

# MATLAB® Distributed Computing Server™

## Installation Guide

**R2012a**

MATLAB®

## How to Contact MathWorks



[www.mathworks.com](http://www.mathworks.com)  
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab@mathworks.com)  
[www.mathworks.com/contact\\_TS.html](http://www.mathworks.com/contact_TS.html)

Web  
Newsgroup  
Technical Support



[suggest@mathworks.com](mailto:suggest@mathworks.com)  
[bugs@mathworks.com](mailto:bugs@mathworks.com)  
[doc@mathworks.com](mailto:doc@mathworks.com)  
[service@mathworks.com](mailto:service@mathworks.com)  
[info@mathworks.com](mailto:info@mathworks.com)

Product enhancement suggestions  
Bug reports  
Documentation error reports  
Order status, license renewals, passcodes  
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.  
3 Apple Hill Drive  
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

*MATLAB® Distributed Computing Server™ Installation Guide*

© COPYRIGHT 2005–2012 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

### Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

### Patents

MathWorks products are protected by one or more U.S. patents. Please see [www.mathworks.com/patents](http://www.mathworks.com/patents) for more information.

### Revision History

April 2011	Online only	Revised for version 5.1 (Release 2011a)
September 2011	Online only	Revised for version 5.2 (Release 2011b)
March 2012	Online only	Revised for version 6.0 (Release 2012a)

## Product Installation

**1**

<b>Cluster Description</b> .....	<b>1-2</b>
<b>Installing Products</b> .....	<b>1-4</b>
On the Cluster Nodes .....	<b>1-4</b>
On the Client Nodes .....	<b>1-4</b>
<b>Configuring Your Cluster</b> .....	<b>1-5</b>

## Configuring Parallel Computing Products for an MJS

**2**

<b>Configure Cluster to Use a MATLAB Job Scheduler (MJS)</b> .....	<b>2-2</b>
Step 1: Set Up Windows Cluster Hosts .....	<b>2-2</b>
Step 2: Stop mdce Services of Old Installation .....	<b>2-4</b>
Step 3: Start the mdce Service, MJS, and Workers .....	<b>2-5</b>
Step 4: Install the mdce Service to Start Automatically at Boot Time (UNIX) .....	<b>2-18</b>
<b>Configure Windows Firewalls on Client</b> .....	<b>2-22</b>
<b>Validate Installation with MJS</b> .....	<b>2-23</b>
Step 1: Verify the Cluster Connection .....	<b>2-23</b>
Step 2: Define a Cluster Profile .....	<b>2-23</b>
Step 3: Validate the Cluster Profile .....	<b>2-24</b>

## Configuring Parallel Computing Products for HPC Server

### 3

<b>Configure Cluster for Microsoft Windows HPC Server</b> .....	<b>3-2</b>
<b>Configure Client Computer for HPC Server 2008</b> .....	<b>3-3</b>
<b>Validate Installation Using Microsoft Windows HPC Server</b> .....	<b>3-4</b>
Step 1: Define a Cluster Profile .....	<b>3-4</b>
Step 2: Validate the Configuration .....	<b>3-5</b>

## Configuring Parallel Computing Products for Supported Third-Party Schedulers (PBS Pro, Platform LSF, TORQUE)

### 4

<b>Configure Platform LSF Scheduler on Windows Cluster</b> .....	<b>4-2</b>
Without Delegation .....	<b>4-2</b>
Using Passwordless Delegation .....	<b>4-4</b>
<b>Configure Windows Firewalls on Client</b> .....	<b>4-5</b>
<b>Validate Installation Using an LSF, PBS Pro, or TORQUE Scheduler</b> .....	<b>4-6</b>
Step 1: Define a Cluster Profile .....	<b>4-6</b>
Step 2: Validate the Cluster Profile .....	<b>4-7</b>

# Configuring Parallel Computing Products for a Generic Scheduler

## 5

<b>Interfacing with Generic Schedulers</b> .....	<b>5-2</b>
Support Scripts .....	<b>5-2</b>
Submission Mode .....	<b>5-2</b>
 <b>Configure Generic Scheduler on Windows Cluster</b> ....	<b>5-4</b>
Without Delegation .....	<b>5-4</b>
Using Passwordless Delegation .....	<b>5-6</b>
 <b>Configure Sun Grid Engine on Linux Cluster</b> .....	<b>5-7</b>
Create the Parallel Environment .....	<b>5-7</b>
 <b>Configure Windows Firewalls on Client</b> .....	<b>5-9</b>
 <b>Validate Installation Using a Generic Scheduler</b> .....	<b>5-10</b>
Example Setup for LSF .....	<b>5-10</b>
Step 1: Set Up Windows Client Host .....	<b>5-12</b>
Step 2: Define a Cluster Profile .....	<b>5-12</b>
Step 3: Validate Cluster Profile .....	<b>5-14</b>



# Product Installation

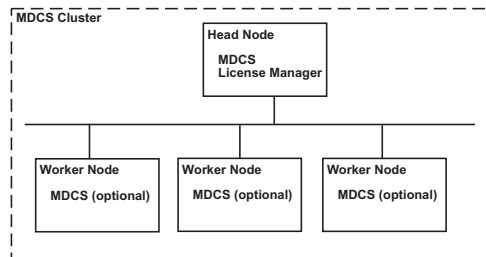
---

- “Cluster Description” on page 1-2
- “Installing Products” on page 1-4
- “Configuring Your Cluster” on page 1-5

## Cluster Description

To set up a cluster, you first install MATLAB® Distributed Computing Server™ (MDCS) on a node called the *head* node. You can also install the license manager on the head node. After performing this installation, you can then optionally install MDCS on the individual cluster nodes, called *worker* nodes. You do not need to install the license manager on worker nodes.

This figure shows the installations that you perform on your MDCS cluster nodes. This is only one possible configuration. (You can install the cluster license manager and MDCS on separate nodes, but this document does not cover this type of installation.)

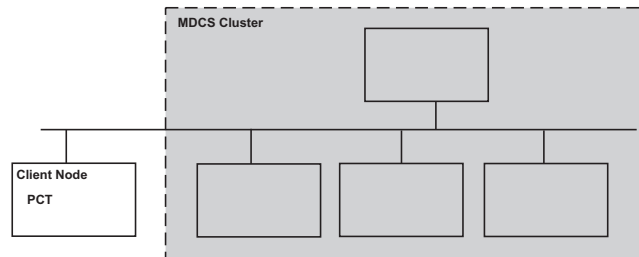


### Product Installations on Cluster Nodes

You install Parallel Computing Toolbox™ (PCT) software on the computer that you use to write MATLAB applications. This is called the *client* node.

This figure shows the installations that you must perform on client nodes.





### Product Installations on Client Nodes

## Installing Products

### On the Cluster Nodes

Install the MathWorks products on your cluster as a network installation according to the instructions found at

<http://www.mathworks.com/help/base/install/>

These instructions include steps for installing, licensing, and activating your installation.

You can install in a central location, or individually on each cluster node.

---

**Note** MathWorks highly recommends installing all MathWorks products on the cluster. MDCS cannot run jobs whose code requires products that are not installed.

---

### On the Client Nodes

On the client computers from which you will write applications to submit jobs to the cluster, install the MathWorks products for which you are licensed, including Parallel Computing Toolbox. You can find installation instructions at

<http://www.mathworks.com/help/base/install/>

These instructions include steps for installing, licensing, and activating your installation.

## Configuring Your Cluster

When the cluster and client installations are complete, you can proceed to configure the products for the job scheduler of your choice. Use one of the following chapters in this document to complete your configuration and to test the installation:

- Chapter 2, “Configuring Parallel Computing Products for an MJS”
- Chapter 3, “Configuring Parallel Computing Products for HPC Server”
- Chapter 4, “Configuring Parallel Computing Products for Supported Third-Party Schedulers (PBS Pro, Platform LSF, TORQUE)”
- Chapter 5, “Configuring Parallel Computing Products for a Generic Scheduler”

---

**Note** You must use the generic scheduler interface for any of the following:

- Any third-party schedule not listed above (e.g., Sun Grid Engine, GridMP, etc.)
  - PBS other than PBS Pro
  - A nonshared file system when the client cannot directly submit to the scheduler (e.g., TORQUE on Windows)
-



# Configuring Parallel Computing Products for an MJS

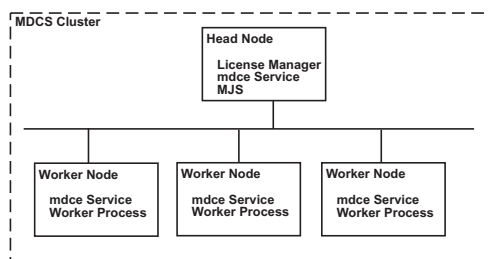
---

- “Configure Cluster to Use a MATLAB Job Scheduler (MJS)” on page 2-2
- “Configure Windows Firewalls on Client” on page 2-22
- “Validate Installation with MJS” on page 2-23

## Configure Cluster to Use a MATLAB Job Scheduler (MJS)

The mdce service must be running on all machines being used for MATLAB job schedulers (MJS) or workers. This service manages the MJS and worker processes. One of the major tasks of the mdce service is to recover the MJS and worker sessions after a system crash, so that jobs and tasks are not lost as a result of such accidents.

The following figure shows the processes that run on your cluster nodes.




---

**Note** The MATLAB job scheduler (MJS) was formerly known as the MathWorks job manager. The process is the same, is started in the same way, and performs the same functions.

---

In the following instructions, *matlabroot* refers to the location of your installed MATLAB Distributed Computing Server software. Where you see this term used in the instructions that follow, substitute the path to your location.

### Step 1: Set Up Windows Cluster Hosts

If this is the first installation of MATLAB Distributed Computing Server on a cluster of Windows machines, you need to configure these hosts for job communications.

---

**Note** If you do not have a Windows cluster, or if you have already installed a previous version of MATLAB Distributed Computing Server on your Windows cluster, you can skip this step and proceed to Step 2.

---

## Configure Windows Firewalls

If you are using Windows® firewalls on your cluster nodes,

**1** Log in as a user with administrator privileges.

**2** Execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them for similar accommodation.

## Configure User Access to Installation

The user that mdce runs as requires access to the cluster MATLAB installation location. By default, mdce runs as the user LocalSystem. If your network allows LocalSystem to access the install location, you can proceed to the next step. (If you are not sure of your network configuration and the access provided for LocalSystem, contact the MathWorks install support team.)

---

**Note** If LocalSystem cannot access the install location, you must run mdce as a different user.

---

You can set a different user with these steps:

**1** With any standard text editor (such as WordPad) open the mdce\_def file found at:

```
matlabroot\toolbox\distcomp\bin\mdce_def.bat
```

**2** Find the line for setting the MDCEUSER parameter, and provide a value in the form domain\username:

```
set MDCEUSER=mydomain\myusername
```

- 3 Provide the user password by setting the MDCEPASS parameter:

```
set MDCEPASS=password
```

- 4 Save the file. Proceed to the next step.

### Step 2: Stop mdce Services of Old Installation

If you have an older version of MATLAB Distributed Computing Server running on your cluster nodes, you should stop the mdce services before starting the services of the new installation.

- “Stop mdce on Windows” on page 2-4
- “Stop mdce on UNIX” on page 2-5

#### Stop mdce on Windows

If this is your first installation of the parallel computing products, proceed to “Step 3: Start the mdce Service, MJS, and Workers” on page 2-5.

- 1 Open a DOS command window with the necessary privileges:
  - a If you are using Windows 7 or Windows Vista™, you must run the command window with administrator privileges. Click the Windows menu **Start > (All) Programs > Accessories**; then right-click **Command Window**, and select **Run as Administrator**. This option is available only if you are running User Account Control (UAC).
  - b If you are using Windows XP, open a DOS command window by selecting the Windows menu **Start > Run**, then in the **Open** field, type

```
cmd
```

- 2 In the command window, navigate to the folder of the old installation that contains the control scripts.

```
cd oldmatlabroot\toolbox\distcomp\bin
```

- 3 Stop and uninstall the old mdce service and remove its associated files by typing the command:



```
mdce uninstall -clean
```

---

**Note** Using the `-clean` flag permanently removes all existing job data. Be sure this data is no longer needed before removing it.

---

**4** Repeat the instructions of this step on all worker nodes.

## Stop mdce on UNIX

**1** Log in as root. (If you cannot log in as root, you must alter the following parameters in the `matlabroot/toolbox/distcomp/bin/mdce_def.sh` file to point to a folder for which you have write privileges: `CHECKPOINTBASE`, `LOGBASE`, `PIDBASE`, and `LOCKBASE` if applicable.)

**2** On each cluster node, stop the mdce service and remove its associated files by typing the commands:

```
cd oldmatlabroot/toolbox/distcomp/bin
./mdce stop -clean
```

---

**Note** Using the `-clean` flag permanently removes all existing job data. Be sure this data is no longer needed before removing it.

---

## Step 3: Start the mdce Service, MJS, and Workers

You can start the MJS (job manager) by using a GUI or the command line. Choose one:

- “Using Admin Center GUI” on page 2-6
- “Using the Command-Line Interface (Windows)” on page 2-13
- “Using the Command-Line Interface (UNIX)” on page 2-16

### Using Admin Center GUI

---

**Note** To use Admin Center, you must run it on a computer that has direct network connectivity to all the nodes of your cluster. If you cannot run Admin Center on such a computer, follow the instructions in “Using the Command-Line Interface (Windows)” on page 2-13 or “Using the Command-Line Interface (UNIX)” on page 2-16.

---

#### Identify Hosts and Start the mdce Service.

- 1 To open Admin Center, navigate to the folder:

`matlabroot\toolbox\distcomp\bin` ( on Windows)

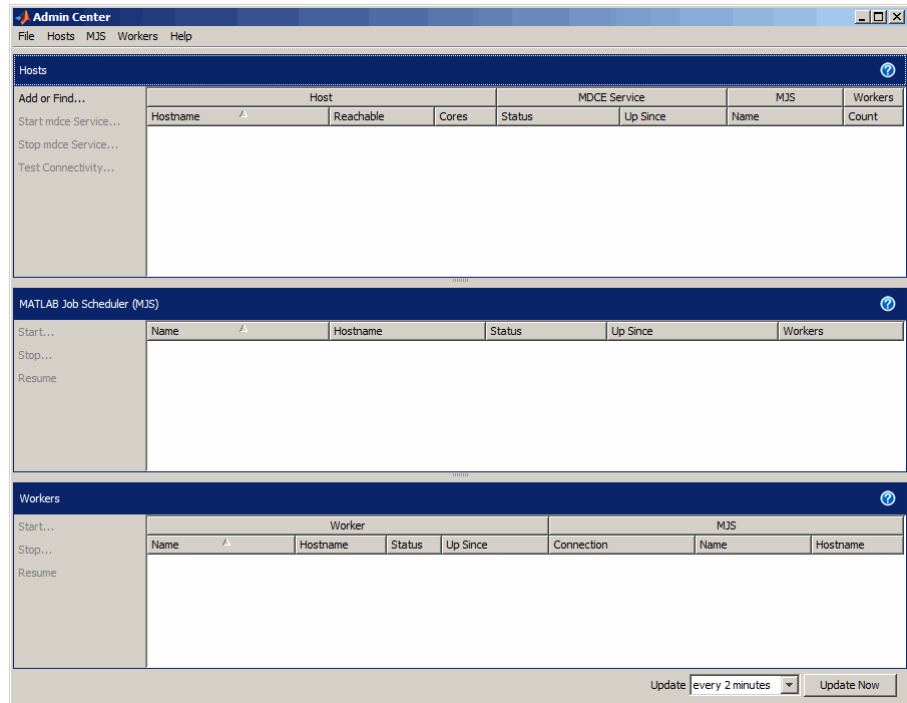
`matlabroot/toolbox/distcomp/bin` ( on UNIX)

Then execute the file:

`admincenter.bat` (on Windows)

`admincenter` (on UNIX)

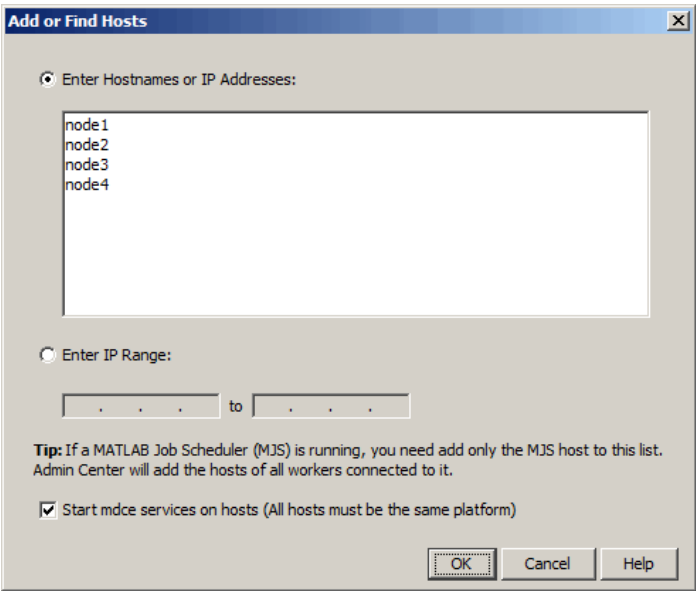
If there are no past sessions of Admin Center saved for you, the GUI opens with a blank listing, superimposed by a welcome dialog box, which provides information on how to get started.



## 2 Click Add or Find.

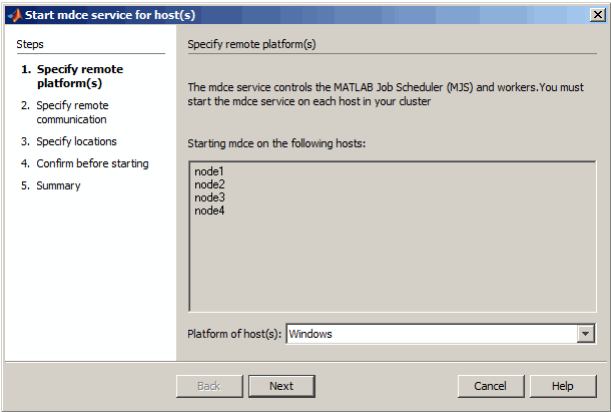
The Add or Find Hosts dialog box opens.

- 3 Select **Enter Hostnames**, then list your hosts in the text box. You can use short host names, fully qualified domain names, or individual IP addresses. The following figure shows an example using host names `node1`, `node2`, `node3`, and `node4`. In your case, use your own host names.

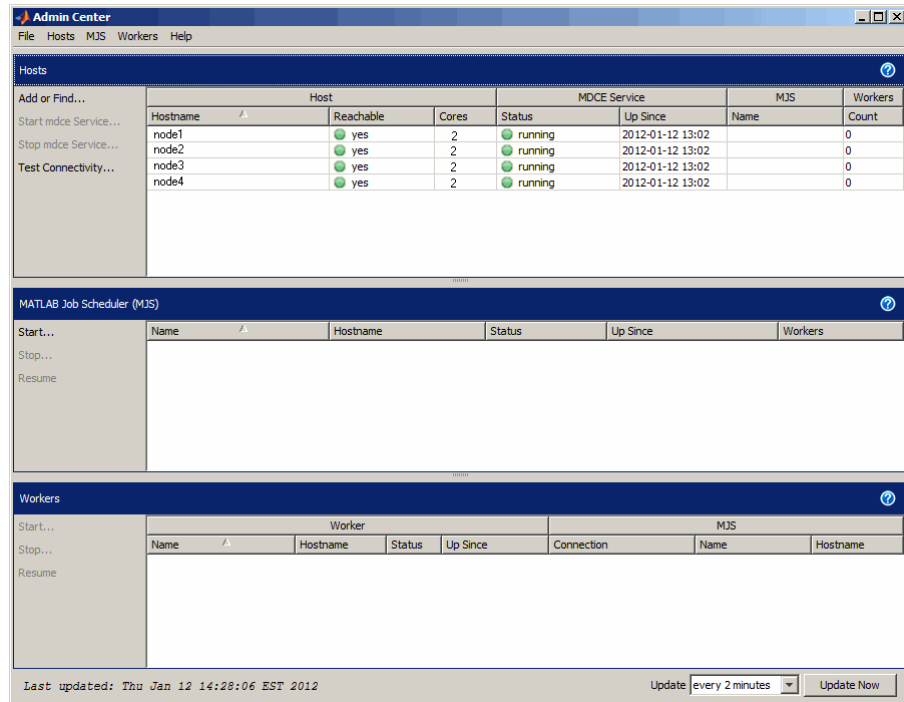


Keep the check to start mdce service.

- 4 Click **OK** to open the Start mdce service dialog box. Proceed through the steps clicking **Next** and checking the settings at each step. For most settings, the default is appropriate.



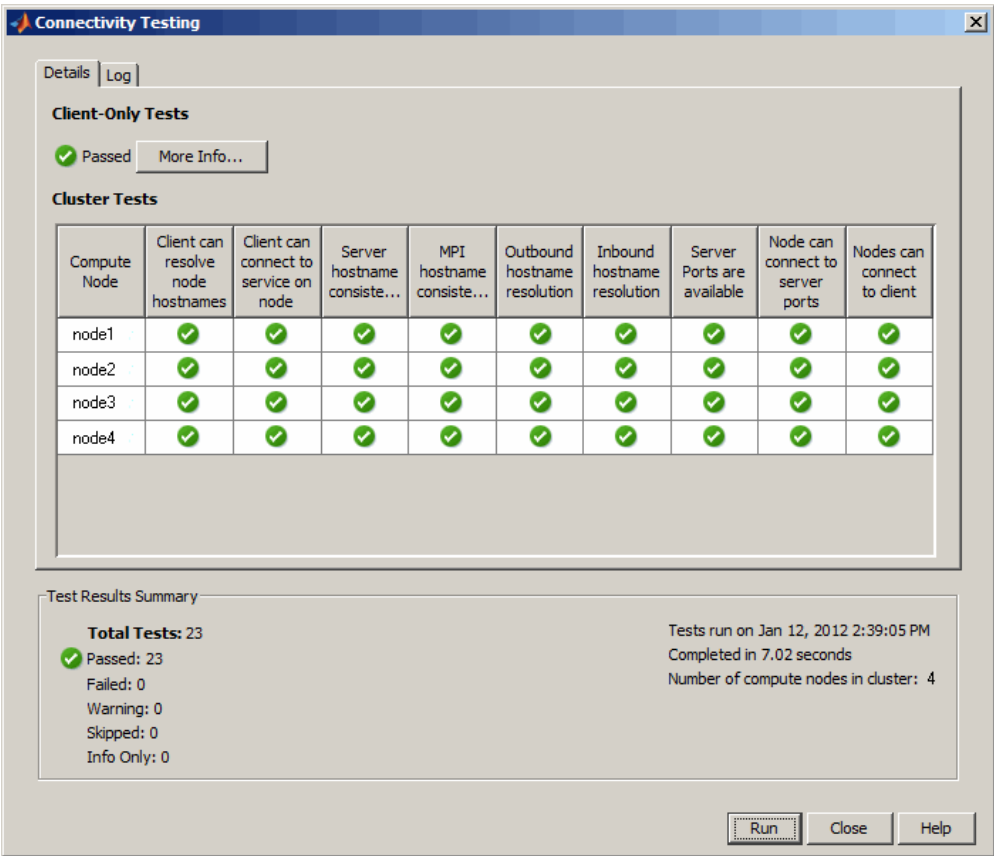
It might take a moment for Admin Center to communicate with all the nodes, start the services, and acquire the status of all of them. When Admin Center completes the update, the listing should look something like the following figure.



- 5 At this point, you should test the connectivity between the nodes. This assures that your cluster can perform the necessary communications for running other MCDS processes.

In the Hosts module, click **Test Connectivity**.

- 6 When the Connectivity Testing dialog box opens, it shows the results of the last test, if there are any. Click **Run** to run the tests and generate new data.

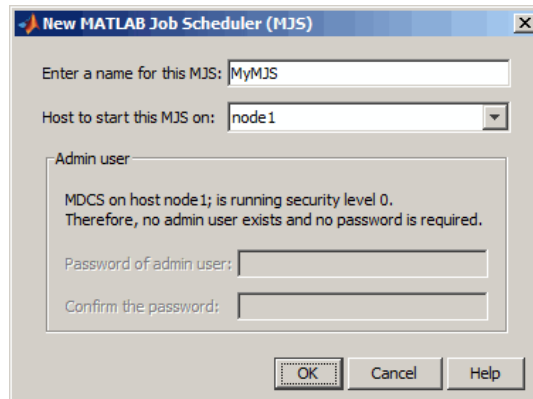


If any of the connectivity tests fail, double-click the icon that indicates a failure to get information about that specific test; or use the **Log** tab to get all test results. With this information, you can refer to the troubleshooting section of the *MATLAB Distributed Computing Server System Administrator's Guide*. If you need further help, contact the MathWorks install support team.

7 If your tests pass, click **Close** to return to the Admin Center GUI.

## Start the MJS.

- 1 To start an MJS (job manager), click **Start** in the MJS module. (This is one of several ways to open the New MJS dialog box.) In the New MJS dialog box, specify a name and host for your MJS. This example shows an MJS called MyMJS to run on host node1.

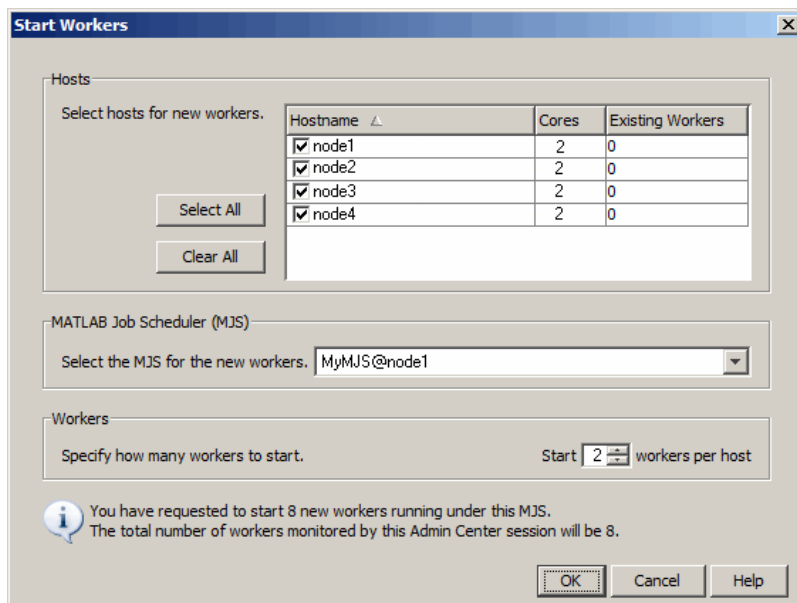


- 2 Click **OK** to start the MJS and return to the Admin Center GUI.

## Start the Workers.

- 1 To start workers, click **Start** in the Workers module. (This is one of several ways to open the Start Workers dialog box.)
- 2 In the Start Workers dialog box, specify the number of workers to start on each host. The number is up to you, but you cannot exceed the total number of licenses you have. A good starting value might be to start one worker per computational core on your hosts.
- 3 Select the hosts to start the workers on. Click **Select All** if you want to start workers on all listed hosts.
- 4 Select the MJS for your workers. If you have only one MJS running in this Admin Center session, that is the default.

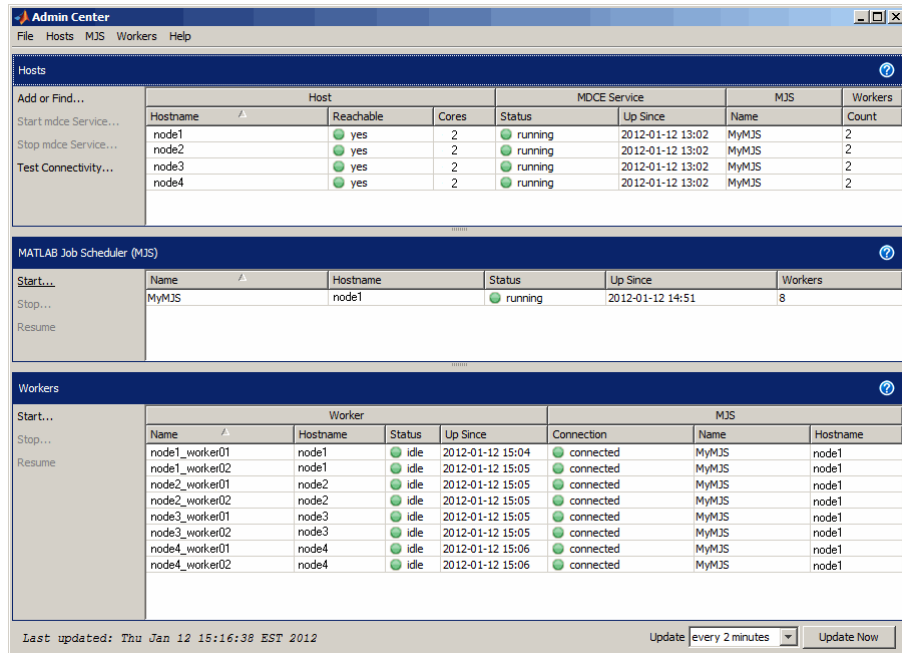
The following example shows a setup for starting eight workers on four hosts (two workers each). Your names and numbers will vary.



- 5 Click OK to start the workers and return to the Admin Center dialog box. It might take a moment for Admin Center to initialize all the workers and acquire their status.

When all the workers are started, Admin Center looks something like the following figure. If your workers are all idle and connected, your cluster is ready for use.





If you encounter any problems or failures, contact the MathWorks install support team.

For more information about Admin Center functionality, such as stopping processes or saving sessions, see the “Admin Center” chapter in the *MATLAB Distributed Computing Server System Administrator’s Guide*.

## Using the Command-Line Interface (Windows)

**Start the mdce Service.** You must install the mdce service on all nodes (head node and worker nodes). Begin on the head node.

- 1 Open a DOS command window with the necessary privileges:
  - If you are using Windows 7 or Windows Vista, you must run the command window with administrator privileges. Click the Windows menu **Start > (All) Programs > Accessories**; then right-click

**Command Window**, and select **Run as Administrator**. This option is available only if you are running User Account Control (UAC).

- b** If you are using Windows XP, open a DOS command window by selecting the Windows menu **Start > Run**, then in the **Open** field, type:

```
cmd
```

- 2** In the DOS command window, navigate to the folder with the control scripts:

```
cd matlabroot\toolbox\distcomp\bin
```

- 3** Install the mdce service by typing the command:

```
mdce install
```

- 4** Start the mdce service by typing the command:

```
mdce start
```

- 5** Repeat the instructions of this step on all worker nodes.

As an alternative to items 3–5, you can install and start the mdce service on several nodes remotely from one machine by typing:

```
cd matlabroot\toolbox\distcomp\bin
remotemdce install -remotehost hostA,hostB,hostC . . .
remotemdce start -remotehost hostA,hostB,hostC . . .
```

where `hostA,hostB,hostC` refers to a list of your host names. Note that there are no spaces between host names, only a comma. If you need to indicate protocol, platform (such as in a mixed environment), or other information, see the help for `remotemdce` by typing:

```
remotemdce -help
```

Once installed, the mdce service starts running each time the machine reboots. The mdce service continues to run until explicitly stopped or uninstalled, regardless of whether an MJS or worker session is running.

**Start the MJS.** To start the MATLAB job scheduler (MJS), enter the following commands in a DOS command window. You do not have to be at the machine on which the MJS runs, as long as you have access to the MDCS installation.

- 1 In your DOS command window, navigate to the folder with the startup scripts:

```
cd matlabroot\toolbox\distcomp\bin
```

- 2 Start the MJS, using any unique text you want for the name <MyMJS>:

```
startjobmanager -name <MyMJS> -remotehost <MJS host name> -v
```

- 3 Verify that the MJS is running on the intended host.

```
nodestatus -remotehost <MJS host name>
```

---

**Note** If you are executing `startjobmanager` on the host where the MJS runs, you do not need to specify the `-remotehost` flag.

If you have more than one MJS on your cluster, each must have a unique name.

---

## Start the Workers.

---

**Note** Before you can start a worker on a machine, the `mdce` service must already be running on that machine, and the license manager for MATLAB Distributed Computing Server must be running on the network.

---

For each node used as a worker, enter the following commands in a DOS command window. You do not have to be at the machines where the MATLAB workers will run, as long as you have access to the MDCS installation.

- 1 Navigate to the folder with the startup scripts:

```
cd matlabroot\toolbox\distcomp\bin
```

- 2** Start the workers on each node, using the text for <MyMJS> that identifies the name of the MJS you want this worker registered with. Enter this text on a single line:

```
startworker -jobmanagerhost <MJS host name>  
            -jobmanager <MyMJS> -remotehost <worker host name> -v
```

To run more than one worker session on the same node, give each worker a unique name by including the `-name` option on the `startworker` command, and run it for each worker on that node:

```
startworker ... -name <worker1 name>  
startworker ... -name <worker2 name>
```

- 3** Verify that the workers are running.

```
nodestatus -remotehost <worker host name>
```

- 4** Repeat items 2–3 for all worker nodes.

For more information about `mdce`, MJS, and worker processes, such as how to shut them down or customize them, see the “Network Administration” chapter in the *MATLAB Distributed Computing Server System Administrator’s Guide*.

### Using the Command-Line Interface (UNIX)

**Start the `mdce` Service.** On each cluster node, start the `mdce` service by typing the commands:

```
cd matlabroot/toolbox/distcomp/bin  
./mdce start
```

Alternatively (on Linux, but not Macintosh), you can start the `mdce` service on several nodes remotely from one machine by typing

```
cd matlabroot/toolbox/distcomp/bin  
./remotemdce start -remotehost hostA,hostB,hostC . . .
```

where `hostA,hostB,hostC` refers to a list of your host names. Note that there are no spaces between host names, only a comma. If you need to indicate

protocol, platform (such as in a mixed environment), or other information, see the help for `remotemdce` by typing

```
./remotemdce -help
```

**Start the MJS.** To start the MATLAB job scheduler (MJS), enter the following commands. You do not have to be at the machine on which the MJS runs, as long as you have access to the MDCS installation.

- 1 Navigate to the folder with the startup scripts:

```
cd matlabroot/toolbox/distcomp/bin
```

- 2 Start the MJS, using any unique text you want for the name <MyMJS>. Enter this text on a single line.

```
./startjobmanager -name <MyMJS> -remotehost <MJS host name> -v
```

- 3 Verify that the MJS is running on the intended host:

```
./nodestatus -remotehost <MJS host name>
```

---

**Note** If you have more than one MJS on your cluster, each must have a unique name.

---

## Start the Workers.

---

**Note** Before you can start a worker on a machine, the `mdce` service must already be running on that machine, and the license manager for MATLAB Distributed Computing Server must be running on the network.

---

For each computer hosting a MATLAB worker, enter the following commands. You do not have to be at the machines where the MATLAB workers run, as long as you have access to the MDCS installation.

- 1 Navigate to the folder with the startup scripts:

```
cd matlabroot/toolbox/distcomp/bin
```

- 2 Start the workers on each node, using the text for <MyMJS> that identifies the name of the MJS you want this worker registered with. Enter this text on a single line:

```
./startworker -jobmanagerhost <MJS host name>  
-jobmanager <MyMJS> -remotehost <worker host name> -v
```

To run more than one worker session on the same machine, give each worker a unique name with the `-name` option:

```
./startworker ... -name <worker1>  
./startworker ... -name <worker2>
```

- 3 Verify that the workers are running. Repeat this command for each worker node:

```
./nodestatus -remotehost <worker host name>
```

For more information about mdce, MJS, and worker processes, such as how to shut them down or customize them, see the “Network Administration” chapter in the *MATLAB Distributed Computing Server System Administrator’s Guide*.

### Step 4: Install the mdce Service to Start Automatically at Boot Time (UNIX)

Although this step is not required, it is helpful in case of a system crash. Once configured for this, the mdce service starts running each time the machine reboots. The mdce service continues to run until explicitly stopped, regardless of whether an MJS or worker session is running.

You must have root privileges to do this step.

#### Debian, Fedora Platforms

On each cluster node, register the mdce service as a known service and configure it to start automatically at system boot time by following these steps:

- 1 Create the following link, if it does not already exist:

```
ln -s matlabroot/toolbox/distcomp/bin/mdce /etc/mdce
```

- 2** Create the following link to the boot script file:

```
ln -s matlabroot/toolbox/distcomp/bin/mdce /etc/init.d/mdce
```

- 3** Set the boot script file permissions:

```
chmod 555 /etc/init.d/mdce
```

- 4** Look in /etc/inittab for the default run level. Create a link in the rc folder associated with that run level. For example, if the run level is 5, execute these commands:

```
cd /etc/rc5.d;
ln -s ../init.d/mdce S99MDCE
```

## SUSE Platform

On each cluster node, register the mdce service as a known service and configure it to start automatically at system boot time by following these steps:

- 1** Create the following link, if it does not already exist:

```
ln -s matlabroot/toolbox/distcomp/bin/mdce /etc/mdce
```

- 2** Create the following link to the boot script file:

```
ln -s matlabroot/toolbox/distcomp/bin/mdce /etc/init.d/mdce
```

- 3** Set the boot script file permissions:

```
chmod 555 /etc/init.d/mdce
```

- 4** Look in /etc/inittab for the default run level. Create a link in the rc folder associated with that run level. For example, if the run level is 5, execute these commands:

```
cd /etc/init.d/rc5.d;
ln -s ../mdce S99MDCE
```

### Red Hat Platform (non-Fedora)

On each cluster node, register the mdce service as a known service and configure it to start automatically at system boot time by following these steps:

- 1 Create the following link, if it does not already exist:

```
ln -s matlabroot/toolbox/distcomp/bin/mdce /etc/mdce
```

- 2 Create the following link to the boot script file:

```
ln -s matlabroot/toolbox/distcomp/bin/mdce /etc/init.d/mdce
```

- 3 Set boot script file permissions:

```
chmod 555 /etc/init.d/mdce
```

- 4 Look in `/etc/inittab` for the default run level. Create a link in the `rc` folder associated with that run level. For example, if the run level is 5, execute these commands:

```
cd /etc/rc.d/rc5.d;  
ln -s ../../init.d/mdce S99MDCE
```

### Macintosh Platform

On each cluster node, register the mdce service as a known service with `launchd`, and configure it to start automatically at system boot time by following these steps:

- 1 Navigate to the toolbox folder and stop the running mdce service:

```
cd matlabroot/toolbox/distcomp/bin  
sudo ./mdce stop
```

- 2 Create the following link if it does not already exist:

```
sudo ln -s matlabroot/toolbox/distcomp/bin/mdce /usr/sbin/mdce
```

- 3 Copy the `launchd.plist` file for mdce to `/Library/LaunchDaemons`:



```
sudo cp ./util/com.mathworks.mdce.plist /Library/LaunchDaemons
```

- 4** Start mdce and observe that it starts inside launchd:

```
sudo ./mdce start
```

The command output should read:

```
Starting the MATLAB Distributed Computing Server using launchctl.
```

# Configure Windows Firewalls on Client

If you are using Windows firewalls on your client node,

- 1 Log in as a user with administrative privileges.
- 2 Execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them for similar accommodation.

## Validate Installation with MJS

This procedure verifies that your parallel computing products are installed and configured correctly.

### Step 1: Verify the Cluster Connection

To verify the network connection from the client computer to the MJS computer, follow these instructions.

---

**Note** In these instructions, *matlabroot* refers to the folder where MATLAB is installed on the client computer. Do not confuse this with the MDCS cluster computers.

---

- 1 On the client computer where Parallel Computing Toolbox is installed, open a DOS command window (for Windows software) or a shell (for UNIX<sup>®</sup> software) and go to the control script folder.

```
cd matlabroot\toolbox\distcomp\bin (for Windows)
cd matlabroot/toolbox/distcomp/bin (for UNIX)
```

- 2 Run `nodestatus` to verify your cluster communications. Substitute <MJS Host> with the host name of your MJS computer.

```
nodestatus -remotehost <MJS Host>
```

If successful, you should see the status of your MJS (job manager) and its workers. Otherwise, refer to the troubleshooting section of the *MATLAB Distributed Computing Server System Administrator's Guide*.

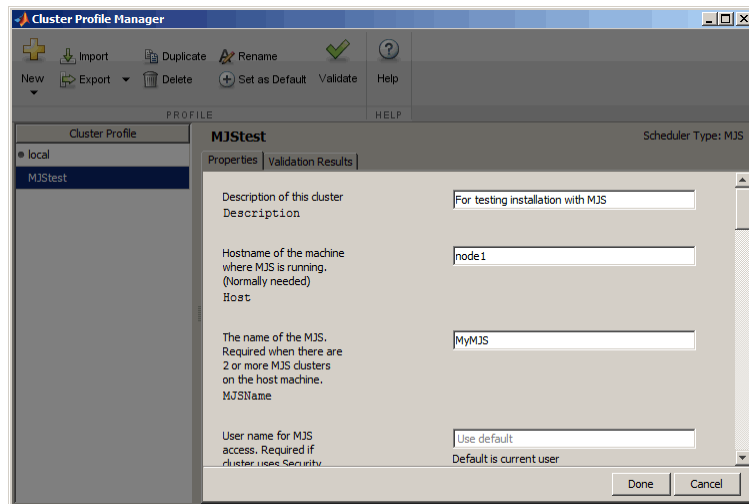
### Step 2: Define a Cluster Profile

In this step you define a cluster profile to use in subsequent steps.

- 1 Start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Cluster Profiles**.
- 2 Create a new profile in the Cluster Profile Manager by selecting **New > MATLAB Job Scheduler (MJS)**.

- 3 With the new profile selected in the list, click **Rename** and edit the profile name to be **MJStest**. Press **Enter**.
- 4 In the Properties tab, provide settings for the following fields:
  - a Set the **Description** field to **For testing installation with MJS**.
  - b Set the **Host** field to the name of the host on which your MJS is running. Depending on your network, this might be only a host name, or it might have to be a fully qualified domain name.
  - c Set the **MJSName** field to the name of your MJS, which you started earlier.

So far, the dialog box should look like the following figure:



- 5 Click **Done** to save your cluster profile.

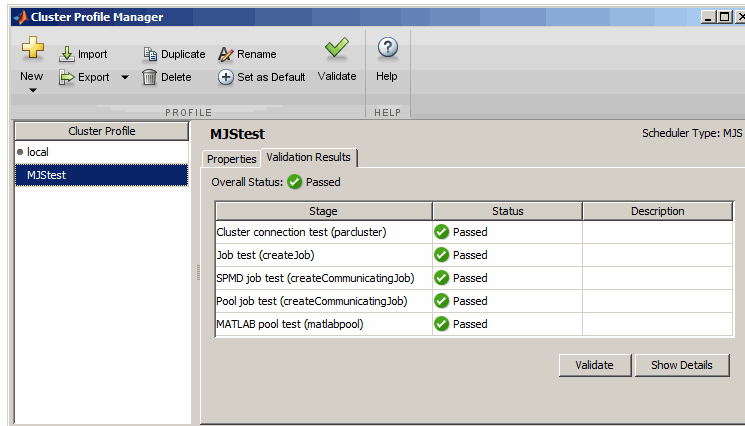
### Step 3: Validate the Cluster Profile

In this step you validate your cluster profile, and thereby your installation.

- 1 If it is not already open, start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Cluster Profiles**.
- 2 Select your cluster profile in the listing.

### 3 Click **Validate**.

The Validation Results tab shows the output. The following figure shows the results of a profile that passed all validation tests.




---

**Note** If your validation does not pass, contact the MathWorks install support team.

---

If your validation passed, you now have a valid profile that you can use in other parallel applications. You can make any modifications to your profile appropriate for your applications, such as `NumWorkersRange`, `AttachedFiles`, `AdditionalPaths`, etc. To save your profile for other users, select the profile and click **Export**, then save your profile to a file in a convenient location. Later, when running the Cluster Profile Manager from a MATLAB client session, other users can import your profile by selecting **Parallel > Import Cluster Profile**.



# Configuring Parallel Computing Products for HPC Server

---

- “Configure Cluster for Microsoft Windows HPC Server” on page 3-2
- “Configure Client Computer for HPC Server 2008” on page 3-3
- “Validate Installation Using Microsoft Windows HPC Server” on page 3-4

# Configure Cluster for Microsoft Windows HPC Server

---

**Note** You must have the 64-bit version of MATLAB Distributed Computing Server installed on your cluster to use it with HPC Server. 32-bit MDCS does not support HPC Server.

---

Follow these instruction to configure your MDCS installation to work with Windows HPC Server or Compute Cluster Server (CCS). In the following instructions, *matlabroot* refers to the MATLAB installation location.

---

**Note** If using HPC Server 2008 in a network share installation, the network share location must be in the “Intranet” zone. You might need to adjust the Internet Options for your cluster nodes and add the network share location to the list of Intranet sites.

---

- 1 Log in on a cluster machine as a user with administrator privileges.
- 2 Open a command window with administrator privileges and run the following file command

```
matlabroot\toolbox\distcomp\bin\MicrosoftHPCServerSetup.bat -cluster
```

This command performs all the setup required for a cluster. The location of the MATLAB installation must be the same on every cluster node.

---

**Note** If you need to override the default values the script, modify the values defined in `MicrosoftHPCServerSetup.xml` before running `MicrosoftHPCServerSetup.bat`. Use the `-def_file` argument to the script when using a `MicrosoftHPCServerSetup.xml` file in a custom location. For example:

```
MicrosoftHPCServerSetup.bat -cluster -def_file <filename>
```

---



## Configure Client Computer for HPC Server 2008

This configuring applies to all versions of HPC Server 2008, including HPC Server 2008 R2.

---

**Note** If using HPC Server 2008 in a network share installation, the network share location must be in the “Intranet” zone. You might need to adjust the Internet Options for your cluster nodes and add the network share location to the list of Intranet sites.

---

- 1 Open a command window with administrator privileges and run the following file command

```
matlabroot\toolbox\distcomp\bin\MicrosoftHPCServerSetup.bat -client
```

This command performs all the setup required for a client machine.

---

**Note** If you need to override the default values the script, modify the values defined in MicrosoftHPCServerSetup.xml before running MicrosoftHPCServerSetup.bat. Use the -def\_file argument to the script when using a MicrosoftHPCServerSetup.xml file in a custom location. For example:

```
MicrosoftHPCServerSetup.bat -client -def_file <filename>
```

---

## Validate Installation Using Microsoft Windows HPC Server

This procedure verifies that your parallel computing products are installed and configured correctly for using Microsoft® Windows HPC Server or Compute Cluster Server (CCS).

### Step 1: Define a Cluster Profile

In this step you define a cluster profile to use in subsequent steps.

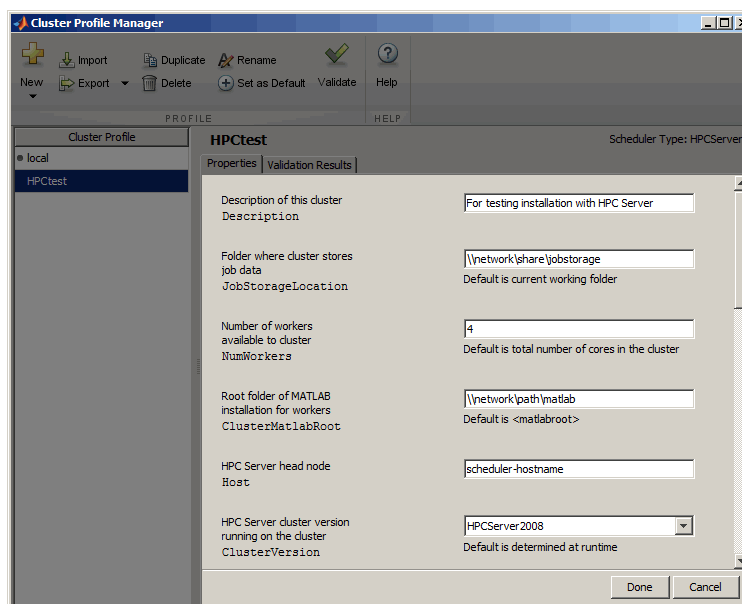
- 1** Start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Profiles**.
- 2** Create a new profile in the Cluster Profile Manager by selecting **New > HPC Server**.
- 3** With the new profile selected in the list, click **Rename** and edit the profile name to be **HPCtest**. Press **Enter**.
- 4** In the Properties tab, provide text for the following fields:
  - a** Set the **Description** field to **For testing installation with HPC Server**.
  - b** Set the **NumWorkers** field to the number of workers you want to run the validation tests on, within the limitation of your licensing.
  - c** Set the **Host** field to the name of the host on which your scheduler is running. Depending on your network, this might be a simple host name, or it might have to be a fully qualified domain name.

Note: The following four property settings (JobStorageLocation, ClusterMatlabRoot, ClusterVersion, and UseSOAJobSubmission) are optional, and need to be set in here in the profile only if you did not run `MicrosoftHPCServerSetup.bat` as described in “Configure Cluster for Microsoft Windows HPC Server” on page 3-2, or if you want to override the setting established by that script.

- d** Set the **JobStorageLocation** to the location where you want job and task data to be stored. This must be accessible to all the worker machines.

- e Set the **ClusterMatlabRoot** to the installation location of the MATLAB to be executed by the worker machines, as determined in Chapter 1 of the installation instructions.
- f Set the **ClusterVersion** field to HPCServer2008 or CCS.
- g If you want to test SOA job submissions on an HPC Server 2008 cluster, set **UseSOAJobSubmission** to true. Otherwise leave the setting Use default or false. If you plan on using SOA job submissions with your cluster, you should test this first without SOA submission, then later return and test it with SOA job submission.

So far, the dialog box should look like the following figure:



- 5 Click **Done** to save your cluster profile.

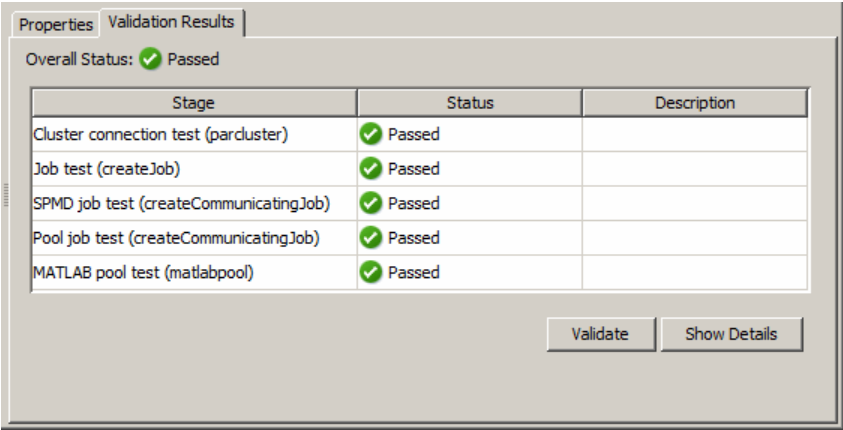
## Step 2: Validate the Configuration

In this step you validate your cluster profile, and thereby your installation.

- 1 If it is not already open, start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Cluster Profiles**.

- 2 Select your cluster profile in the listing.
- 3 Click **Validate**.

The Validation Results tab shows the output. The following figure shows the results of a profile that passed all validation tests.



**Note** If your validation does not pass, contact the MathWorks install support team.

If your validation passed, you now have a valid profile that you can use in other parallel applications. You can make any modifications to your profile appropriate for your applications, such as `NumWorkersRange`, `AttachedFiles`, `AdditionalPaths`, etc. To save your profile for other users, select the profile and click **Export**, then save your profile to a file in a convenient location. Later, when running the Cluster Profile Manager from a MATLAB client session, other users can import your profile by selecting **Parallel > Import Cluster Profile**.

# Configuring Parallel Computing Products for Supported Third-Party Schedulers (PBS Pro, Platform LSF, TORQUE)

---

- “Configure Platform LSF Scheduler on Windows Cluster” on page 4-2
- “Configure Windows Firewalls on Client” on page 4-5
- “Validate Installation Using an LSF, PBS Pro, or TORQUE Scheduler” on page 4-6

---

**Note** You must use the generic scheduler interface for any of the following:

- Any third-party schedule not listed above (e.g., Sun Grid Engine, GridMP, etc.)
  - PBS other than PBS Pro
  - A nonshared file system when the client cannot directly submit to the scheduler (e.g., TORQUE on Windows)
-

# Configure Platform LSF Scheduler on Windows Cluster

If your cluster is already set up to use mpiexec and smpd, you can use Parallel Computing Toolbox™ software with your existing configuration if you are using a compatible MPI implementation library (as defined in *matlabroot\toolbox\distcomp\mpi\mpiLibConf.m*). However, if you do not have mpiexec on your cluster and you want to use it, you can use the mpiexec software shipped with the parallel computing products.

For further information about mpiexec and smpd, see the MPICH2 home page at <http://www.mcs.anl.gov/research/projects/mpich2/>. For user's guides and installation instructions on that page, select **Documentation > User Docs**.

In the following instructions, *matlabroot* refers to the MATLAB installation location.

To use mpiexec to distribute a job, the smpd service must be running on all nodes that will be used for running MATLAB workers.

---

**Note** The smpd executable does not support running from a mapped drive. Use either a local installation, or the full UNC pathname to the executable. Microsoft Windows Vista does not support the smpd executable on network share installations, so with Vista the installation must be local.

---

Choose one of the following configurations:

In this section...
“Without Delegation” on page 4-2
“Using Passwordless Delegation” on page 4-4

## Without Delegation

- 1 Log in as a user with administrator privileges.
- 2 Start smpd by typing in a DOS command window one of the following, as appropriate:

```
matlabroot\bin\win32\smpd -install
```

or

```
matlabroot\bin\win64\smpd -install
```

This command installs the service and starts it. As long as the service remains installed, it will start each time the node boots.

- 3** If this is a worker machine and you did not run the installer on it to install MDCS software (for example, if you are running MDCS software from a shared installation), execute the following command in a DOS command window.

```
matlabroot\bin\matlab.bat -install_vcrt
```

This command installs the Microsoft run-time libraries needed for running distributed and parallel jobs with the your scheduler.

- 4** If you are using Windows firewalls on your cluster nodes, execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them to make similar accommodation.

- 5** Log in as the user who will be submitting jobs for execution on this node.
- 6** Register this user to use mpiexec by typing one of the following, as appropriate:

```
matlabroot\bin\win32\mpiexec -register
```

or

```
matlabroot\bin\win64\mpiexec -register
```

- 7** Repeat steps 5–6 for all users who will run jobs on this machine.
- 8** Repeat all these steps on all Windows nodes in your cluster.

### Using Passwordless Delegation

- 1 Log in as a user with administrator privileges.
- 2 Start `smpd` by typing in a DOS command window one of the following, as appropriate:

```
matlabroot\bin\win32\smpd -register_spn
```

or

```
matlabroot\bin\win64\smpd -register_spn
```

This command installs the service and starts it. As long as the service remains installed, it will start each time the node boots.

- 3 If this is a worker machine and you did not run the installer on it to install MDCS software (for example, if you are running MDCS software from a shared installation), execute the following command in a DOS command window.

```
matlabroot\bin\matlab.bat -install_vcrt
```

This command installs the Microsoft run-time libraries needed for running distributed and parallel jobs with the your scheduler.

- 4 If you are using Windows firewalls on your cluster nodes, execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them for similar accommodation.

- 5 Repeat these steps on all Windows nodes in your cluster.



## Configure Windows Firewalls on Client

If you are using Windows firewalls on your cluster nodes,

- 1 Log in as a user with administrative privileges.
- 2 Execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them for similar accommodation.

# Validate Installation Using an LSF, PBS Pro, or TORQUE Scheduler

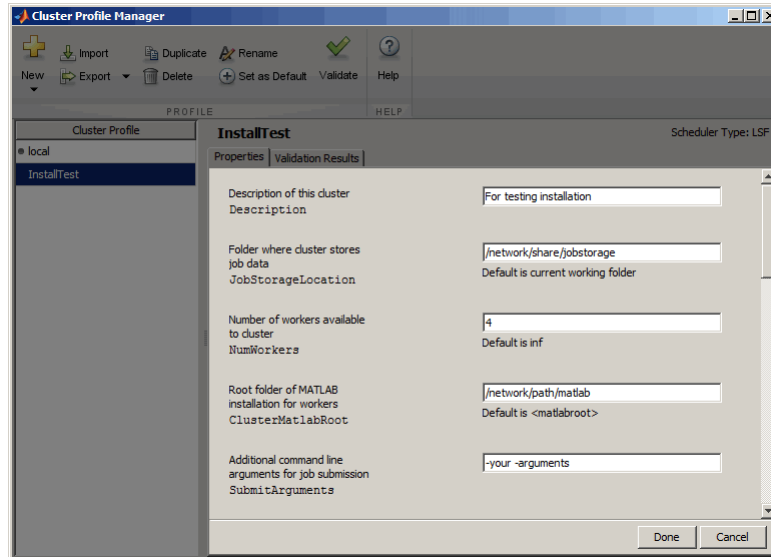
This procedure verifies that the parallel computing products are installed and configured correctly on your cluster.

## Step 1: Define a Cluster Profile

In this step you define a cluster profile to use in subsequent steps.

- 1 Start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Profiles**.
- 2 Create a new profile in the Cluster Profile Manager by selecting **New > LSF** (or **PBS Pro** or **Torque**, as appropriate).
- 3 With the new profile selected in the list, click **Rename** and edit the profile name to be `InstallTest`. Press **Enter**.
- 4 In the Properties tab, provide settings for the following fields:
  - a Set the **Description** field to `For testing installation`.
  - b Set the **JobStorageLocation** to the location where you want job and task data to be stored (accessible to all the worker machines if you have a shared file system).
  - c Set the **NumWorkers** field to the number of workers you want to run the validation tests on, within the limitation of your licensing.
  - d Set the **ClusterMatlabRoot** to the installation location of the MATLAB to be executed by the worker machines, as determined in Chapter 1 of the installation instructions.
  - e Set the **SubmitArguments** to include any additional command arguments required by your particular cluster and scheduler.
  - f If using LSF®, set the **OperatingSystem** to the operating system of your worker machines.
  - g Set **HasSharedFilesystem** to indicate if client and workers can share the same data location.

The dialog box should look something like this, or slightly different for PBS Pro® or TORQUE schedulers.



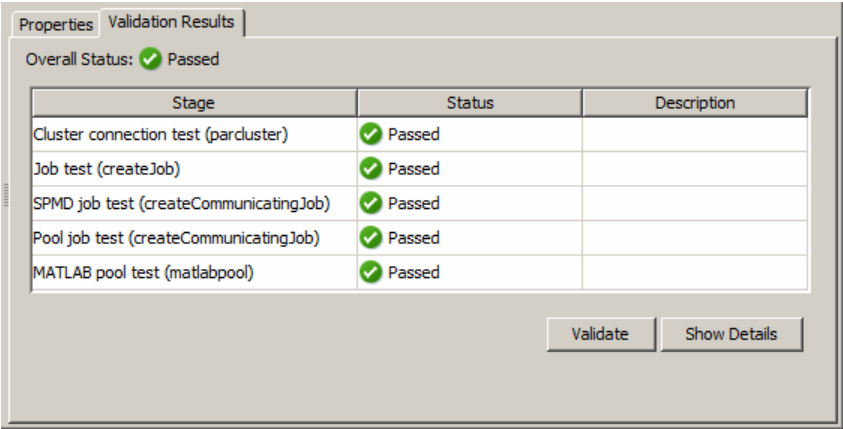
**5** Click **Done** to save your cluster profile.

## Step 2: Validate the Cluster Profile

In this step you verify your cluster profile, and thereby your installation.

- 1** If it is not already open, start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Cluster Profiles**.
- 2** Select your cluster profile in the listing.
- 3** Click **Validate**.

The Validation Results tab shows the output. The following figure shows the results of a profile that passed all validation tests.



**Note** If your validation does not pass, contact the MathWorks install support team.

If your validation passed, you now have a valid profile that you can use in other parallel applications. You can make any modifications to your profile appropriate for your applications, such as `NumWorkersRange`, `AttachedFiles`, `AdditionalPaths`, etc. To save your profile for other users, select the profile and click **Export**, then save your profile to a file in a convenient location. Later, when running the Cluster Profile Manager from a MATLAB client session, other users can import your profile by selecting **Parallel > Import Cluster Profile**.

# Configuring Parallel Computing Products for a Generic Scheduler

---

---

**Note** You must use the generic scheduler interface for any of the following:

- Any third-party schedule not listed in previous chapters (e.g., Sun Grid Engine, GridMP, etc.)
- PBS other than PBS Pro
- A nonshared file system when the client cannot directly submit to the scheduler (e.g., TORQUE on Windows)

---

This chapter includes the following sections. Read all that apply to your configuration:

- “Interfacing with Generic Schedulers” on page 5-2
- “Configure Generic Scheduler on Windows Cluster” on page 5-4
- “Configure Sun Grid Engine on Linux Cluster” on page 5-7
- “Configure Windows Firewalls on Client” on page 5-9
- “Validate Installation Using a Generic Scheduler” on page 5-10

## Interfacing with Generic Schedulers

In this section...
“Support Scripts” on page 5-2
“Submission Mode” on page 5-2

### Support Scripts

To support usage of the generic scheduler interface, templates and scripts are provided with the product in the folder:

*matlabroot\toolbox\distcomp\examples\integration* (on Windows)

*matlabroot/toolbox/distcomp/examples/integration* (on UNIX)

Subfolders are provided for several different kinds of schedulers, and each of those contains a subfolder for the supported usage modes for shared file system, nonshared file system, or remote submission. Each folder contains a file named README that provides specific instructions on how to use the scripts.

Further information on programming independent jobs for generic schedulers, see:

<http://www.mathworks.com/access/helpdesk/help/toolbox/distcomp/bqur7ev-35.html>

Further information on programming communicating jobs for generic schedulers, see:

<http://www.mathworks.com/access/helpdesk/help/toolbox/distcomp/bqxceii-1.html>

### Submission Mode

The provided scripts support three possible submission modes:

- Shared — When the client machine is able to submit directly to the cluster and there is a shared file system present between the client and the cluster machines.
- Remote Submission — When there is a shared file system present between the client and the cluster machines, but the client machine is not able to

submit directly to the cluster (for example, if the scheduler's client utilities are not installed).

- Nonshared — When there is not a shared file system between client and cluster machines.

Before using the support scripts, decide which submission mode describes your particular network setup.

## Configure Generic Scheduler on Windows Cluster

If your cluster is already set up to use mpiexec and smpd, you can use Parallel Computing Toolbox™ software with your existing configuration if you are using a compatible MPI implementation library (as defined in *matlabroot*\toolbox\distcomp\mpi\mpiLibConf.m). However, if you do not have mpiexec on your cluster and you want to use it, you can use the mpiexec software shipped with the parallel computing products.

For further information about mpiexec and smpd, see the MPICH2 home page at <http://www.mcs.anl.gov/research/projects/mpich2/>. For user's guides and installation instructions on that page, select **Documentation > User Docs**.

In the following instructions, *matlabroot* refers to the MATLAB installation location.

To use mpiexec to distribute a job, the smpd service must be running on all nodes that will be used for running MATLAB workers.

---

**Note** The smpd executable does not support running from a mapped drive. Use either a local installation, or the full UNC pathname to the executable. Microsoft Windows Vista does not support the smpd executable on network share installations, so with Vista the installation must be local.

---

Choose one of the following configurations:

In this section...
"Without Delegation" on page 5-4
"Using Passwordless Delegation" on page 5-6

### Without Delegation

- 1 Log in as a user with administrator privileges.
- 2 Start smpd by typing in a DOS command window one of the following, as appropriate:



```
matlabroot\bin\win32\smpd -install
```

or

```
matlabroot\bin\win64\smpd -install
```

This command installs the service and starts it. As long as the service remains installed, it will start each time the node boots.

- 3** If this is a worker machine and you did not run the installer on it to install MDCS software (for example, if you are running MDCS software from a shared installation), execute the following command in a DOS command window.

```
matlabroot\bin\matlab.bat -install_vcrt
```

This command installs the Microsoft run-time libraries needed for running distributed and parallel jobs with the your scheduler.

- 4** If you are using Windows firewalls on your cluster nodes, execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them to make similar accommodation.

- 5** Log in as the user who will be submitting jobs for execution on this node.
- 6** Register this user to use mpiexec by typing one of the following, as appropriate:

```
matlabroot\bin\win32\mpiexec -register
```

or

```
matlabroot\bin\win64\mpiexec -register
```

- 7** Repeat steps 5–6 for all users who will run jobs on this machine.
- 8** Repeat all these steps on all Windows nodes in your cluster.

## Using Passwordless Delegation

- 1 Log in as a user with administrator privileges.
- 2 Start `smpd` by typing in a DOS command window one of the following, as appropriate:

```
matlabroot\bin\win32\smpd -register_spn
```

or

```
matlabroot\bin\win64\smpd -register_spn
```

This command installs the service and starts it. As long as the service remains installed, it will start each time the node boots.

- 3 If this is a worker machine and you did not run the installer on it to install MDCS software (for example, if you are running MDCS software from a shared installation), execute the following command in a DOS command window.

```
matlabroot\bin\matlab.bat -install_vcrt
```

This command installs the Microsoft run-time libraries needed for running distributed and parallel jobs with the your scheduler.

- 4 If you are using Windows firewalls on your cluster nodes, execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them for similar accommodation.

- 5 Repeat these steps on all Windows nodes in your cluster.

## Configure Sun Grid Engine on Linux Cluster

To run parallel jobs with MATLAB Distributed Computing Server and Sun™ Grid Engine (SGE), you need to establish a “matlab” parallel environment for SGE. The “matlab” parallel environment described in these instructions is based on the “MPI” example shipped with SGE. To use this parallel environment, you must use the `matlabpe.template`, customized to match the number of slots available, and to indicate where the `startmatlabpe.sh` and `stopmatlabpe.sh` scripts are installed on your cluster.

In the following instructions, *matlabroot* refers to the MATLAB installation location.

### Create the Parallel Environment

The following steps create the parallel environment (PE), and then make the parallel environment runnable on a particular queue. You should perform these steps on the head node of your cluster.

- 1 Navigate to the folder of the integration files appropriate for your cluster: `shared`, `nonshared`, or `remoteSubmission`, with one of the following shell commands.

```
cd matlabroot/toolbox/distcomp/examples/integration/sge/shared
cd matlabroot/toolbox/distcomp/examples/integration/sge/nonshared
cd matlabroot/toolbox/distcomp/examples/integration/sge/remoteSubmission
```

- 2 Modify the contents of `matlabpe.template` to use the desired number of slots and the correct location of the `startmatlabpe.sh` and `stopmatlabpe.sh` files. (These files can exist in a shared location accessible by all hosts, or they can be copied to the same local on each host.) You can also change other values or add additional values to `matlabpe.template` to suit your cluster. For more information, refer to the `sge_pe` documentation provided with your scheduler.

- 3 Add the “matlab” parallel environment, using a shell command like:

```
qconf -Ap matlabpe.template
```

- 4 Make the “matlab” parallel environment runnable on all queues:

```
qconf -mq all.q
```

This will bring up a text editor for you to make changes: search for the line `pe_list`, and add `matlab`.

- 5** Ensure you can submit a trivial job to the PE:

```
$ echo "hostname" | qsub -pe matlab 1
```

- 6** Use `qstat` to check that the job runs correctly, and check that the output file contains the name of the host that ran the job. The default filename for the output file is `~/STDIN.o###`, where `###` is the SGE job number.

---

**Note** The example submit functions for SGE rely on the presence of the “matlab” parallel environment. If you change the name of the parallel environment to something other than “matlab”, you must ensure that you also change the submit functions.

---

## Configure Windows Firewalls on Client

If you are using Windows firewalls on your cluster nodes,

- 1 Log in as a user with administrative privileges.
- 2 Execute the following in a DOS command window.

```
matlabroot\toolbox\distcomp\bin\addMatlabToWindowsFirewall.bat
```

This command adds MATLAB as an allowed program. If you are using other firewalls, you must configure them to for accommodation.

# Validate Installation Using a Generic Scheduler

Testing the installation of the parallel computing products with a generic scheduler requires familiarity with your network configuration, with your scheduler interface, and with the generic scheduler interface of Parallel Computing Toolbox software.

---

**Note** The remainder of this chapter illustrates only the case of using LSF in a nonshared file system. For other schedulers or a shared file system, look for the appropriate scripts and modify them as necessary, using the following instructions as a guide. If you have any questions, contact the MathWorks install support team.

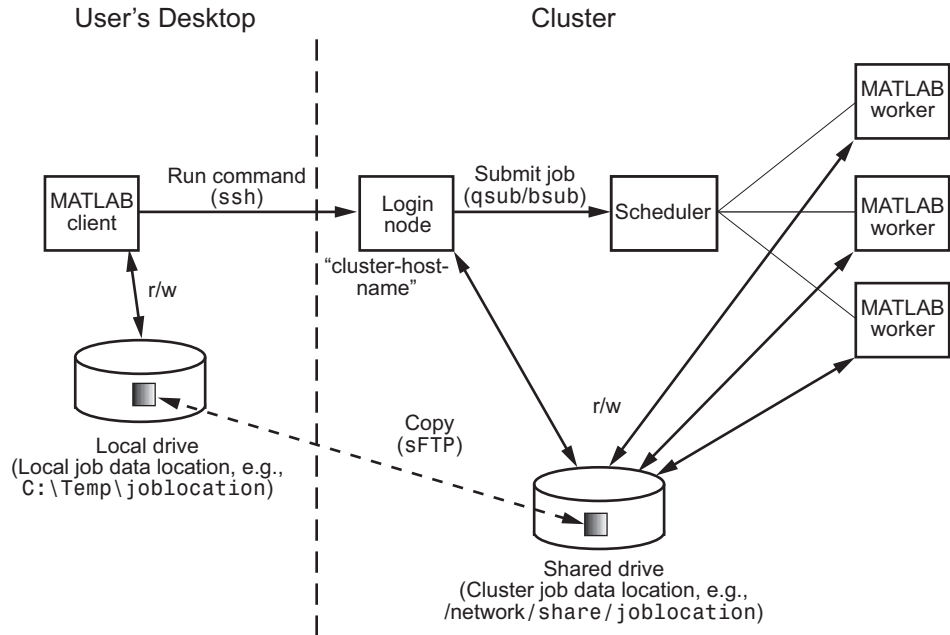
---

## Example Setup for LSF

This section provides guidelines for setting up your cluster profile to use the generic scheduler interface with an LSF scheduler in a network without a shared file system between the client and the cluster machines. The scripts necessary to set up your test are found in:

`matlabroot/toolbox/distcomp/examples/integration/lsf/nonshared`

These scripts are written for an LSF scheduler, but might require modification to work in your network. The following diagram illustrates the cluster setup:



In this type of configuration, job data is copied from the client host running a Windows operating system to a host on the cluster (cluster login node) running a UNIX operating system. From the cluster login node, the LSF `bsub` command submits the job to the scheduler. When the job finishes, its output is copied back to the client host.

## Requirements

For this setup to work, the following conditions must be met:

- The client node and cluster login node must support `ssh` and `sFTP`.
- The cluster login node must be able to call the `bsub` command to submit a job to an LSF scheduler. You can find more about this in the file:

`matlabroot\toolbox\distcomp\examples\integration\lsf\nonshared\README`

If these requirements are met, use the following steps to implement the solution:

### Step 1: Set Up Windows Client Host

#### On the Client Host

- 1 You need the necessary scripts on the path of the MATLAB client. You can do this by copying them to a folder already on the path.

Browse to the folder:

```
matlabroot\toolbox\distcomp\examples\integration\lsf\nonshared
```

Copy all the files from that folder, and paste them into the folder:

```
matlabroot\toolbox\local
```

### Step 2: Define a Cluster Profile

In this step you define a cluster profile to use in subsequent steps.

- 1 Start a MATLAB session on the client host.
- 2 Start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Cluster Profiles**.
- 3 Create a new profile in the Cluster Profile Manager by selecting **New > Generic**.
- 4 With the new profile selected in the list, click **Rename** and edit the profile name to be `InstallTest`. Press **Enter**.
- 5 In the Properties tab, provide settings for the following fields:
  - a Set the **Description** field to `For testing installation`.
  - b Set the **JobStorageLocation** to the location where you want job and task data to be stored *on the client machine* (not the cluster location).
  - c Set the **NumWorkers** to the number of workers you want to test your installation on.
  - d Set the **ClusterMatlabRoot** to the installation location of the MATLAB to be executed by the worker machines, as determined in Chapter 1 of the installation instructions.



- e Set **IndependentSubmitFcn** with the following text:

```
{@independentSubmitFcn, 'cluster-host-name', '/network/share/joblocation'}
```

where

`cluster-host-name` is the name of the *cluster host* from which the job will be submitted to the scheduler; and, `/network/share/joblocation` is the location *on the cluster* where the scheduler can access job data. This must be accessible from all cluster nodes.

- f Set **CommunicatingSubmitFcn** with the following text:

```
{@communicatingSubmitFcn, 'cluster-host-name', '/network/share/joblocation'}
```

- g Set the **OperatingSystem** to the operating system of your cluster worker machines.
- h Set **HasSharedFilesystem** to `false`, indicating that the client node and worker nodes cannot share the same data location.
- i Set the **GetJobStateFcn** to `@getJobStateFcn`.
- j Set the **DeleteJobFcn** field to `@deleteJobFcn`.

- 6 Click **Done** to save your cluster profile changes.

The dialog box should look something like this.

InstallTest

Scheduler Type: Generic

Properties

Validation Results

Description of this cluster

Description

For testing installation

Folder where cluster stores job data

JobStorageLocation

c:\Temp\joblocation

Number of workers available to cluster

NumWorkers

4

Root folder of MATLAB installation for workers

ClusterMatlabRoot

/network/path/matlab

SUBMIT FUNCTIONS

Function called when submitting independent jobs

IndependentSubmitFcn

{@independentSubmitFcn, 'cluster-host-name', '/network/share/joblocation'}

Function called when submitting communicating jobs

CommunicatingSubmitFcn

{@communicatingSubmitFcn, 'cluster-host-name', '/network/share/joblocation'}

CLUSTER ENVIRONMENT

Cluster nodes' operating system

OperatingSystem

unix

Job storage location is accessible from client and cluster nodes

HasSharedFilesystem

false

FILES AND FOLDERS

JOBS AND TASK FUNCTIONS

Function to query cluster about job state

GetJobStateFcn

@getJobStateFcn

Function to manage cluster when you call delete on a job

DeleteJobFcn

@deleteJobFcn

Function to manage cluster when you call delete on a task

DeleteTaskFcn

<none>

Function to manage cluster when you call cancel on a job

CancelJobFcn

<none>

Function to manage cluster when you call cancel on a task

CancelTaskFcn

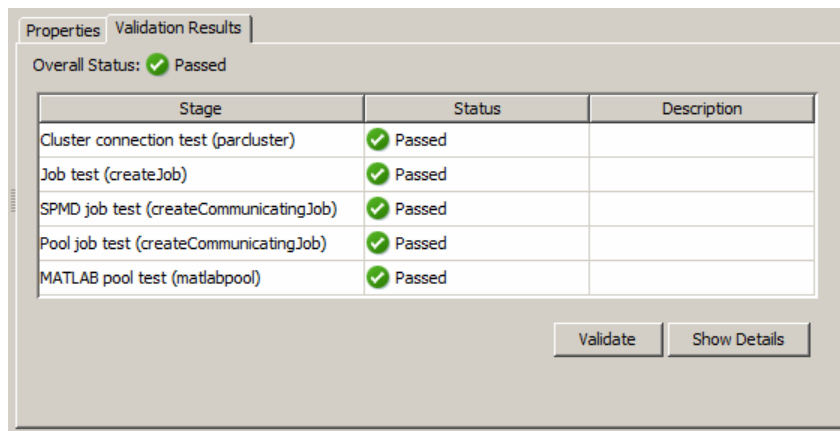
<none>

Step 3: Validate Cluster Profile

In this step you validate your cluster profile, and thereby your installation.

- 1 If it is not already open, start the Cluster Profile Manager from the MATLAB desktop by selecting **Parallel > Manage Cluster Profiles**.
- 2 Select your cluster profile in the listing.
- 3 Click **Validate**.

The Validation Results tab shows the output. The following figure shows the results of a profile that passed all validation tests.



**Note** If your validation fails any stage, contact the MathWorks install support team.

If your validation passed, you now have a valid profile that you can use in other parallel applications. You can make any modifications to your profile appropriate for your applications, such as `NumWorkersRange`, `AttachedFiles`, `AdditionalPaths`, etc. To save your profile for other users, select the profile and click **Export**, then save your profile to a file in a convenient location. Later, when running the Cluster Profile Manager from a MATLAB client session, other users can import your profile by selecting **Parallel > Import Cluster Profile**.