Windows HPC Server 2008 Job Templates

Microsoft Corporation
Published: June 2008

Abstract

Windows® HPC Server 2008 job templates are custom submission policies that can help streamline the job submission process. The IT administrator defines the job template, then the user specifies the desired job template at the command line or in the Administration Console, and the job is off to the compute cluster.

For the IT administrator, job templates provide a way to control the job submission process— they provide a means of partitioning the cluster, prioritizing the jobs, and limiting the resources used by jobs. They let the IT administrator specify different handling for different types of jobs. For the user, job templates provide a way to easily submit a job without having to obtain an in-depth understanding of how the Job Scheduler works. Job templates serve as a pattern for the creation of jobs that meet the user's processing needs, while also conforming to the organization's resource allocation policies. Job templates are powerful, easy to use, and can save time and effort for both users and IT administrators.

Microsoft

Microsoft®

Contents

Job Templates in Windows HPC Server 2008

High performance computing (HPC) applications use a cluster of computers working together to solve a single computational problem or single set of closely related computational problems. Windows® HPC Server 2008 enables such cluster-based supercomputing based on x64 versions of the Windows Server® 2008 operating system. Windows HPC Server 2008 can efficiently scale to thousands of processing cores and provides a comprehensive set of deployment, administration, and monitoring tools that are easy to deploy, manage, and integrate with an existing infrastructure. A wide range of software vendors in various vertical markets have designed their applications to work seamlessly with Windows HPC Server 2008, so users can submit and monitor jobs from within familiar applications without having to learn new or complex user interfaces.

Windows HPC Server 2008 provides a host of features, including a more advanced Job Scheduler, a faster Microsoft® Message Passing Interface (MS-MPI), rapid deployment options using Windows® Deployment Services (WDS), and a new management interface built on the Microsoft® System Center user interface (UI) that supports Windows PowerShell™ as a preferred scripting language. Windows HPC Server 2008 takes advantage of Windows Server 2008 failover services, in addition to the failover clustering capabilities of Microsoft® SQL Server®, to provide for high availability and redundancy.

One of the most powerful new features in Windows HPC Server 2008 is the ability to create job templates. A job template is a custom submission policy created by an IT administrator to define the job parameters for an application, freeing the user from having to learn the Job Scheduler nomenclature before submitting a job from the HPC Job Manager or command line. The IT administrator can use job templates to easily manage job submission and optimize cluster usage. Jobs submitted with job templates all enter a single queue, making the job queue easier to monitor and manage. For the user, job templates simplify the process of submitting the job—the user just specifies the desired template to use, so no in-depth job scheduling knowledge is required.

In this white paper, we describe how job templates can help the IT administrator partition, prioritize, and limit cluster resources. We show how easy it is to create, edit, and use job templates for some common scenarios. We also describe some more advanced options that an IT administrator may want to employ. This white paper provides IT administrators and users with important information needed to put job templates to work in their own cluster environments.

What Is a Job Template?

Job templates are one of the most helpful features in Windows HPC Server 2008. Simply put, job templates are custom submission policies configured by the IT administrator and employed by the cluster users to submit jobs. Administrators can create a number of different job templates, and then let users pick the one that is right for their job. Most job admission policies can be created using job templates with just a few mouse clicks. Submission filters (which parse and validate a job's XML file) are also supported in Windows HPC Server 2008 to enable more complex scenarios.

Job templates enable IT administrators to provide sets of defaults and constraints for the different jobs submitted to the system. Job templates let the IT administrator:

- Enforce run time and size limits on jobs.
- Limit job priority.
- Send different types of jobs to different sets of nodes.
- Partition the cluster for different users groups.
- Simplify job submission for the user.
- Customize the job submission process in many ways to support a variety of scenarios.

To accomplish these goals, the IT administrator can:

- Create one or more job templates.
- Select a set of job properties for the job template. For each job property, the administrator can provide:
    - o A default value that will be used on all job properties not set by the user.
    - o A list or range of acceptable values that the user can provide.
- Set permissions for which users or groups can submit jobs using each template.

The job queues used in many third-party scheduling products also provide some of these abilities; however, the jobs submitted with job templates in Windows HPC Server 2008 all enter a single queue. This keeps all of the jobs that are waiting to execute in one place, so job queue is easier to monitor and manage.

Because job templates can define default values for any job properties, they free the user from having to learn the Job Scheduler nomenclature before submitting a job. Users can select the desired template and easily submit the job from the command-line interface or the HPC Job Manager.

How Is a Job Template Created?

The IT administrator creates the job templates using an easy wizard-driven process that provides many useful options. We start by describing how to create a simple job template and how to set permissions to control access to it. We then explain how to use job templates for some common scenarios: partitioning the cluster, prioritizing access to shared resources, and constraining resources such as the run time and size of jobs. Finally, we discuss the default job template and explore the more advanced options that an administrator may want to use.

Creating a Basic Job Template

We begin by creating a simple, basic job template that limits the priorities that users can assign to their jobs. The process starts in the Administration Console (from the **Start** menu, click on **HPC Cluster Manager**).

1. Click **Configuration** in the lower-left pane of the HPC Cluster Manager, and then click **Job Templates** in the **Navigation** pane.

2. In the **Job Templates** results pane, a new installation displays a default job template. You can clone that template and edit it, or you can create a new job template. In this example, we create a new job template.

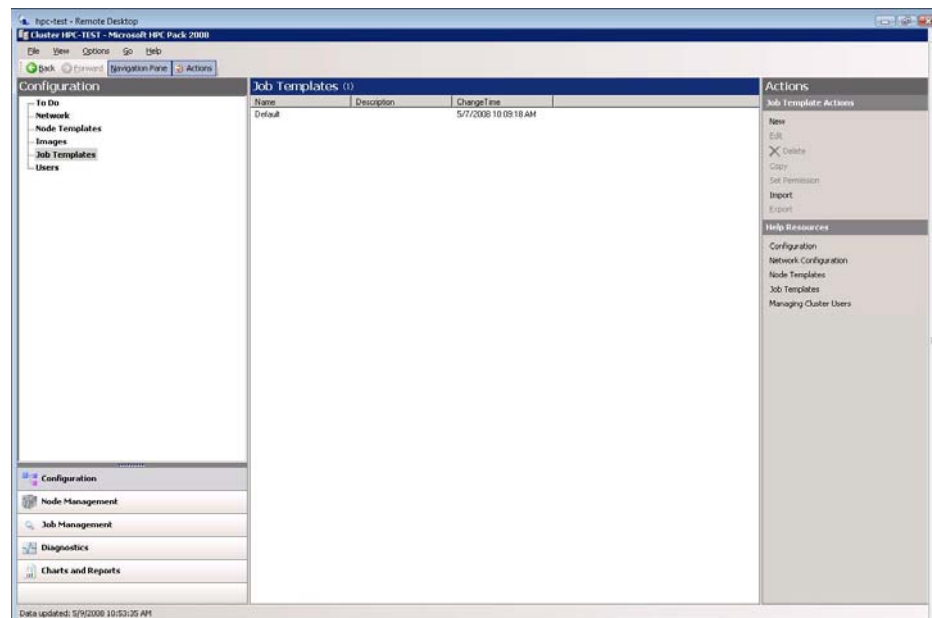3. Select **New** in the **Actions** pane on the right.

Figure 1 Job templates pane

4. The **Generate Job Template** wizard launches. On the **Welcome** page, you can provide a name for the new job template. You can also provide a description for the new job template—this is optional, but descriptions may be helpful if you create many job templates.

   In this example, we use the default, **New Job Template 1**, as a name for our template. Click **Next**.
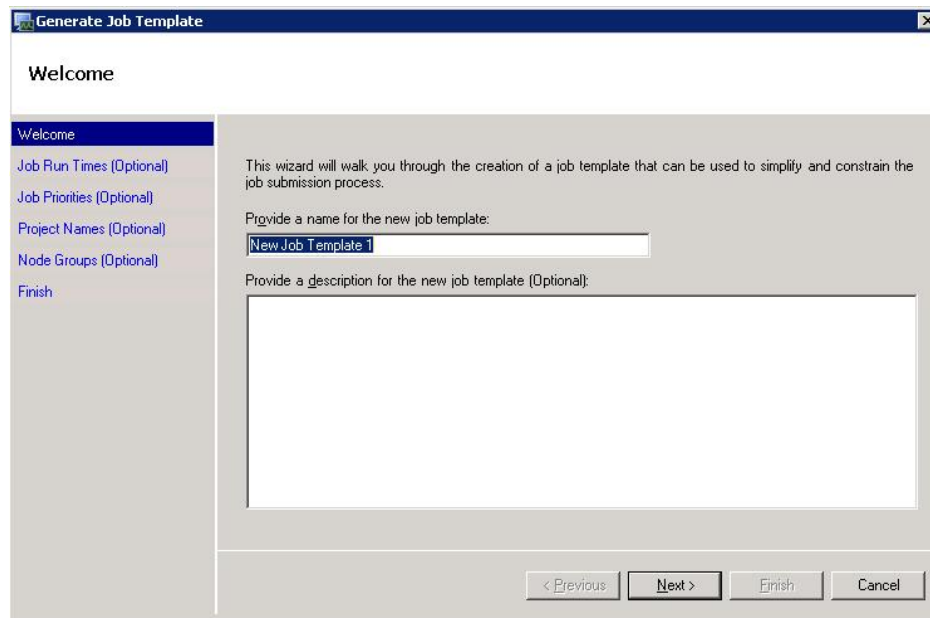

Figure 2 Welcome page

5. On the **Job Run Times** page, you can enter a maximum run time. This is optional, but in general it is recommended that all cluster administrators enforce a maximum run time on jobs.
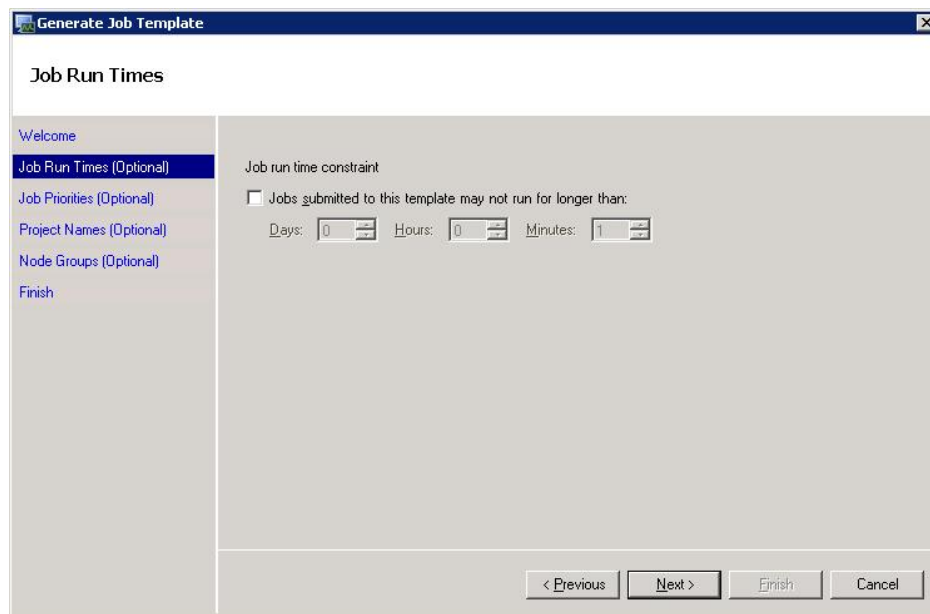

Figure 3 Job Run Times page

6. On the **Job Priorities** page, you can set a default priority and a maximum priority from the drop-down menus. There are five options: **Lowest**, **BelowNormal**, **Normal**, **AboveNormal**, and **Highest**.

For this example template, set the **Default priority** to **Lowest** and the **Maximum priority** to **BelowNormal**, and then click **Next**. A user submitting a job with this template now has only those choices—if the user does not specify the priority as **BelowNormal**, the job will be submitted as **Lowest** priority; no jobs submitted with this template can submit a job with a higher priority.

Figure 4 Job Priorities page

7. On the **Project Names** page, you can ensure that jobs submitted to this template must name a project from a specified list. This lets you more easily catalog and track the jobs submitted to your cluster.

   In this example, select **Don't constrain the Project Name for jobs submitted to this template**. Click **Next**.

Figure 5 Project Names page

8. On the **Node Groups** page, you can constrain the node groups on which jobs submitted to this template may run. You may choose to specify that jobs submitted to this template may run only on nodes that are in all of the selected node groups—effectively partitioning the cluster—or choose to route jobs to nodes that meet certain characteristics.

> **Note:** For more details on how to use job templates to partition clusters, see the section Using a Job Template to Constrain a Group of Users to a Particular Node Group later in this document.

For this example, select **Don't constrain the Project Name for jobs submitted to this template**.

Figure 6 Node Groups page

9. On the **Finish** page, click **Finish**.
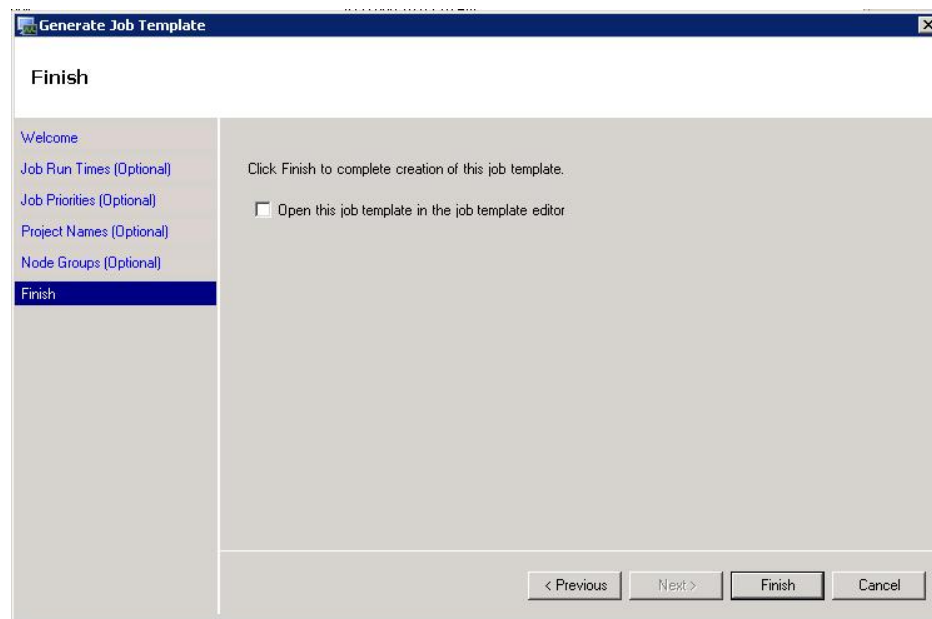
Figure 7 Finish page

10. In the Administration Console, you can see that the new template has been added to the list. You now have a job template that is ready for use, but you may want to set permissions so that users only have access to the appropriate templates.
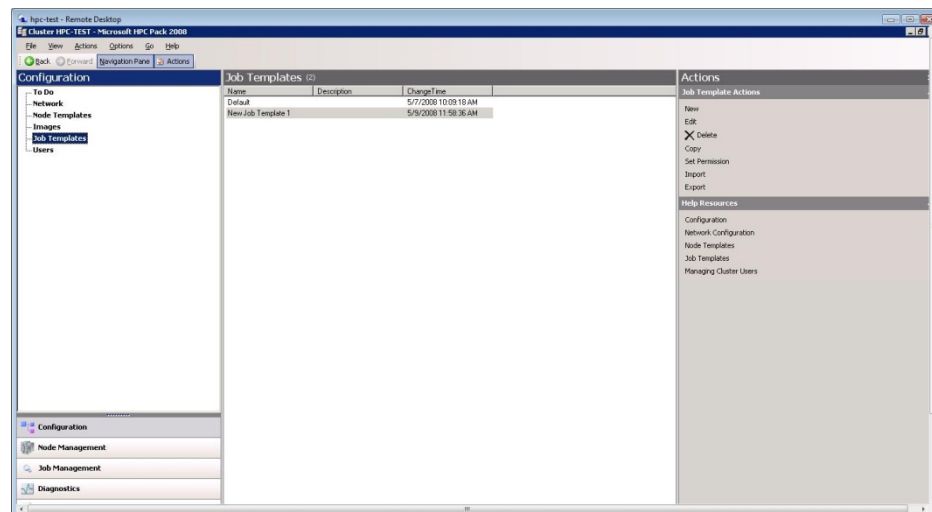
Figure 8 Administration Console

Setting Permissions on the Job Template

Access control is the process of authorizing users, groups, and computers to access objects on the network or computer. The access control list (ACL) is a list of permissions—which define the type of access granted to a user or group for an object or object property—attached to the job templates. Permissions let the IT administrator control who can use each particular template to submit jobs.

1. In the **Job Templates** view, highlight **New Job Template 1** and click **Set Permission** in the **Actions** pane on the right.
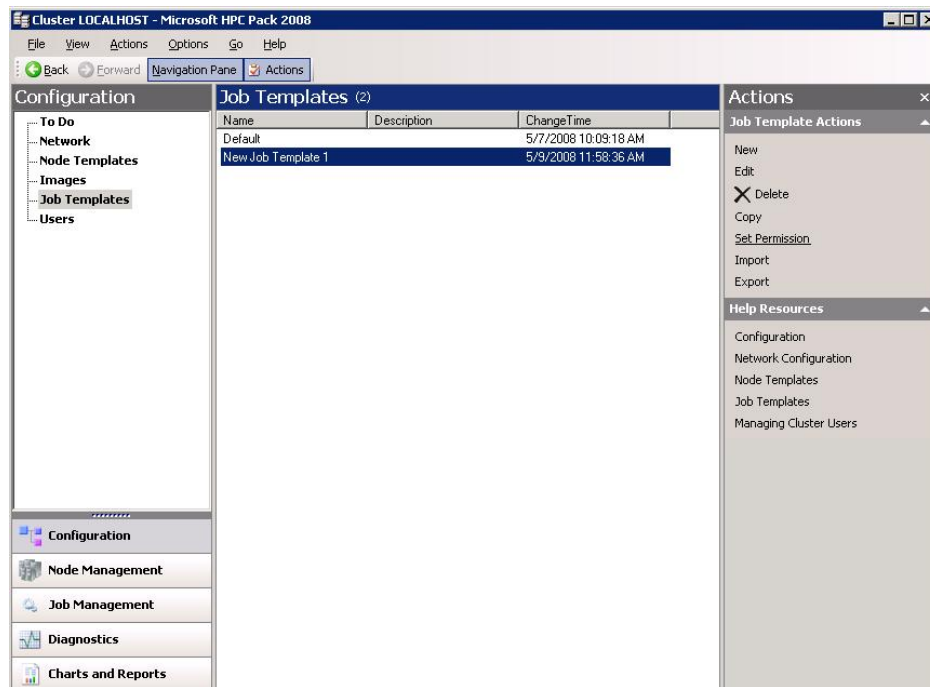
Figure 9 Jobs Template view

2. The **Permissions for New Job Template 1** dialog box appears. Here you can choose the permissions to grant or deny for the different groups of users. Click **OK**, and then click **Apply**.
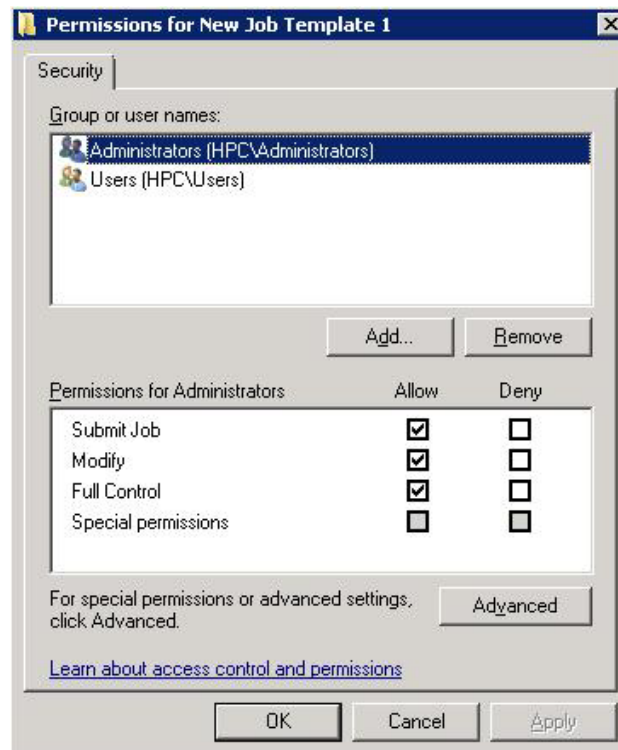
Figure 10 Set permissions

Note: For each user or group, the IT administrator can specify privileges: **Submit Job**, **Modify**, **Full Control,** or **Special Permissions**. For normal users, **Submit Jobs** is sufficient. The **Modify** privilege lets the IT administrator delegate the management of the template to the selected user or group. The **Full Control** option should be reserved for the cluster administrator, because this privilege allows the user to remove the job template altogether.

For more information about defining permissions with ACLs, see: http://technet2.microsoft.com/WindowsVista/en/library/ba1a3800-ce29-4f09-89ef-65bce923cdb51033.mspx?mfr=true or click on the **Learn about access control and permissions** link at the bottom of the dialog box.

In this example, we have created and set permissions on a basic template. Now we explain how an IT administrator can use templates to streamline the management of the job submission process.

How Does an IT Administrator Use Job Templates to Address Common Scenarios?

Using a Job Template to Constrain Resource Allocation

Job templates bind the user's processing needs and the organization's resource allocation policies together. With job templates, the IT administrator can enforce and mandate the delivery of resources to user groups in ways that support the organization's productivity goals. For example, administrators can use job templates to require that:
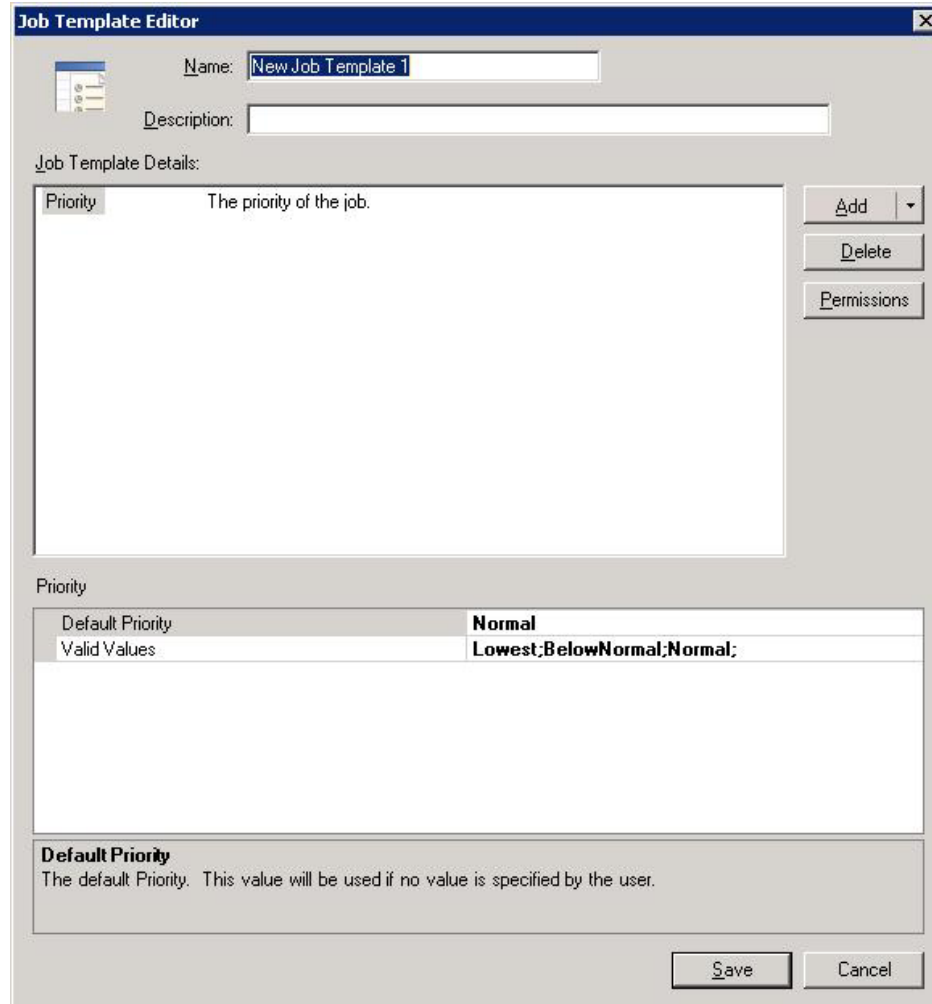
- All jobs share nodes (run non-exclusively).

- Run-time limits are enforced on all the jobs.

- Nodes are partitioned and divided among projects in particular ways that reflect the processing deadline constraints or organizational structure.

Suppose an organization has two groups of users: one group that owns the cluster and has mission-critical applications to run, and the rest of the users, who may occasionally have jobs they want to run on the cluster. The IT administrator can easily manage both of these groups—giving priority to the group that owns the cluster, while letting the others use the cluster for limited amounts of time when it doesn't interfere with the more important applications.

The IT administrator can create two templates—one for each group—giving the owners' group higher job priority and unlimited job run times. The IT administrator can also keep the other users from submitting jobs that require exclusive use of nodes, or that use more than a certain number of cores or sockets. With all of the options available, the IT administrator can enforce the company's resource allocation policy and let all users efficiently share the cluster with limited direct intervention.

Returning to our example, let's require that jobs using **New Job Template 1** share the nodes.

1. Launch the Administration Console, navigate to **Configuration**, and select **Job Templates**.
2. Select **New Job Template 1**, and click **Edit** in the **Actions** pane. The **Job Template Editor** dialog box appears.

Figure 11 Job Templates Editor dialog box

3. Click **Add**. The **Job Term Constraints** list, a listing of the job properties (or terms) appears. Note all of the options available.
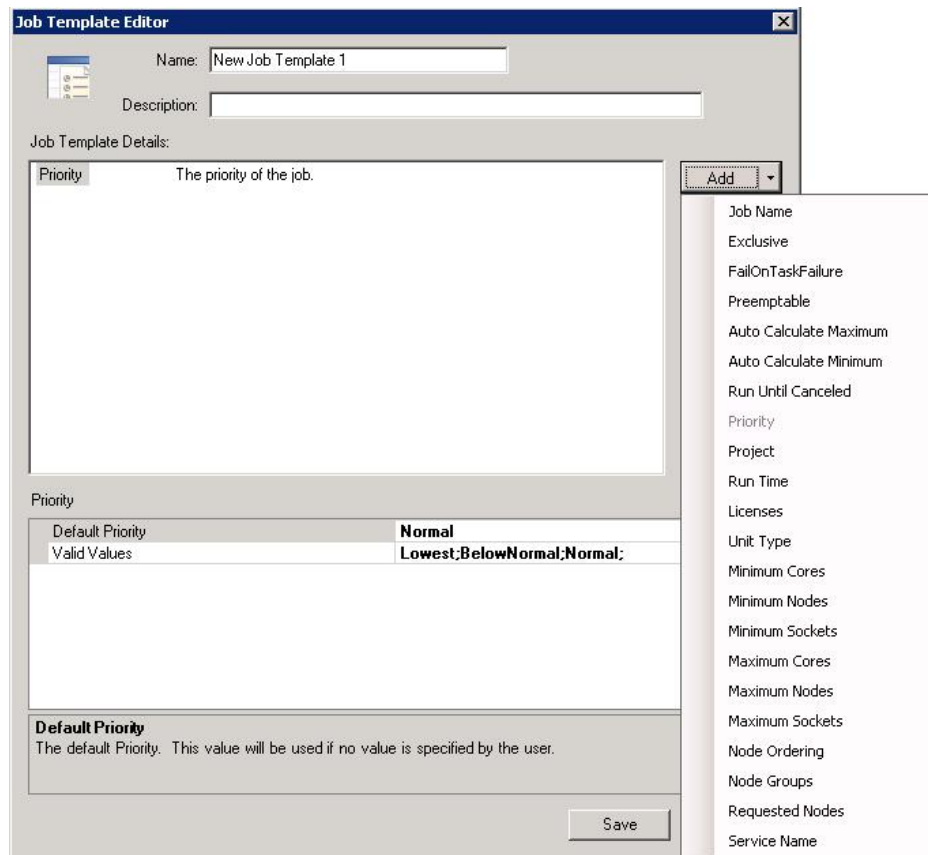
Figure 12 Job properties list

4. For this example, click **Exclusive**. The property appears under **Job Template Details** with a brief explanation of this property.

5. Highlight the **Exclusive** constraint. The default value and the valid values are listed in the lower pane.

Figure 13 Set a job property

6. Click **Valid Values**, and a drop-down menu appears. Leave only **False** selected, so that jobs submitted by **New Job Template 1** are always marked as non-exclusive, and then click **Save**. If a user tries to mark a job submitted with the template as exclusive, the job will fail.

Figure 14 Set the valid values

Now, all of the jobs using this template are forced to share their nodes unless they have reserved all of the nodes' processors. To let users override the default (False) mode, the administrator can select both **True** and **False** as the **Valid Values** for the **Exclusive** job property.

Using a Job Template to Prioritize Access to Shared Resources

A job template can help the IT administrator define the order in which different user groups access the shared resources. For example, suppose both simple and advanced jobs share the nodes in a node group. If the simple jobs are short and small (fewer cores required) and others are long and large (more cores required), the IT administrator may want to give the simple, short jobs higher priority access to compute resources , allowing faster design iterations at the early phases.

Let's look again at the previous example.

1. Navigate to **Configuration**, and then select **Job Templates**.
2. Select **New Job Template 1**, and then click **Edit** in the **Actions** pane.
3. In the **Job Template Editor**, click **Add**, and then click **Priority**. Note that the **Priority** property is added to the **Job Term Constraints** list. If **Priority** is already listed, you need only highlight it.
4. Click the combo box associated with the Default Priority property, and select **AboveNormal**. Click **Save**. Now, jobs submitted using **New Job Template 1** have **AboveNormal** priority by default, and thus they are scheduled before any jobs submitted with **Normal** priority.

Figure 15 Set the default

5. Click the combo box associated with the Valid Values property, and ensure that **Lowest**, **BelowNormal**, **Normal**, and **AboveNormal** are all checked, and then click **Save**. This lets users of this template submit jobs with priorities lower than the default if they desire.

Figure 16 Set the valid values

Using a Job Template to Constrain a Group of Users to a Particular Node Group

IT administrators can also use a job template to limit jobs to a subset of nodes in the cluster. The job therefore is only submitted to that particular node group.

1. From the **Start** menu, open the **HPC Cluster Manager**, and navigate to **Node Management**. You can see a listing of your compute nodes, head node, and WCF broker nodes using the navigation tree in the left pane. We start by creating a node group.

2. Highlight and right-click on the nodes you would like to place in the node group, select **Groups**, and then select **New Group**.
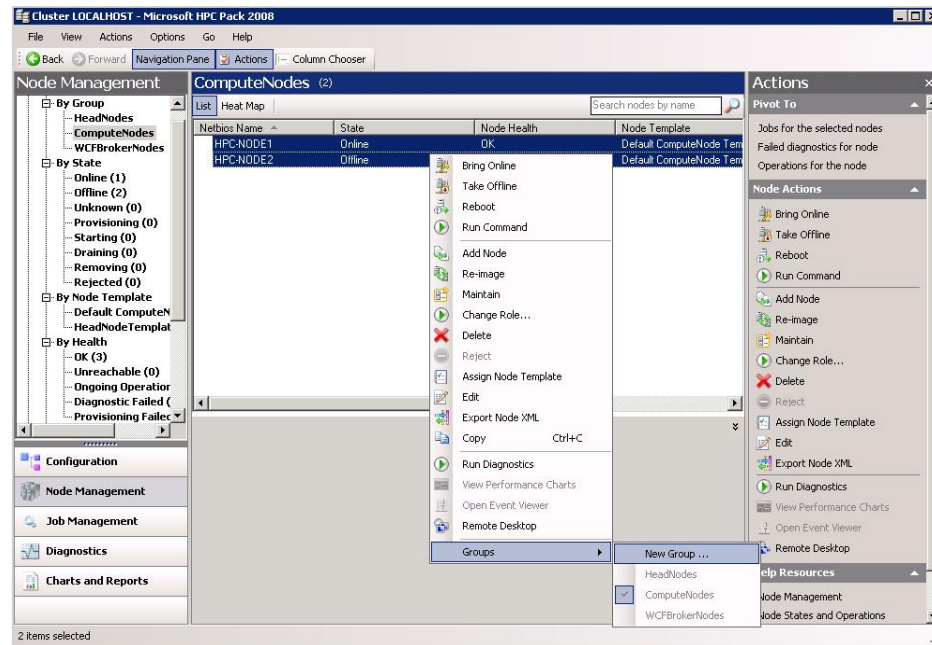
Figure 17 Add a new group

3. Provide a name for the node group and add a description, if desired. In this example, we name the node group **Special Nodes**.
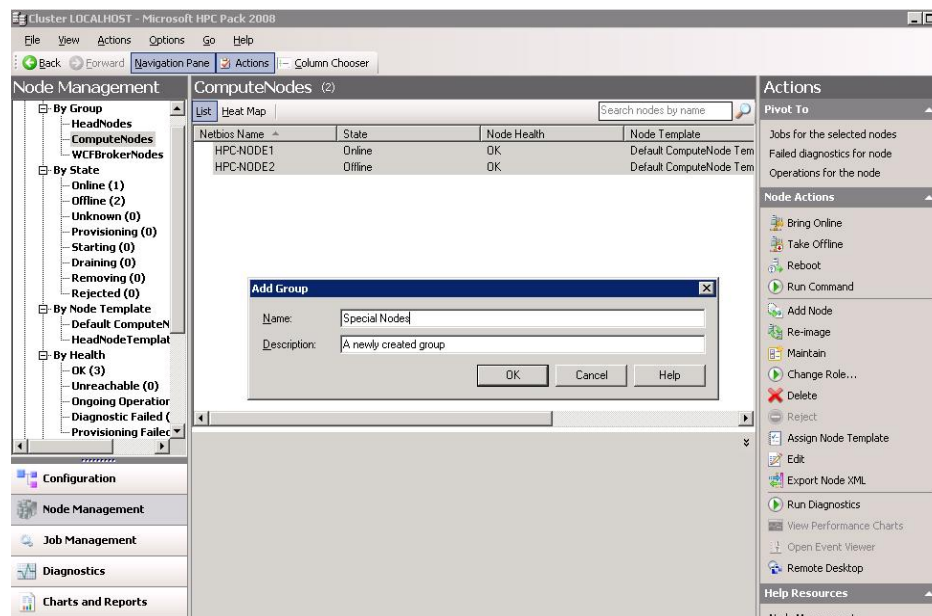
Figure 18 Add a Special Nodes group

4. Return to the **HPC Cluster Manager** and navigate to **Configuration**. Select **Job Templates** in the left pane.

5. We now edit the job template we created previously. Highlight **New Job Template 1**, and click on **Edit** in the **Actions** pane on the right.
6. Click on the **Add** drop-down menu. A list of options appears. Click on **Node Groups**.
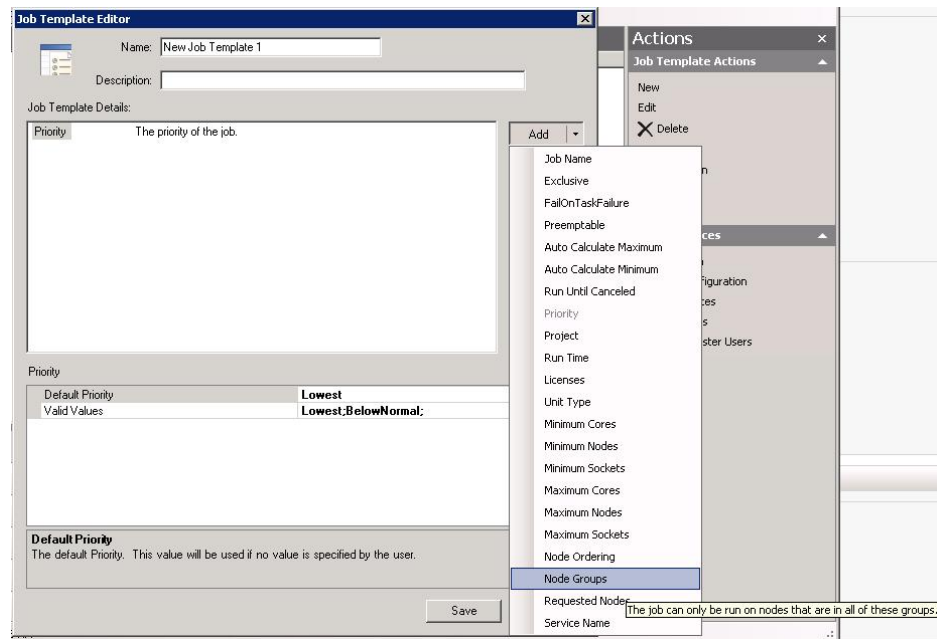
Figure 19 Add the node group

7. In the lower **NodeGroups** pane, click on **DefaultValue**, and then click on the drop-down arrow. Check the **Special Nodes** box, then click **Save**. Click on **Required Values**, click on the **drop-down arrow**, and check the **Special Nodes** box. Click **Save**.

    With **Special Nodes** in the list of **Required Values**, users submitting jobs using this template must always specify the **Special Nodes** group (along with any other groups that they would like to specify).
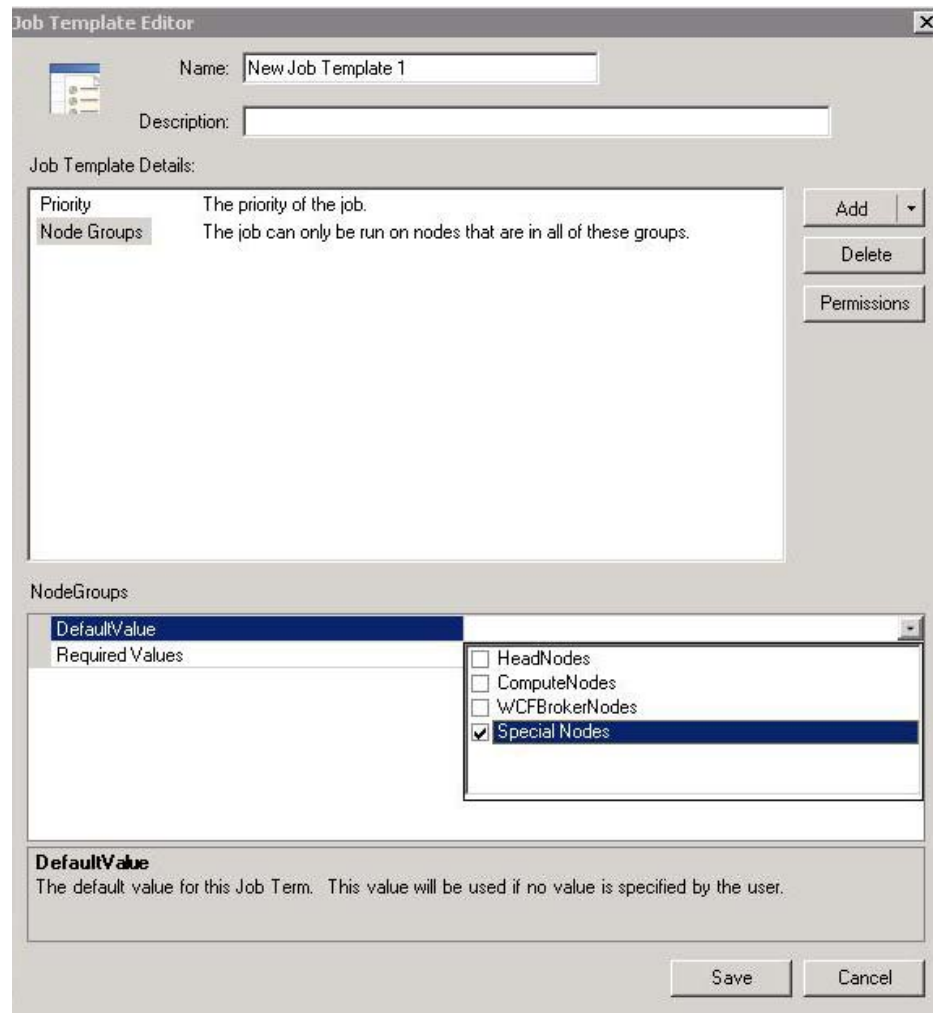
Figure 20 Set the default values

8. It is also possible to create a Windows user group on the domain controller; this user group, called **Special Users**, is constrained to the **Special Nodes** node group.

9. We now set permissions for the job template. Open the **HPC Cluster Manager**, and navigate to **Configuration**. Select **Job Templates** in the left navigation pane.

10. Right-click on **New Job Template 1**, and select **Set permission** (or click on **Set Permission** in the right **Actions** pane.)

11. Add an access control list (ACL) entry for the **Special Users** group. Click **Add**, and type **Special Users**. Click **OK**.
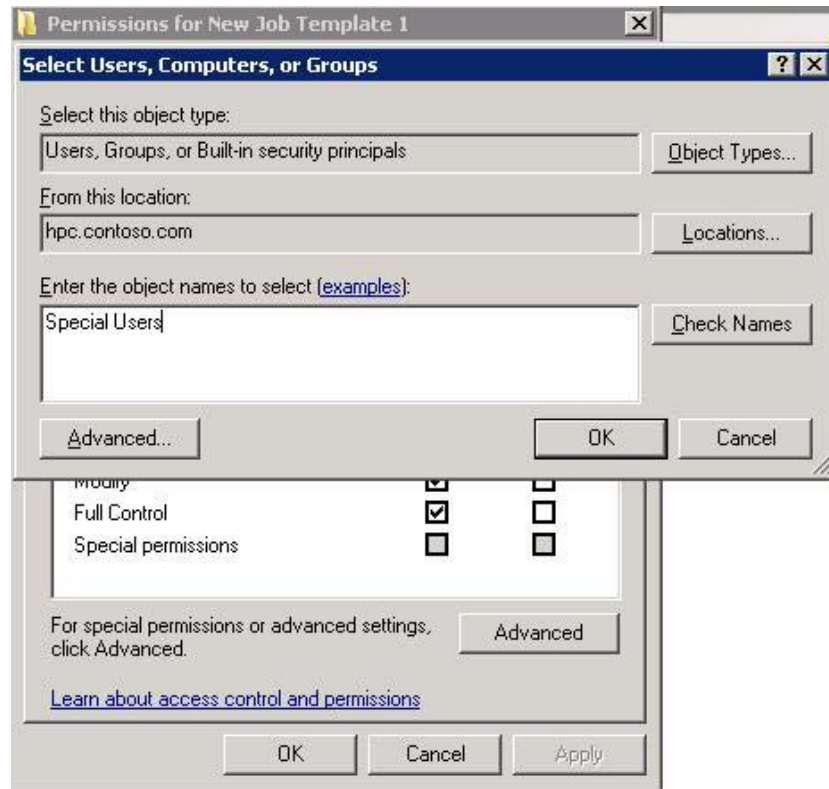
Figure 21 Add an ACL entry for Special Users

12. Verify that the box in the **Allow** column is checked so that users in the **Special Users** group have permission to submit jobs using **New Job Template 1**.

13. If you want users to submit jobs using *only* this template, you can deny permissions on the other job templates.

   Right-click on the other job templates that you want to restrict users from, and add an ACL entry for the **Special Users** group as described in step 11. Deny **Submit Job** permissions by checking the box in the **Deny** column, and then click **OK**.
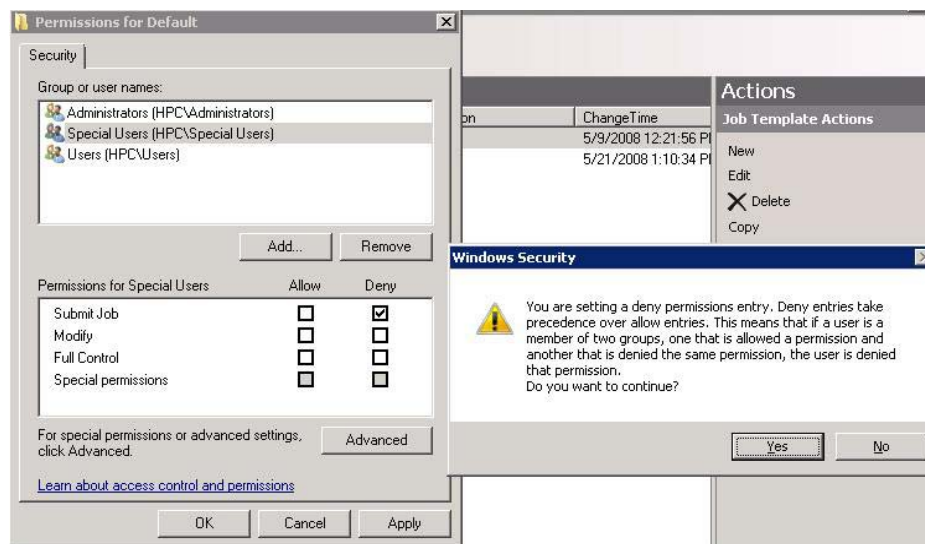
Figure 22 Deny permissions

We have now created a template that constrains users to certain node groups and certain templates. Now we look at the default template that is provided with Windows HPC Server 2008.

Using the Default Template

With the Windows HPC Server 2008 Job Scheduler, all job properties are optional. If the user does not specify a job property, the default value for that property is pulled from the job template. This lets users focus on creating tasks for their jobs and setting the metadata pieces that are important to them, without worrying about the more complex workings of the Job Scheduler.

All Windows HPC Server 2008 clusters also have a preconfigured default job template (named **Default**) that is accessible to all cluster users. If a job template is not specified at the time of submission, the default job template is used. It is important to note that a user can access all of the nodes in the cluster with the default job template. If this is not desirable, the administrator can restrict the nodes that are accessible. If there are no applications that require the default template, the administrator can consider revoking the **Submit Job** privileges to the default job template.

The **Job Template Editor** for the **Default** job template is shown in Figure 23. You can click on each property under **Job Template Details** to view the defaults. For example, in the **Default** template, the minimum number of cores on which a job may run is 1, the maximum is 2147483647, and the default (if no value is supplied by the user) is 1 core.

Figure 23 Default job template

How Is the Job Template Enforced?

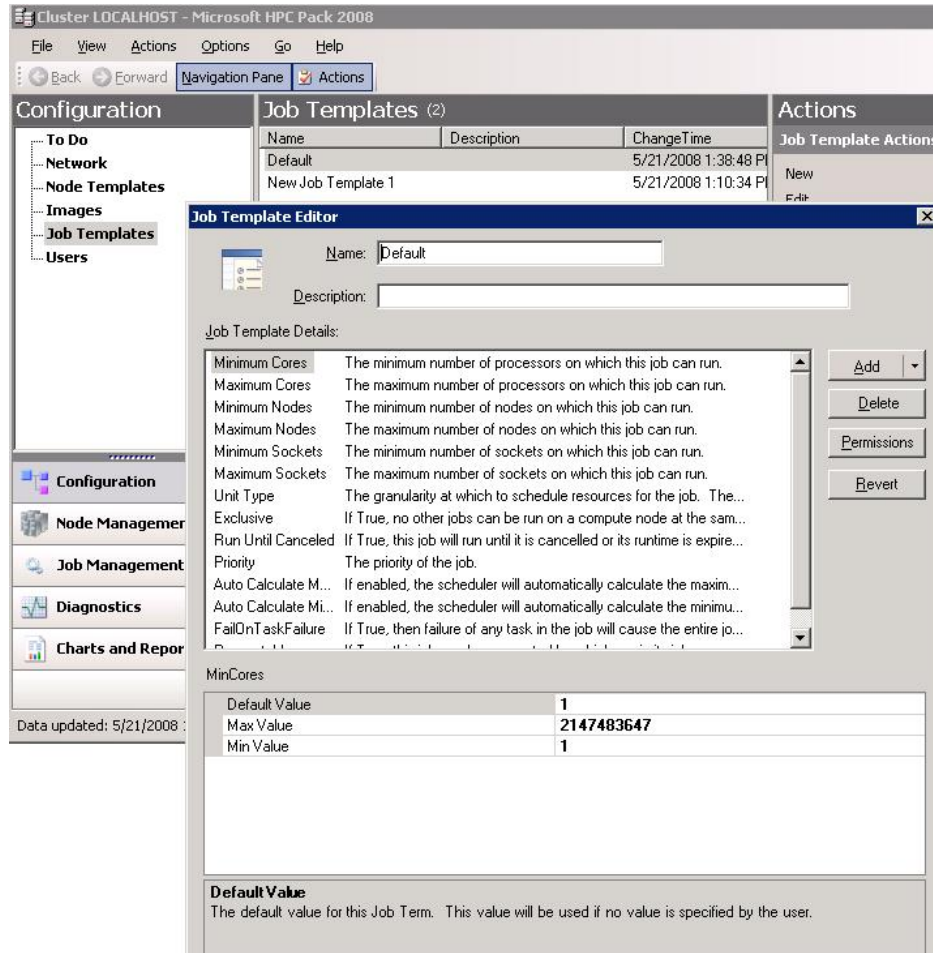Once defined, job templates work by letting the user define properties and by applying the administrator-defined defaults and constraints for properties that the user does not address for every job that comes into the system at submission time. This process is shown in Figure 27:
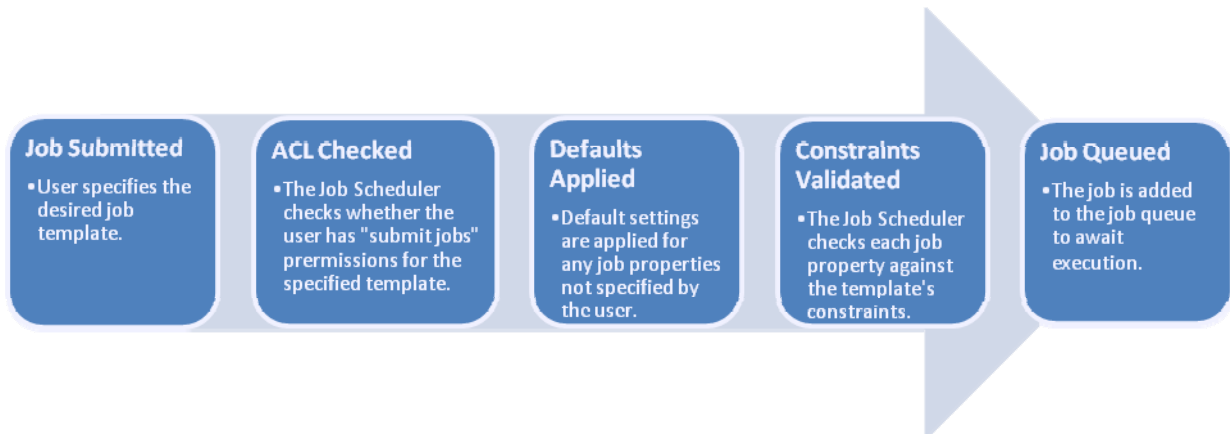
Figure 24 Job template enforcement

In the first step, a user can specify a job template when submitting a job through any of the various scheduler interfaces, including the command line, Windows PowerShell, APIs, or the HPC Job Manager. The job also contains its tasks and any job properties that the user would like to submit.

Next, the Job Scheduler confirms that the user submitting the job has the **Submit jobs** privilege for the specified template; if not, the job is rejected and submission fails. This is known as the ACL check, and it lets the IT administrator control which sets of users can submit which types of jobs.

For any job properties that the user has not specified, default settings are applied from the specified job template. These defaults are set by the IT administrator when the template is created. If a job property is not specified by the job or by the job template, the value from the default job template is used.

The Job Scheduler checks each job property against the constraints specified in the job template. If any job property falls outside of the valid range specified in the template, the job is rejected and the submission fails.

Finally, if the job has passed the above checks, it is added to the job queue to await execution.

How Does the User Submit Jobs with a Job Template?

The user can easily use a job template to submit a job to the Job Scheduler—users only need to understand the job properties that are important to them.

Let's submit a job with the template created earlier:

1. From the HPC Job Manager (click **Job Management** in the **HPC Cluster Manager**), click **Create New Job** under **Job Submission** in the **Actions** pane.

2. On the **Create New Job** dialog box, provide a name for the job (in this case, **MyTask**) and select **New Job Template 1** using the **Job Template** drop-down box.

> **Note:** Once a job template is specified, the UI adapts to this change. The **Priority** drop-down menu, for example, only shows the priority levels that are valid under the new template, and the run-time setting cannot be raised above the defined maximum.

Figure 25 Using the job template

3. Click **Task List**, and then click **Add**.
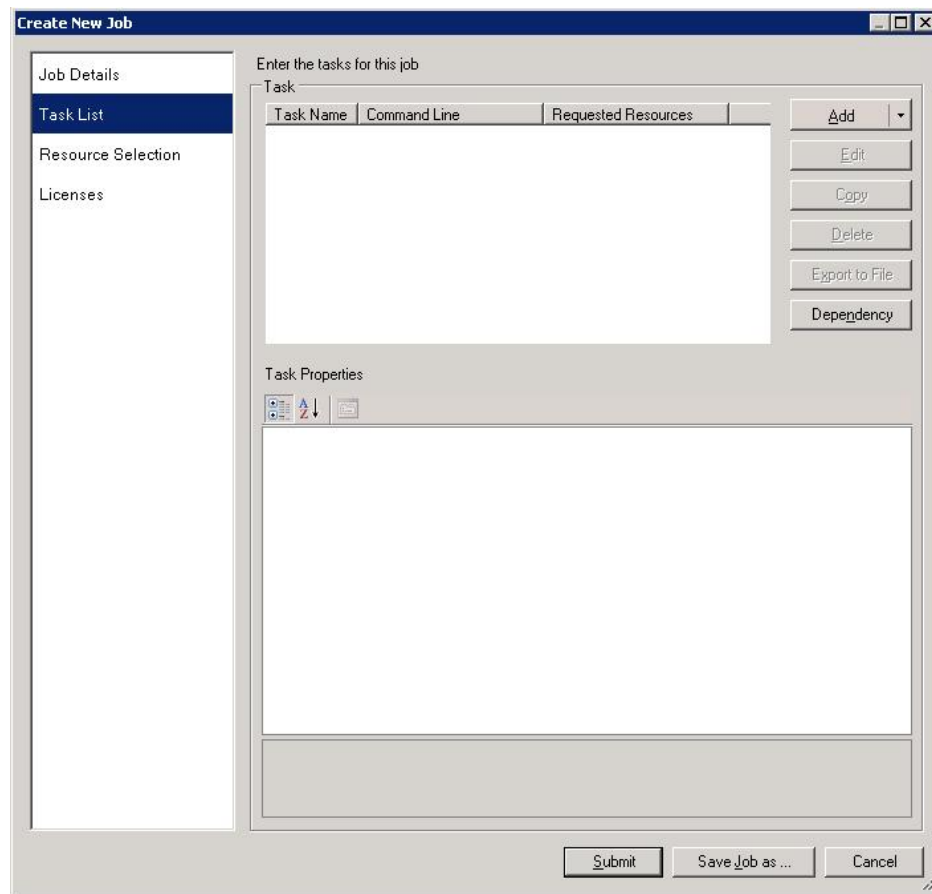
Figure 26 Add tasks

4. In the **Task Details and I/O Redirection** dialog box, type the name of the task (here, a task for the job **MyTask** is named **MyTask1**), and type the task at the command line (here, **MyApp.exe**). Click **Save**.

Figure 27 List the tasks

5. The task and command line will be listed on the **Create New Job** page.

Figure 28 List of tasks

6. Click **Resource Selection.** Here you can select the resources that you would like used for the job. Selecting a node group under **Node preferences** filters the nodes available for the job in the node selection list. Entering a hardware preference limits the node groups that you have selected to those that meet the specified hardware preferences. Click **Submit**.

Figure 29 Resource Selection page

Job templates can be used to easily submit single-task jobs or parametric sweep jobs. Users do not need to concern themselves with the resource requirements, cluster resources, or other aspects of the cluster workings—all of the job properties are determined in the job template. Users can instead focus on their applications and the results of their jobs.

Figure 30 shows the submission of a single-task job (**MyTask1**) with **New Job Template 1**.

Figure 30 Submit single-task jobs

Figure 31 shows the submission of a parametric sweep job with **New Job Template 1**.

> **Note:** Parametric sweep jobs consist of multiple instances of the same program (usually serial) running concurrently. With parametric sweep jobs, there is no communication or interdependency among the tasks; typically, the parallelization is performed by the Job Scheduler, based on the fact that all of the tasks are in the same job.

Figure 31 Submit parametric sweep jobs

Conclusion: Powerful and Easy to Use

Windows HPC Server 2008 provides job templates to help streamline the job submission process. For the IT administrator, job templates provide a way to manage the job submission process, letting the administrator easily partition the cluster, prioritize the jobs, or limit the resources used by jobs. For the user, job templates provide a way to submit a job without having to learn the Job Scheduler nomenclature. Job templates help make it easy to conform to the organization's resource allocation policies. They are powerful and easy to use—and save time and effort for both users and IT administrators.

Appendix

A wide range of job property (or job term) options is available to administrators to specify when creating a job template. The following table shows some of these options:

Table 1 Job and Task Properties

Job Property	Job, Task, or Sweep Task Property	Description
Auto calculation	Job	If enabled, the scheduler will automatically calculate the <Minimum/Maximum> based on the tasks in the job.
Command line	Task	Use to specify the command that will be executed by the task.
Environmental variables	Task	Use to set variables in the runtime environment (separate by commas in the format: name1=value1).
Exclusive	Job or Task	When set to True, no other job/task may access the node while this job/task is running.
Fail on task failure	Job	When set to True, job fails if any task within it fails.
Job ID	Job	Use to give a unique numeric ID for the job.
Job name	Job	Use to give a unique name of the job.
Licenses	Job	Use to specify licenses for the software running as part of a job.
Minimum (or maximum) cores	Job or Task	Use to specify the minimum (or maximum) number of cores for the job/task (each node may contain one or more sockets, which may contain one or more cores).
Minimum memory	Job	Use to specify the minimum amount of memory (in MB) that must be present on any nodes for the job to run.
Minimum (or maximum) nodes	Job or Task	Use to specify the minimum (or maximum) number of nodes for the job/task (each node may contain one or more sockets, which may contain one or more cores).
Minimum (or maximum) sockets	Job or Task	Use to specify the minimum (or maximum) number of core sockets for the job/task (each socket may contain one or more cores).

Node groups	Job	Use to set specific node group(s) on which to run the job.
Node ordering	Job	Use to set the specific core or memory allocations for your job without pre-determining specific nodes, number of sockets, or available memory.
Preemptable	Job	When set to True, the job may be preempted by a job assigned a higher priority.
Priority	Job	Use to establish job priority compared to other jobs (default is Normal).
Project	Job	Use to associate the job with a project.
Requested nodes	Job	Use to specify the nodes on which the job can run.
Required nodes	Task	Use to list the nodes that must be assigned to this task and its job in order to run.
Rerunnable	Task	When set to True, the Job Scheduler will attempt to rerun the task if it fails. If set to False, the task will fail after the first run attempt fails.
Run time	Job or Task	Use to limit amount of time a job/task will run.
Run until cancelled	Job	When set to True, the job runs until it is cancelled or run time expires.
Standard input/output	Task	Use to set the path (relative to the working directory) to the file to which the *stdout*/*stderr* of this task should be written.
Sweep increment	Sweep Task	Use to set the amount to increment the sweep index at each step of the sweep.
Sweep start/end index	Sweep Task	Use to set the starting/ending index for the sweep.
Task ID	Task	Use to give a unique numeric ID for the task.
Task name	Task	Use to give a unique name of the task.
Unit type	Job	Use to set the granularity for scheduling resources (minimum and maximum specify the number of resources required).
Working directory	Task	Use to set the working directory to be used during execution for this task.

Glossary

The following terminology is helpful when running Windows HPC Server 2008.

Administration Console

The Administration Console in Windows HPC Server 2008 is the overall management interface for cluster administration. Based on the Microsoft System Center user interface, it uses navigation bars to quickly change the context and view. The Job Manager Navigation Bar provides a graphical interface to job management and scheduling.

Cluster

A cluster is the top-level unit of Windows HPC Server. A cluster contains the following elements:

- **Node:** A single physical or logical computer with one or more processors. Nodes can be a head node, compute nodes, or WCF Broker nodes.

- **Queue:** An element providing queuing and job scheduling. Each Windows HPC Server cluster contains only one queue, and that queue contains pending jobs. Completed jobs are purged periodically from the queue.

- **Job:** A collection of tasks that a user initiates. Jobs are used to reserve resources for subsequent use by one or more tasks. Users can submit jobs in either interactive or batch processes.

- **Tasks:** A task represents the execution of a program on given compute nodes. A task can be a serial program (single process), or a parallel program (using multi-threading, OpenMP, or MPI).

Job Scheduler

The Job Scheduler queues jobs and their tasks. It allocates resources to these jobs; initiates the tasks on the compute nodes of the cluster; and monitors the status of jobs, tasks, and compute nodes. Job scheduling uses scheduling policies to decide how to allocate resources.

- The interface layer provides for job and task submission, manipulation, and monitoring services accessible through various entry points.

- The scheduling layer provides a decision-making mechanism that balances supply and demand by applying scheduling policies. The workload is distributed across available nodes in the cluster according to the job profile.

- The execution layer provides the workspace used by tasks. This layer creates and monitors the job execution environment and releases the resources assigned to the task upon task completion. The execution environment supplies the workspace customization for the task, including environment variables, scratch disk settings, security context, and execution integrity in addition to application-specific launching and recovery mechanisms.

Navigation Buttons

A set of buttons at the lower left of the Administration Console that shift the view and context to different areas of Windows HPC Server 2008 management and administration. For example, clicking the Job Management Navigation button opens the Job Manager.

Node Manager

The job agent and authorization service on the compute node. The Node Manager also starts the job on the node.

Scheduling Policies

Windows HPC Server uses nine scheduling policies:

- Priority-based first come, first served (FCFS)
- Backfilling
- Exclusive scheduling
- Resource matchmaking
- Job template
- Multilevel compute resource allocation (MCRA)
- Pre-emption
- Adaptive allocation (grow/shrink)

Task Execution

Windows HPC Server 2008 has two types of tasks—basic tasks and parametric tasks.

A basic task uses a single command line that includes the run command, along with metadata that describes how to run the command. A basic task can be a parallel task and can be run across multiple nodes or cores. Parallel tasks typically communicate with other parallel tasks in the job using the Microsoft Message Passing Interface (MS-MPI), or through shared memory when running on multiple cores on a single node.

A parametric task contains a command line with wildcards, letting the same task run many times with different inputs for each step. A parametric task can be a parallel task and can be run across multiple nodes or cores.

Windows® Communication Foundation (WCF) Broker

Stores and forwards request/response messages between client application and service instances.

Windows HPC Server 2008

For those seeking productive solutions for high performance computing, Windows HPC Server 2008 provides a comprehensive platform built on Windows Server 2008 that helps to simplify deployment, management, and integration with existing infrastructure, thus helping to improve the productivity of your system administrators, application developers, and users. Windows HPC Server 2008 unites the power of commodity x64-based computers, the security of Active Directory®, and the Windows Server 2008 operating system to provide an affordable, easy-to-use, and scalable HPC solution. Windows HPC Server 2008 uses Node Templates to help simplify and speed deployment of compute nodes using standard Windows Server 2008 deployment technologies. Additional compute nodes can be added to a cluster by simply connecting computers to the network. The Microsoft Message Passing Interface implementation is compatible with the reference MPICH2 and uses high-speed Network Direct drivers. Integration with Active Directory helps enable role-based security for administration and users, and System Center user interface model provides a familiar administrative and scheduling interface. The Windows HPC Server 2008 Job Scheduler supports heterogeneous clusters and enables the use of Service-Oriented Architecture applications on the cluster.