, twsClusterAdm.exe parameter 21  
-new, twsClusterAdm.exe parameter 18  
-res, twsClusterAdm.exe parameter 19  
-sharedDesktop, twsClusterAdm.exe parameter 21  
-twsupd, twsClusterAdm.exe argument 22  
-uninst, twsClusterAdm.exe parameter 21  
-update, twsClusterAdm.exe parameter 22

## A

abend state 34  
accessibility ix  
agent, installing in a cluster-aware configuration 12  
arguments, to twsClusterAdm.exe 18  
ask, twsClusterAdm.exe argument 22

## B

backup domain manager  
IBM Workload Scheduler 1

## C

changeResName, twsClusterAdm.exe parameter 22  
check\_interval, twsClusterAdm.exe argument 19  
Cloud & Smarter Infrastructure technical training x  
clu\_offline.log 33  
clu\_online.log 33  
Cluster Administrator console, using 31  
Cluster Administrator extension  
installing 32  
overview 31  
Parameters tab 32  
cluster HACMP upgrading  
to version 8.6 49  
to version 9.x 48  
cluster instance name  
local option 30  
modifying  
in the localopts file 31  
in the ShutDown\_clu.cmd script 31  
in the StartUp\_clu.cmd script 31  
cluster resource dependencies 29  
IP Address 29

cluster resource dependencies (*continued*)  
Network Name 29  
Physical Disk 29  
cluster.log 34  
commands  
cluster res 30  
ShutDown\_clu.cmd 30  
Shutdown.cmd 13  
StartUp\_clu.cmd 30  
tws\_env.cmd 13  
twsClusterAdm.exe 16  
twsClusterAdmin.exe, local option 30  
compatibility  
with previous versions 6  
configuration 41  
Conman shut 30  
Conman start 30  
contents, Windows Cluster Enabler 9  
core dump 35  
custom resource DLL 8  
entry-points  
IsAlive 8  
Offline 8  
Online 8  
Terminate 8

## D

disk, twsClusterAdm.exe argument 19  
DLL  
custom resource DLL 8  
DLL files  
ITWSResources.dll 9  
ITWSWorkstationEx.dll 9  
dll, twsClusterAdm.exe parameter 21  
domain, twsClusterAdm.exe argument 18, 21  
Dynamic Workload Console  
accessibility ix

## E

education x  
error 1314 34  
examples  
twsClusterAdm.exe 22

## F

failover, twsClusterAdm.exe argument 19  
fault-tolerance  
definition 1  
force, twsClusterAdm.exe argument 22  
force, twsClusterAdm.exe parameter 21

## G

group, twsClusterAdm.exe argument 19

## H

HACMP cluster upgrading  
to version 8.6 49  
high availability  
types of 1  
high-availability  
definition 1  
hosts, twsClusterAdm.exe argument 18, 21, 22

## I

IBM Workload Scheduler 40, 41, 44  
authorization 7  
backup domain manager 1  
benefits 37  
clients 40  
Cluster Administrator extension 31  
installing 32  
overview 31  
cluster environment  
integrating into 5  
cluster resource dependencies 29  
command-line  
not automatically taken offline  
during a failover 6, 44  
compatibility  
with previous versions 6  
configuring  
with twsClusterAdm 16  
HACMP 37, 38, 39  
in the Windows cluster environments 29  
integrating into  
cluster environment 5  
make cluster-aware 13  
master domain manager 44  
not supported on Cluster Virtual Server 6  
networks 40  
nodes 39  
physical components 38  
product design limitations 5, 44  
resource instance name  
changing 31  
security authentication 7  
shared external disk 40  
standby 41  
starting 30  
cluster res command 30  
stopping 30  
cluster res command 30  
troubleshooting 33  
abend state 34  
core dump 35  
error 1314 34

- IBM Workload Scheduler (*continued*)
  - troubleshooting (*continued*)
    - trace files 33
  - uninstalling 32
  - UNIX 37
  - where to find information 5, 41
  - Windows 5
  - Windows Cluster Enabler 7
    - components 7
    - prerequisites 6

- IBM Workload Scheduler agent
  - installing
    - in a cluster-aware configuration 12
- IBM Workload Scheduler agent from V9.1
  - and fix packs
    - upgrading
      - in a cluster-aware configuration with twsClusterUpg 14
  - installation
    - installing in a cluster 9
    - prerequisites 11
  - installing
    - Cluster Administrator extension 32
    - IBM Workload Scheduler agent 12
    - Windows Cluster Enabler 9
  - installing in a cluster 9
  - IP Address 29
  - ip, twsClusterAdm.exe argument 19
  - Isalive, twsClusterAdm.exe argument 20
  - ITWSExInst.cmd 9
  - ITWSResources.dll 9
  - ITWSWorkstationEx.dll 9

## L

- localopts file
  - modifying 31
- lookalive, twsClusterAdm.exe
  - argument 19

## M

- master domain manager 44
  - not supported on Cluster Virtual Server
    - reasons why 6

## N

- name, twsClusterAdm.exe argument 21
- net, twsClusterAdm.exe argument 19
- Network Name 29
- new\_resource\_instance\_name, twsClusterAdm.exe argument 22
- new, twsClusterAdm.exe parameter 18

## O

- operating systems 6
- opts, twsClusterAdm.exe argument 20

## P

- Parameters tab
  - Cluster Administrator extension 32
- parameters, to twsClusterAdm.exe 18
- path, twsClusterAdm.exe argument 21
- Physical Disk 29
- prerequisites 11
- pwd, twsClusterAdm.exe argument 18

## R

- recovery action in case of failover 1
  - allowing for 1
- Replace a process level token
  - security policy 12, 34
- res, twsClusterAdm.exe parameter 19
- rename, twsClusterAdm.exe argument 19
- resource instance name
  - changing 31
- resource\_instance\_name, twsClusterAdm.exe argument 22
- resource, twsClusterAdm.exe argument 22
- roll back or recovery action 1
  - allowing for 1

## S

- scripts
  - ShutDown\_clu.cmd 30
  - StartUp\_clu.cmd 30
- security policy
  - Replace a process level token 12, 34
- sharedDesktop, twsClusterAdm.exe
  - parameter 21
- ShutDown\_clu.cmd 30
- Shutdown.cmd 13, 30
- ShutdownLwa.cmd 30
- StartUp\_clu.cmd 30
- Startup.cmd 30
- StartupLwa.cmd 30
- syntax
  - twsClusterAdm.exe 17

## T

- takeover 41
- tcpport, twsClusterAdm.exe
  - argument 20
- technical training x
  - to version 9.x
    - cluster HACMP upgrading to 48
- trace files 33
  - clu\_offline.log 33
  - clu\_online.log 33
  - cluster.log 34
- training
  - technical x
- troubleshooting 33
  - abend state 34
  - error 1314 34
  - Jobmon
    - core dump 35
    - trace files 33

- tws\_env.cmd 13
- twsClusterAdm
  - IBM Workload Scheduler agent 16
- twsClusterAdm.exe 9, 13, 16
  - arguments 18
  - examples 22
  - parameters 18
  - syntax 17
- twsClusterAdmin.exe
  - cluster instance name
    - local option 30
- twsClusterUpg syntax 14
- twshome, twsClusterAdm.exe
  - argument 19
- twsupd, twsClusterAdm.exe
  - argument 22

## U

- uninst, twsClusterAdm.exe
  - parameter 21
- UNIX 37
  - where to find information 41
- update, twsClusterAdm.exe
  - parameter 22
- upgrading
  - Workload Scheduler agent 15
- upgrading in HACMP cluster
  - to version 8.6 49
  - to version 9.x 48
- upgrading manually 15
- upgrading with twsClusterUpg
  - IBM Workload Scheduler agent from V9.1, V9.2 and their fix packs 14
  - IBM Workload Scheduler agent from V9.2 and fix packs 14
- user, twsClusterAdm.exe argument 18, 21
- utility
  - Shutdown.cmd 13

## V

- version 8.6
  - cluster HACMP upgrading to 49

## W

- Windows 5
- Windows Cluster Enabler
  - components 7
    - custom resource DLL 8
  - contents 9
    - ITWSExInst.cmd 9
    - ITWSResources.dll 9
    - ITWSWorkstationEx.dll 9
    - twsClusterAdm.exe 9
  - installing 9
  - installing in a cluster 9
  - prerequisites 11
- Windows clusters
  - where to find information 5
- Workload Scheduler agent
  - upgrading manually
    - in a cluster-aware configuration 15





Product Number: 5698-WSH

Printed in USA