HPCC Systems®

Installing & Running the HPCC Platform

Boca Raton Documentation Team

LexisNexis® Risk Solutions

Installing & Running the HPCC Platform

Boca Raton Documentation Team

We welcome your comments and feedback about this document via email to <docfeedback@hpccsystems.com>

Please include **Documentation Feedback** in the subject line and reference the document name, page numbers, and current Version Number in the text of the message.

2015 Version 5.4.2-1

Welcome

These instructions will guide you through installing and running the HPCC[1] Community Edition on a single node to start and then optionally, expand it to a larger cluster of nodes.

The HPCC Thor technology is designed to effectively process, analyze, and find links and associations within high volumes of complex data. This can detect non-obvious relationships, scale to support petabytes of data, and is significantly faster than competing technologies while requiring less hardware and resources.

The HPCC Roxie technology - also known as the Rapid Data Delivery Engine or RDDE - uses a combination of technologies and techniques that produce extremely fast throughput for queries on indexed data.

This translates into better quality answers in less time so that organizations can cope with massive data and efficiently turn information into knowledge.

We suggest reading this document in its entirety before beginning. The entire process can take an hour or two, depending on your download speed.

[1] **H**igh **P**erformance **C**omputing **C**luster (HPCC) is a massively parallel processing computing platform that solves Big Data problems. See http://hpccsystems.com/Why-HPCC/How-it-works for more details.

Quick Start Guide

We recommend taking the time to read this manual in its entirety; however, the following is a quick start summary of steps. There are many aspects of the HPCC System platform and this guide is intended to help you get the most out of your system. This section is not intended to replace the more comprehensive material in the remainder of this book.

1. Install HPCC.

   Download the installation package from http://hpccsystems.com/download/free-community-edition and install.

   On CentOS/Red Hat:

   ```
   sudo rpm -Uvh <rpm file name>
   ```

   On Ubuntu/Debian:

   ```
   sudo dpkg -i <deb filename>
   ```

2. Start your HPCC System.

   On CentOS/Red Hat:

   ```
   sudo /sbin/service hpcc-init start
   ```

   Ubuntu:

   ```
   sudo service hpcc-init start
   ```

   Debian 6 (Squeeze):

   ```
   sudo /etc/init.d/hpcc-init start
   ```

3. Run **ECL Watch**. Check out your system.

   Using a browser, go to **ECL Watch** running on port 8010 of your HPCC Node.

   For example, http://nnn.nnn.nnn.nnn:8010, where nnn.nnn.nnn.nnn is your node's IP address.

4. Create and run some ECL.

   You can do this right from ECL Watch, click on the Playground link.

5. Get and install the ECL IDE and Client tools.

Now What?

Now that you have HPCC started and running, what do you want to do? Maybe evaluate your needs and proceed to develop a custom configuration suitable for those needs. Maybe you want to expand your system and add nodes. Those topics and several others are covered in the following sections.

To familiarize yourself with what your system can do we recommend following the steps in:

- The **HPCC Data Tutorial**
- **The Six Degrees of Kevin Bacon** example
- Read **Using Config Manager** to learn how to configure an HPCC platform using Advanced View.
- Use your new skills to process your own massive dataset!

Hardware and Software Requirements

The following section describes the various hardware and software required in order to run the HPCC.

Network Switch

A significant component of HPCC is the infrastructure it runs on, specifically the switch.

Switch requirements

- Sufficient number of ports to allow all nodes to be connected directly to it;
- IGMP v.2 support
- IGMP snooping support

Small: For a very small test system, almost any gigabit switch will suffice. These are inexpensive and readily available in six to 20-port models.

Figure 1. 1 GigE 8-port Switch

Medium: For medium sized (10-48 node) systems, we recommend using a Force10 s25, s50, s55, or s60 switch

Figure 2. Force10 S55 48-port Network Switch

Large: For large (48-350 node) system, the Force10 c150 or c300 are good choices.

Figure 3. Force 10 c150

Very Large: For very large (more than 300 nodes) system, the Force10 e600 or e1200 are good choices.

Figure 4. Force 10 e600 and e1200

Switch additional recommended features

- Non-blocking backplane
- Low latency (under 35usec)
- Layer 3 switching
- Managed and monitored (SNMP is a plus)
- Port channel (port bundling) support

Load Balancer

In order to take full advantage of a Roxie cluster, a load balancer is required. Each Roxie Node is capable of receiving requests and returning results. Therefore, a load balancer distributes the load in an efficient manner to get the best performance and avoid a potential bottleneck.

We recommend the Web Accelerator product line from F5 Networks. See http://www.f5.com/pdf/products/big-ip-webaccelerator-ds.pdf for more information.

Figure 5. F5 Load Balancers

Load Balancer Requirements

Minimum requirements

- Throughput: 1Gbps Gigabit
- Ethernet ports: 2
- Balancing Strategy: Round Robin

Standard requirements

- Throughput: 8Gbps
- Gigabit Ethernet ports: 4
- Balancing Strategy: Flexible (F5 iRules or equivalent)

Recommended capabilities

- Ability to provide cyclic load rotation (not load balancing).
- Ability to forward SOAP/HTTP traffic
- Ability to provide triangulation/n-path routing (traffic incoming through the load balancer to the node, replies sent out the via the switch).
- Ability to treat a cluster of nodes as a single entity (for load balancing clusters not nodes)

  or
- Ability to stack or tier the load balancers for multiple levels if not.

Nodes-Hardware

The HPCC can run as a single node system or a multi node system.

These hardware recommendations are intended for a multi-node production system. A test system can use less stringent specifications. Also, while it is easier to manage a system where all nodes are identical, this is not required. However, it is important to note that your system will only run as fast as its slowest node.

Node minimum requirements

- Pentium 4 or newer CPU
- 32-bit
- 1GB RAM per slave

  (Note: If you configure more than 1 slave per node, memory is shared. For example, if you want 2 slaves per node with each having 4 GB of memory, the server would need 8 GB total.)
- One Hard Drive (with sufficient free space to handle the size of the data you plan to process) or Network Attached Storage.
- 1 GigE network interface

Node recommended specifications

- Nehalem Core i7 CPU
- 64-bit
- 4 GB RAM (or more) per slave
- 1 GigE network interface
- PXE boot support in BIOS

  PXE boot support is recommended so you can manage OS, packages, and other settings when you have a large system
- Optionally IPMI and KVM over IP support

For Roxie nodes:

- Two 10K RPM (or faster) SAS Hard Drives

  Typically, drive speed is the priority for Roxie nodes

For Thor nodes:

- Two 7200K RPM (or faster) SATA Hard Drives (Thor)
- Optionally 3 or more hard drives can be configured in a RAID 5 container for increased performance and availability

  Typically, drive capacity is the priority for Thor nodes

Nodes-Software

All nodes must have the identical operating systems. We recommend all nodes have identical BIOS settings, and packages installed. This significantly reduces variables when troubleshooting. It is easier to manage a system where all nodes are identical, but this is not required.

Operating System Requirements

Binary packages are available for the following:

- 64-bit CentOS 5
- 64-bit CentOS 6
- 64-bit CentOS 7
- 64-bit RedHat Enterprise 5
- 64-bit RedHat Enterprise 6
- 64-bit Ubuntu 12.04 (LTS)
- 64-bit Ubuntu 13.10
- 64-bit Ubuntu 14.04 (LTS)

Dependencies

Installing HPCC on your system depends on having required component packages installed on the system. The required dependencies can vary depending on your platform. In some cases the dependencies are included in the installation packages. In other instances the installation may fail, and the package management utility will prompt you for the required packages. Installation of these packages can vary depending on your platform. For details of the specific installation commands for obtaining and installing these packages, see the commands specific to your Operating System.

Note: For CentOS installations, the Fedora EPEL repository is required.

SSH Keys

The HPCC components use ssh keys to authenticate each other. This is required for communication between nodes. A script to generate keys has been provided .You should run that script and distribute the public and private keys to all nodes after you have installed the packages on all nodes, but before you configure a multi-node HPCC.

- As root (or sudo as shown below), generate a new key using this command:

```
sudo /opt/HPCCSystems/sbin/keygen.sh
```

- Distribute the keys to all nodes. From the **/home/hpcc/.ssh** directory, copy these three files to the same directory (**/home/hpcc/.ssh**) on each node:
  - **id_rsa**
  - **id_rsa.pub**
  - **authorized_keys**

  Make sure that files retain permissions when they are distributed. These keys need to be owned by the user "**hpcc**".

User Workstation Requirements

- Running the HPCC platform requires communication from your user workstation with a browser to the HPCC. You will use it to access ECL Watch—a Web-based interface to your HPCC system. ECL Watch enables you to examine and manage many aspects of the HPCC and allows you to see information about jobs you run, data files, and system metrics.

  Use one of the supported web browsers with Javascript enabled.
  - Internet Explorer® 9 (or later)
  - Firefox™ 3.0 (or later.)
  - Google Chrome 10 (or later)

  If browser security is set to **High**, you should add ECLWatch as a Trusted Site to allow Javascript execution.
- Install the ECL IDE

  The ECL IDE (Integrated Development Environment) is the tool used to create queries into your data and ECL files with which to build your queries.

  Download the ECL IDE from the HPCC Systems web portal. http://hpccsystems.com

  You can find the ECL IDE and Client Tools on this page using the following URL:

  http://hpccsystems.com/download/free-community-edition/ecl-ide

  The ECL IDE was designed to run on Windows machines. See the appendix for instructions on running on Linux workstations using Wine.
- Microsoft VS 2008 C++ compiler (either Express or Professional edition). This is needed if you are running Windows and want to compile queries locally. This allows you to compile and run ECL code on your Windows workstation.
- GCC. This is needed if you are running under Linux and want to compile queries locally on a standalone Linux machine, (although it may already be available to you since it usually comes with the operating system).

HPCC Installation and Startup

Follow these steps to install the packages and start components in a single-node configuration to begin. Once it is successfully installed, you will use the Configuration Manager to customize or expand your system.

Configuration Manager is the utility with which we configure the HPCC platform. It is run on your Linux Server and you access its interface using a browser.

Figure 6. System Overview: Thor

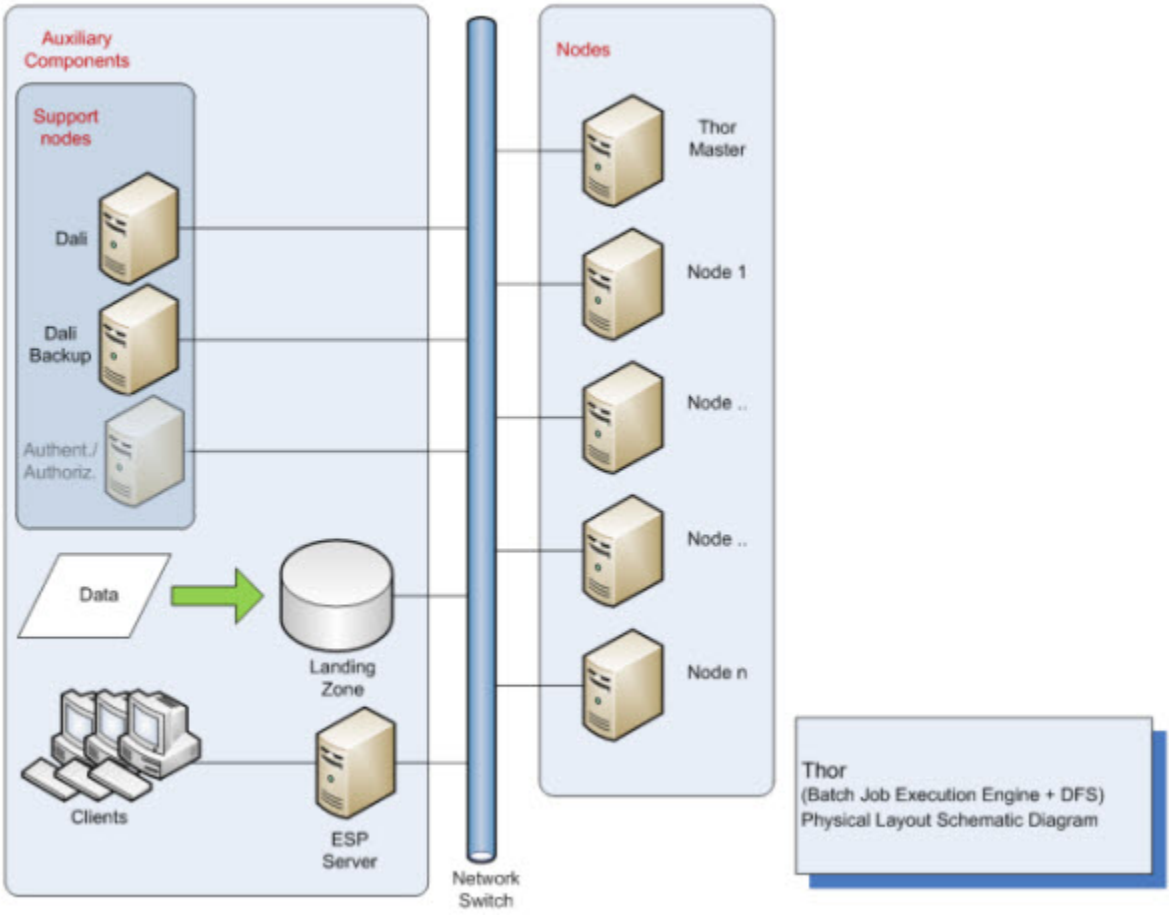
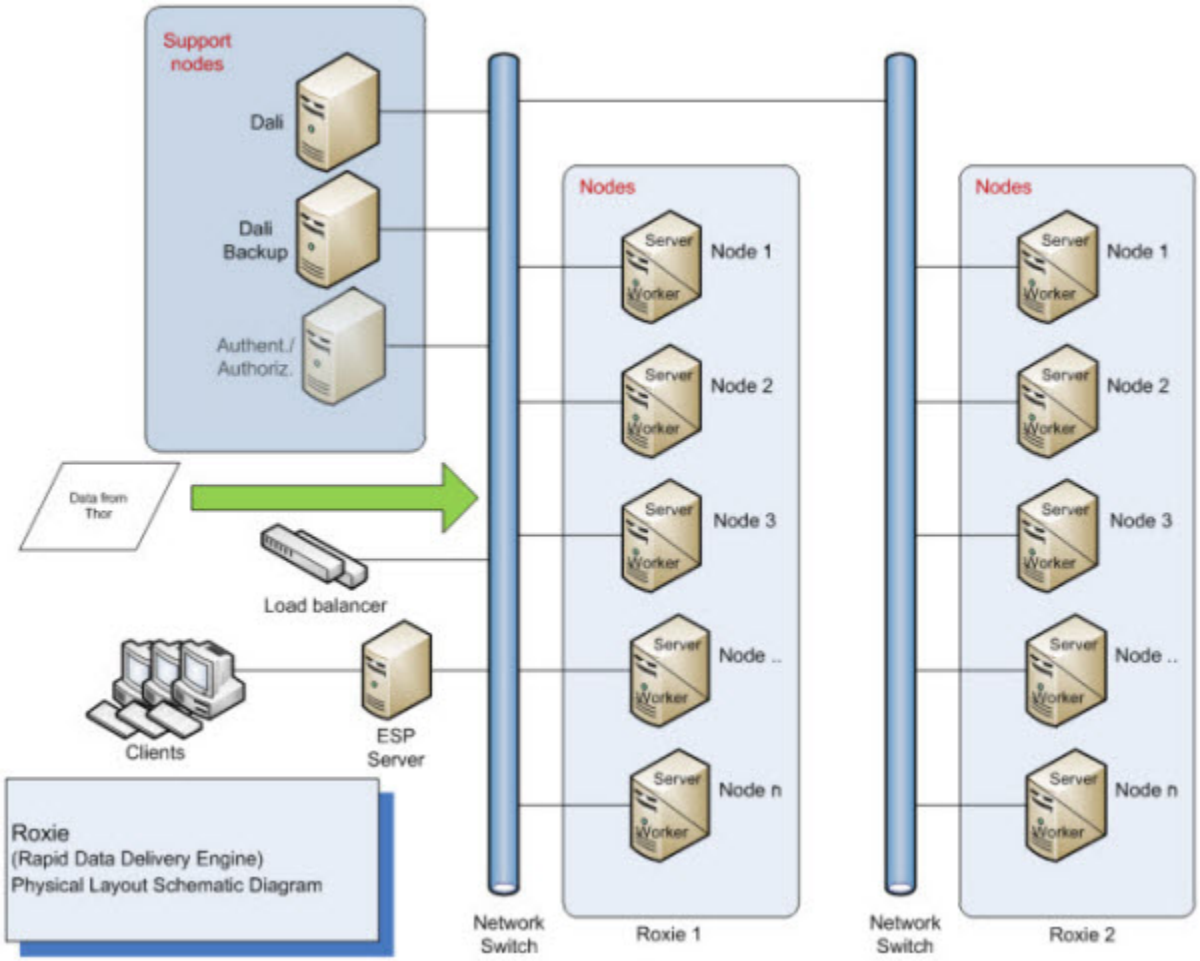


Figure 7. System Overview: Roxie



Initial Setup-Single Node

This section covers installing the HPCC on a single node. This will enable the HPCC system to operate successfully; however, the real strength of the HPCC is when it is run in a multi-node environment and can leverage the ability to perform operations using Massively Parallel Processing (MPP).

In addition, on a production system, you would dedicate one or more nodes to each server process. See the *Using Configruation Manager* manual for more details.

Installing the Package

The installation and package that you download is different depending on the operating system you plan to use. The installation packages will fail to install if their dependencies are missing from the target system.

Packages are available from the HPCC Systems® website: http://hpccsystems.com/download/free-community-edition

To install the package, follow the appropriate installation instructions:

CentOS/Red Hat

Install RPM with the -Uvh switch.

This is the upgrade command and will perform an automatic upgrade if a previous version is installed or it will just install fresh if no other version has been installed.

```
sudo rpm -Uvh <rpm file name>
```

Optional Plug-ins

For RPM based systems, there are two different installation packages available. One package includes the optional plug-ins to support embedded code from other languages, such as JAVA, JavaScript, R, or Python.

If you do not want support for other languages, choose the package for your distro that begins with:

```
 hpccsystems-platform_community-
```

If you want support for other languages, choose the package for your distro that begins with:

```
hpccsystems-platform_community-with-plugins-
```

You must install the packages that **have** the plug-ins using the --nodeps option. For example:

```
sudo rpm -Uvh --nodeps <rpm file name>
```

Then you must install the dependencies for each language you wish to support. The dependencies to support each language are installed separately.

The optional plug-ins are:

- Python : pyembed
- JAVA : jniembed
- JavaScript : v8embed
- R : Rembed

Ubuntu/Debian

For Ubuntu installations a Debian package is provided. To install the package, use:

```
sudo dpkg -i <deb filename>
```

Initial Startup

1. Start the system using the default configuration.

   Centos/Red Hat

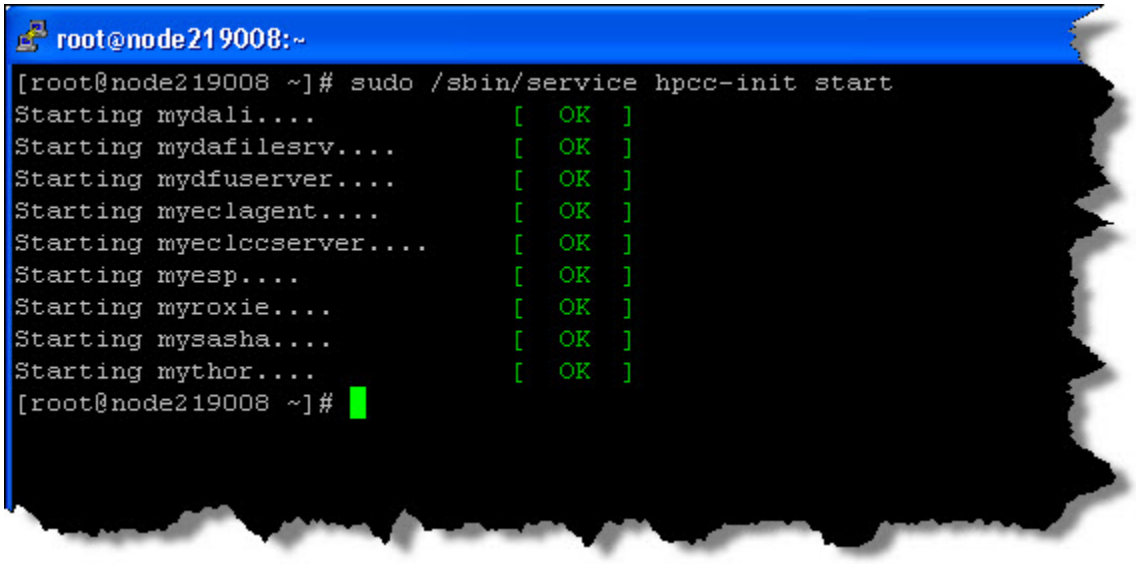
   ```
   sudo /sbin/service hpcc-init start
   ```

   Ubuntu

   ```
   sudo service hpcc-init start
   ```

   Debian 6 (Squeeze)

   ```
   sudo /etc/init.d/hpcc-init start
   ```



There are log files for each component in directories below **/var/log/HPCCSystems** (default location) including an hpcc-init log for the start up process. If any component fails to start, these logs can help in troubleshooting.

Running an ECL Query on your Single-Node System

The single node system is running, and you can now create and run some ECL[1] code using either ECL IDE, the command line ECL compiler, or the ECL Command line tool.

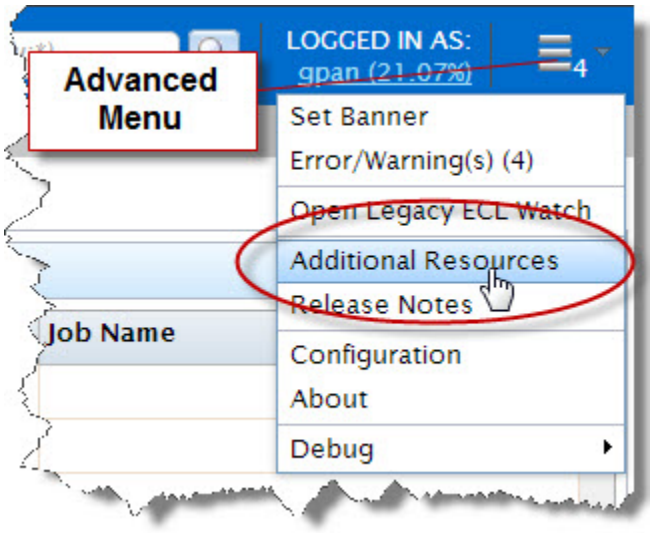
Install the ECL IDE and HPCC Client Tools

1. In your browser, go to the **ECL Watch** URL. For example, http://nnn.nnn.nnn.nnn:8010, where nnn.nnn.nnn.nnn is your node's IP address.

	Your IP address could be different from the ones provided in the example images. Please use the IP address of **your** node.

2. From the ECL Watch Advanced menu, select on the **Additional Resources** link.

Figure 8. ECL Watch Resource Page



Follow the link to the HPCC System's portal download page.

3. Click on the **ECL IDE** link. (on the right hand side in the Download column, under the Free Community Edition heading)

4. Follow the instructions on the web page to install the ECL IDE.

5. Install the ECL IDE, following the prompts in the installation program. Once the ECL IDE is installed successfully, you can proceed.

[1] **E**nterprise **C**ontrol **L**anguage (ECL) is a declarative, data centric programming language used to manage all aspects of the massive data joins, sorts, and builds that truly differentiate HPCC (High Performance Computing Cluster) from other technologies in its ability to provide flexible data analysis on a massive scale.

Running a basic ECL program

Now that the package is installed on your Linux node and ECL IDE is installed on your Windows workstation, you can run your first ECL program. ECL programs may be run locally or remotely. For larger ECL jobs, you will want to target a remote cluster of machines, which may not be running the same operating system as the machine you are working on.

In this section we will use the **ECL Command line interface** to the compiler to compile and run ECL code locally.

The ECL compiler (eclcc) installs on to the eclcc server node when a package is installed. :This should be in your path, so you can run it from anywhere on the server. It is also installed on a Windows machine when you install the ECL IDE. To compile and run on Windows, you also need the Visual Studio 2008 C++ compiler (see *User Workstation Requirements* for details).

1. Create a file called hello.ecl and type in the following text (including the quotes):

   ```
   output('Hello world');
   ```

   You can either use your favorite editor, or you can use the command line by typing the following

   ```
   echo "Output('Hello world');" > hello.ecl
   ```

2. Compile your program using eclcc by typing the following command:

   ```
   eclcc hello.ecl
   ```

3. An executable file is created which you can run as follows:

   ```
   # on a Linux machine:
      ./a.out
   # on a Windows machine:
      a.out
   ```

   This generates the output "Hello world" (excluding quotes), to the std output, your terminal window in this example. You can redirect or pipe the output to a file or program if you choose. This verifies that the compiler is working properly.

Running remotely using ECL Command Line

The **ECL Command Line Interface (CLI)** application accepts command line parameters to send directly to an ECL execution engine. You can use this utility to control the creation and execution of larger ECL jobs which target a remote system. To compile jobs on a remote system, eclcc is used to create an archive of the ECL code to be compiled, and the ecl CLI is used to submit it to a target cluster for compilation by the remote compiler server (eclccserver).

To submit a job using the ecl CLI, make sure the HPCC has been started and use the following syntax:

```
ecl run  hello.ecl --target=hthor --server=<IP Address of the ESP node>:8010
```

or

```
ecl run hello.ecl --target=hthor --server=.
```

Where "." indicates the IP of the current box.

The workunit[2] result is returned to the command line.

[2]A Workunit is a record of a task submitted to an HPCC. It contains an identifier--workunit ID, the ECL code, results, and other information about the job.

View the full details of the workunit using the ECL Watch interface for your HPCC at this location http://nnn.nnn.nnn.nnn:8010, where nnn.nnn.nnn.nnn is the IP of your ESP server node. Either search for the workunit using the workunit ID or select ECL Workunits/Browse and find your workunit in the list provided.

Setting up an **ecl.ini** file makes running a workunit a little easier when you want to use the same settings every time you submit a workunit in this way. See the *HPCC Client Tools* manual for details.

If your ECL is more complex than a single source file, you can use the eclcc compiler locally to create an archive to be sent to the eclccServer:

```
eclcc hello.ecl -E | ecl run - --target=thor --server=<IP Address of the ESP>:8010
```

The target parameter must name a valid target cluster name as listed in your environment's topology section.

Running a basic ECL program from the ECL IDE

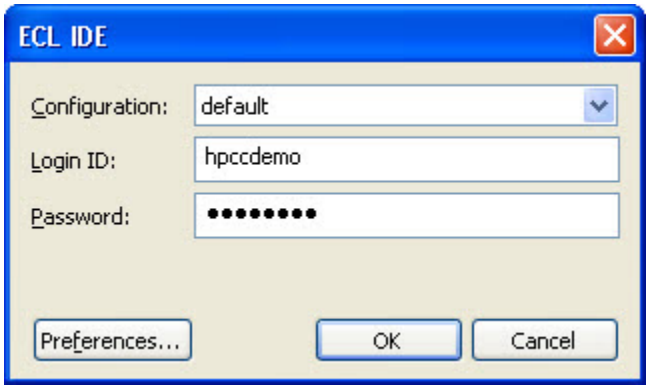
1. Open the ECL IDE on your Windows workstation, from your start menu. (Start >> All Programs >> HPCCSystems >> ECL IDE).

	You can create a shortcut on your desktop to provide quick access to the ECL IDE.

2. Enter the **Login ID** and **Password** provided in the Login dialog.

Login ID	hpccdemo
Password	hpccdemo

Figure 9. Login Window



3. Open a new **Builder Window** (CTRL+N) and write the following code:

```
OUTPUT('Hello World');
```

This could also be written as:

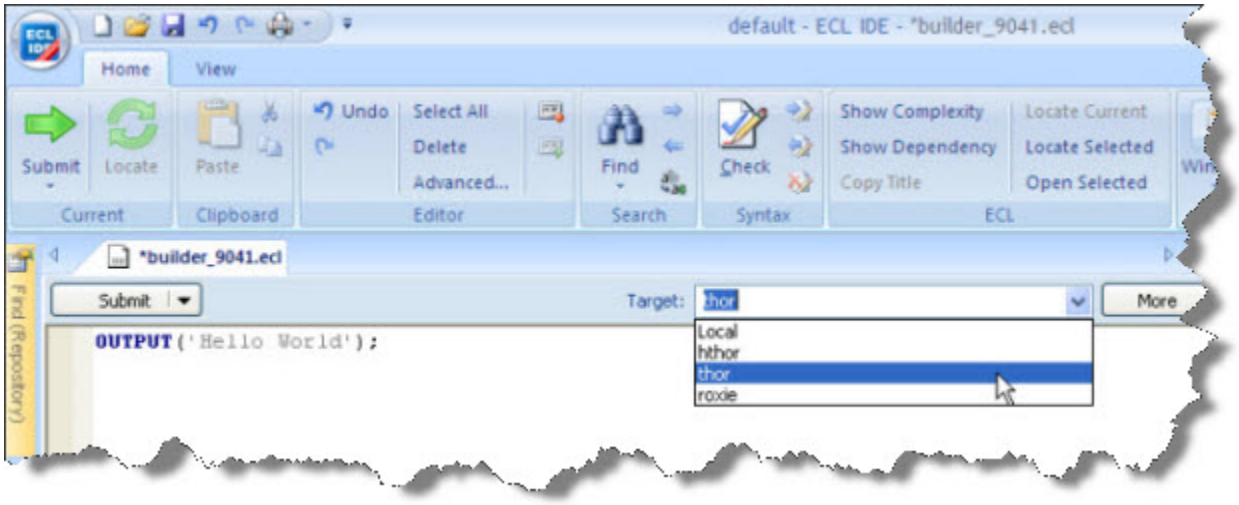
```
'Hello World';
```

In the second program listing, the OUTPUT keyword is omitted. This is possible because the language is declarative and the OUTPUT action is implicit.

4. Select **thor** as your target cluster.

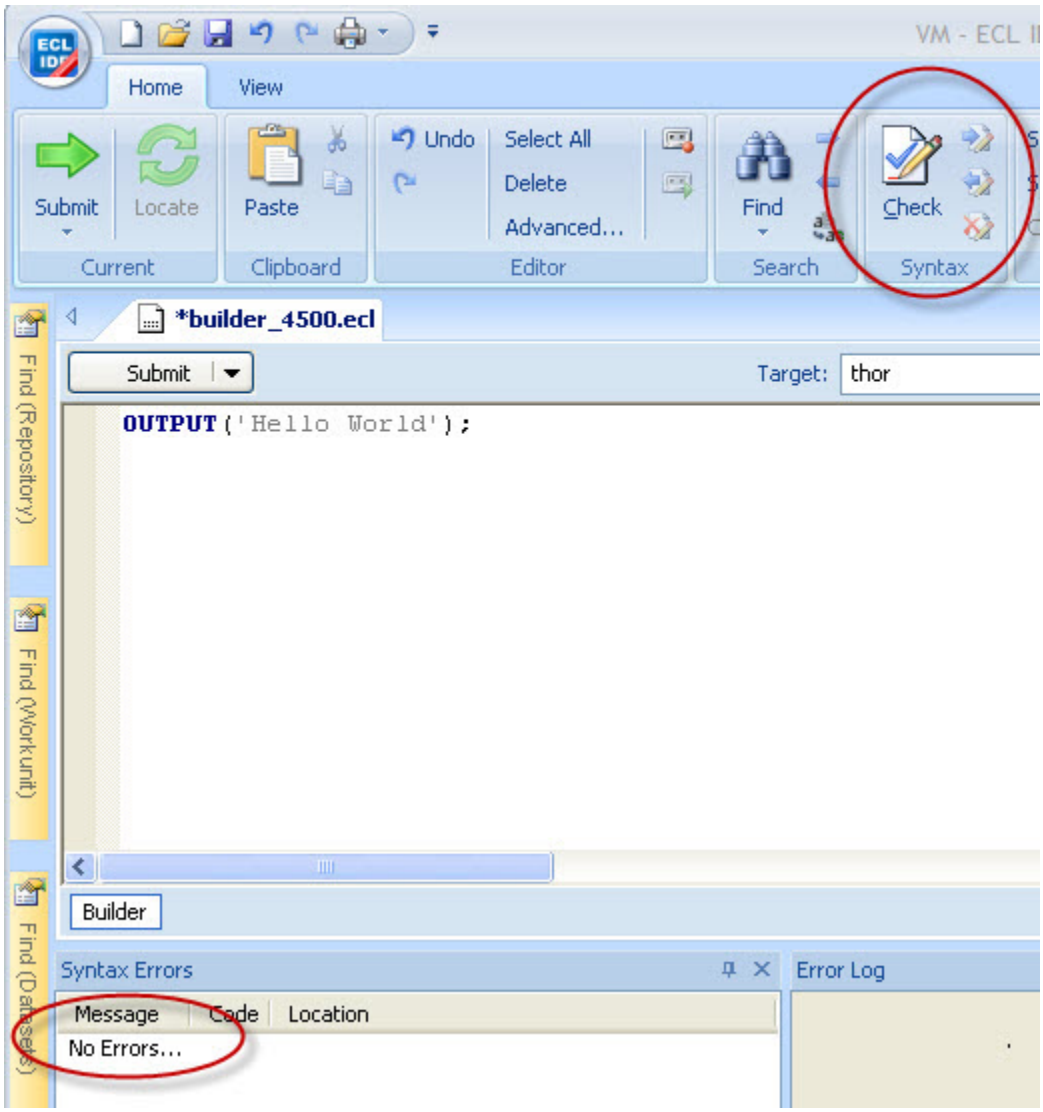
   Thor is the Data Refinery component of your HPCC. It is a disk based massively parallel computer cluster, optimized for sorting, manipulating, and transforming massive data.

Figure 10. Select target



5. Press the syntax check button on the main toolbar (or press F7).

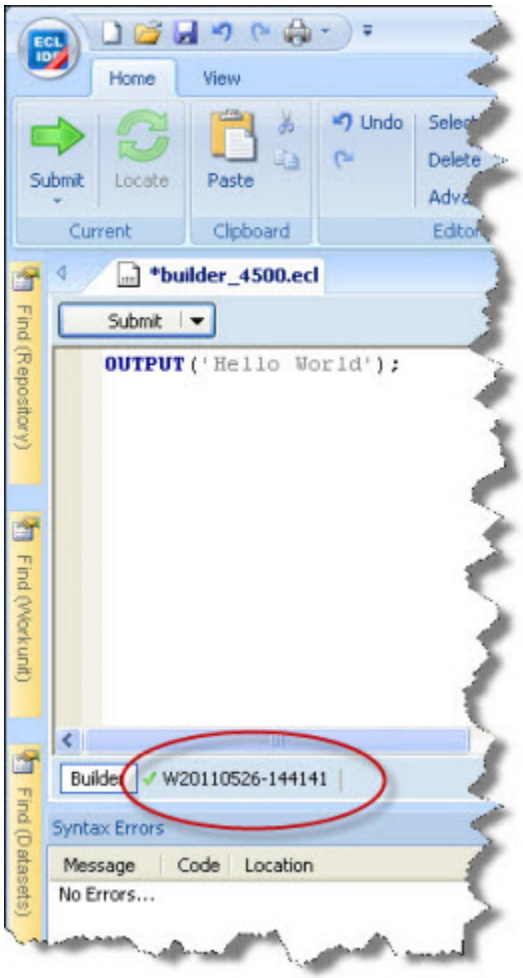
Figure 11. Syntax Check



A successful syntax check displays the "No Errors" message.

6. Press the **Go** button (or press ctrl+enter).

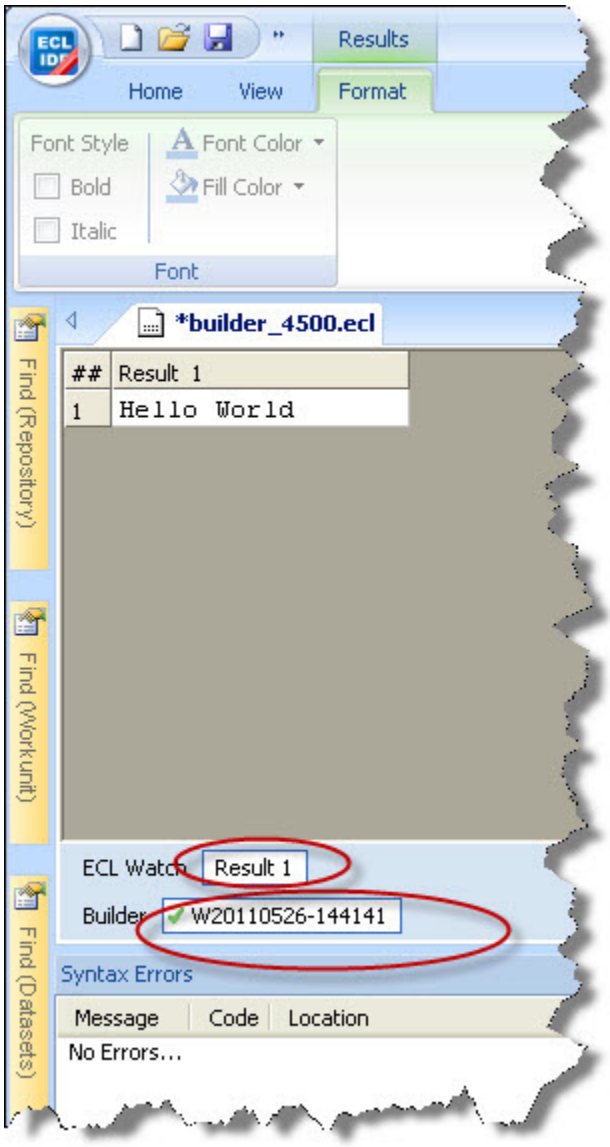
Figure 12. Completed job



The green check mark indicates successful completion.

7. Click on the workunit number tab and then on the Result 1 tab to see the output.

Figure 13. Completed job output



Configuring a Multi-Node System

While the single-node system is fully-functional, it does not take advantage of the true power of an HPCC—the ability to perform operations using Massively Parallel Processing (MPP). This section provides the steps to expand your single-node system into a multi-node system using the Configuration Manager Wizard.

To run a multi-node system, ensure that you have exactly the same packages installed on every node. Follow the steps below to configure your multi-node system to leverage the full power of Massively Parallel Processing.

Using the Configuration Manager Wizard

This section details reconfiguring a system to use multiple nodes. Before you start this section, you must have already downloaded the correct packages for your distro from the HPCC Systems® website: http://hpccsystems.com/download/free-community-edition.

1. If it is running, stop the HPCC system, using this command:

   Centos/Red Hat

   ```
   sudo /sbin/service hpcc-init stop
   ```

   Ubuntu

   ```
   sudo service hpcc-init stop
   ```

   Debian 6 (Squeeze)

   ```
   sudo /etc/init.d/hpcc-init stop
   ```

   You can use this command to confirm HPCC processes are stopped (on Centos/Red Hat):

   ```
   sudo /sbin/service hpcc-init status
   ```

   For Ubuntu

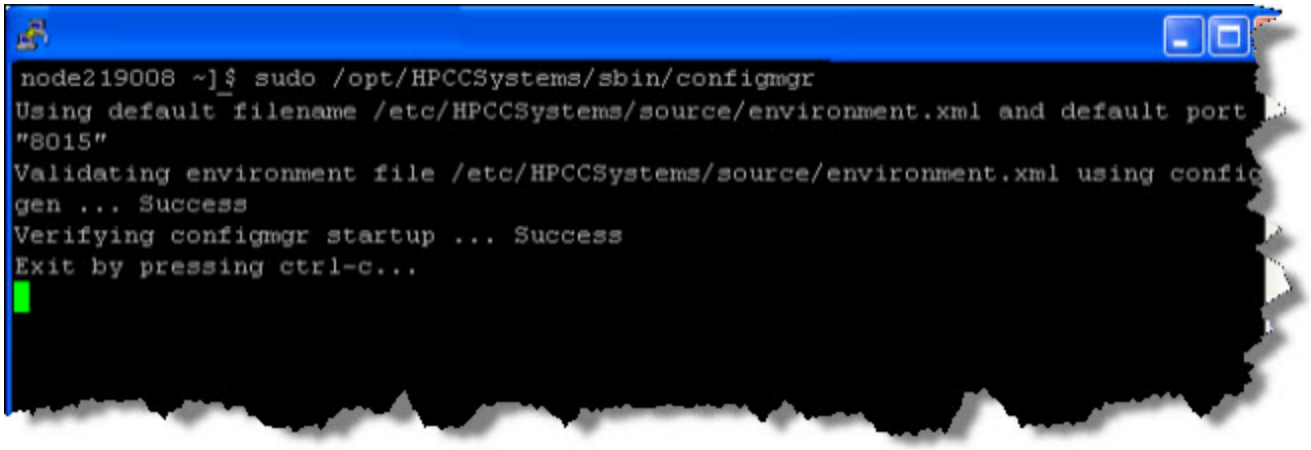
   ```
   sudo service hpcc-init status
   ```

   For Debian 6 (Squeeze)

   ```
   sudo /etc/init.d/hpcc-init status
   ```

2. Start the Configuration Manager service.

   ```
   sudo /opt/HPCCSystems/sbin/configmgr
   ```

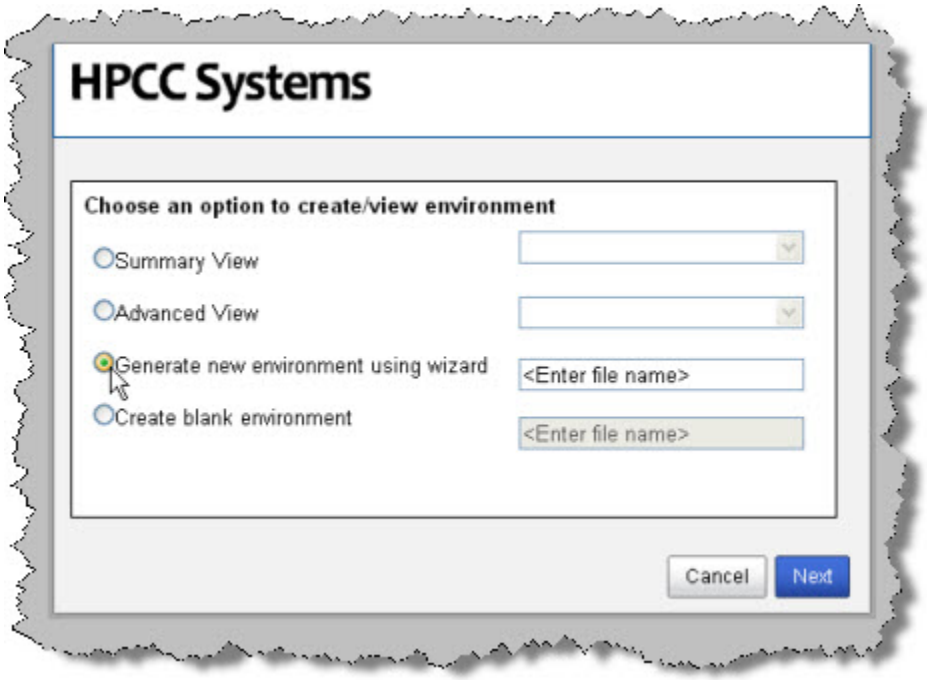


3. Leave this window open. You can minimize it, if desired.

4. Using a Web browser, go to the Configuration Manager's interface:

```
http://<node ip >:8015
```

5. The Configuration Manager startup wizard displays. To use the wizard, select the Generate new environment using wizard button.



6. Provide a name for the environment file.

   This will then be the name of the configuration xml. For example, we will name this *NewEnvironment.xml*.

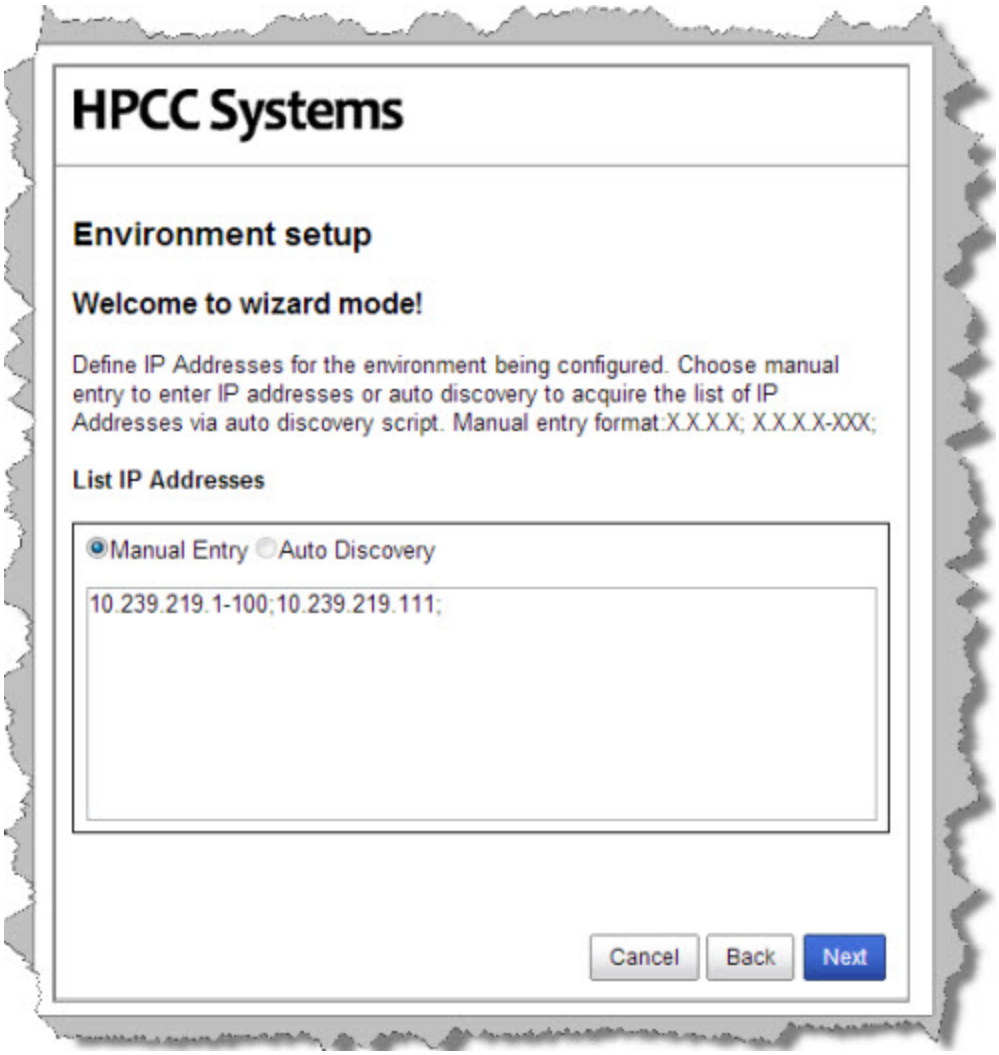
7. Press the **Next** button.

   Next you will need to define the IP addresses that your system will use.

8. Enter the all the IP addresses you want to use in this HPCC.

   The IP addresses do not need to be contiguous. In the image below, we specified the IP addresses nn.nnn.nnn.1-100 and nn.nnn.nnn.111. These are separated with a semi-colon.

   You can specify a range of IPs using a hyphen (for example, NNN.NNN.NNN.1-100). IP Addresses can be specified individually using semi-colon delimiters.

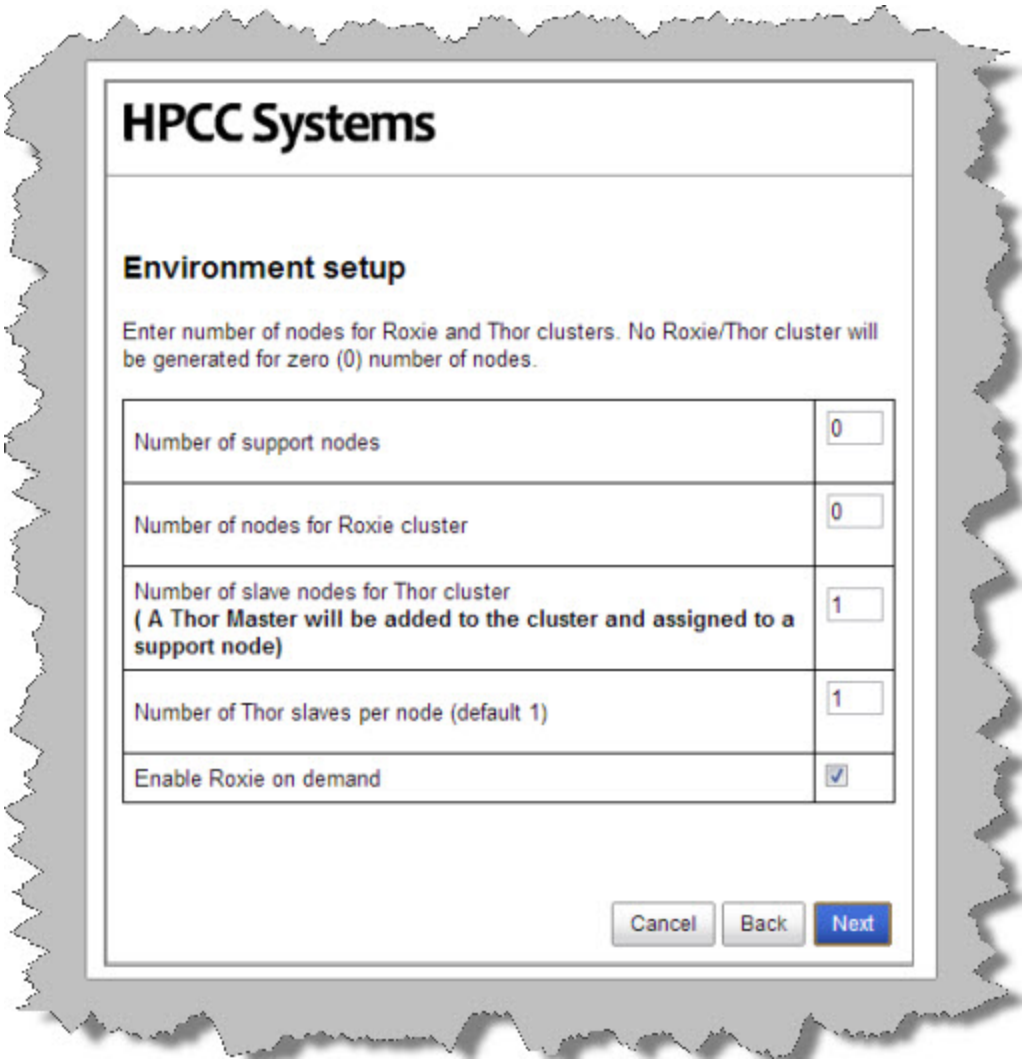


9. Press the **Next** button.

   Alternatively, you could find the IP addresses using Auto Discovery by selecting the Auto Discovery button.

   Now you will define how many nodes to use for the Roxie and Thor clusters.

10.Enter the appropriate values as indicated.



Number of support nodes: Specify the number of nodes to use for support components. The default is 1.

Number of nodes for Roxie cluster: Specify the number of nodes to use for your Roxie cluster. Enter zero (0) if you do not want a Roxie cluster.

Number of slave nodes for Thor cluster Specify the number of slave nodes to use in your Thor cluster. A Thor master node will be added automatically.

Number of Thor slaves per node (default 1) Specify the number of Thor slave processes to instantiate on each slave node. Enter zero (0) if you do not want a Thor cluster.

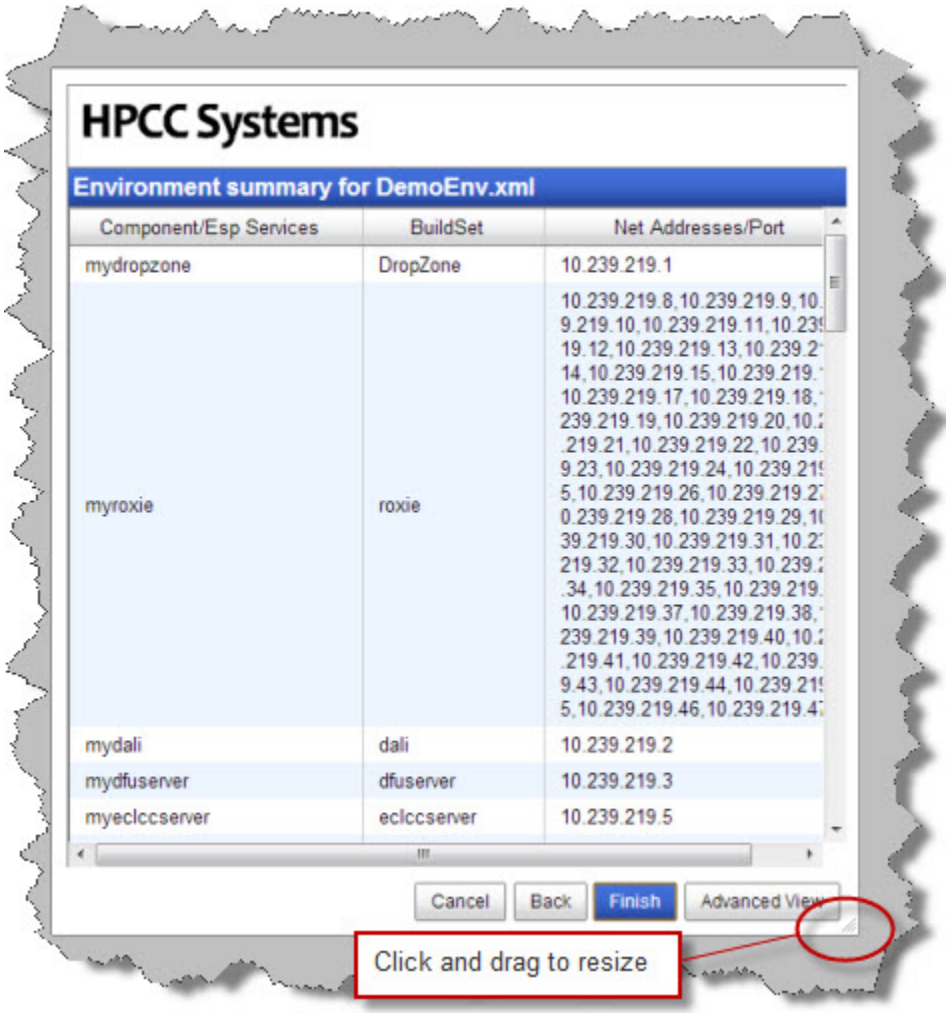
Enable Roxie on demand Specify whether or not to allow queries to be run immediately on Roxie. This must be enabled to run the debugger. (Default is true)

11.Press the Next button

The Environment Summary displays.

12.Click on **Finish** to accept these values. This saves the file.

	Keep in mind, that your HPPC configuration may be different depending on your needs. For example, you may not need a Roxie or you may need several smaller Roxie clusters. In addition, in a production [Thor] system, you would ensure that Thor and Roxie nodes are dedicated and have no other processes running on them. This document is intended to show you how to use the configuration tools. Capacity planning and system design is covered in a training module.



	You can resize the Environment Summary by clicking and dragging the lower right corner.

13.You will now be notified that you have completed the wizard.

At this point the system has created a file named NewEnvironment.xml in the **/etc/HPCCSystems/source** directory

14.Stop the Configuration Manager in the terminal where you started it by pressing CTRL-C.

Be sure system is stopped before attempting to move the environment.xml file.

15.Copy the NewEnvironment.xml file from the source directory to the /etc/HPCCSystems and rename the file to environment.xml

```
# for example
sudo cp /etc/HPCCSystems/source/NewEnvironment.xml /etc/HPCCSystems/environment.xml
```

Make sure that you have sufficient privileges to write file(s) to the destination directory before attempting to copy. If prompted to overwrite the destination file, you should answer **yes**.

16.If you have added new machines to the cluster, you need to copy and install the HPCC package onto all nodes, and generate and clone the SSH keys. This can be done using the install-cluster.sh script which is provided with HPCC. Use the following command:

```
/opt/HPCCSystems/sbin/install-cluster.sh -k <package-file-name>
```

Where <package-file-name> is the name of the package file that you want to install on every node - this will be in the form hpccsystems-platform-*xxxx-n.n.nnnn*.rpm (or .deb) depending on the version and distro. More details including other options that may be used with this command are included in the appendix.

17.Copy the **/etc/HPCCSystems/environment.xml** to **/etc/HPCCSystems/** on **every** node.

You may want to create a script to push out the XML file to all nodes. A sample script is provided with HPCC. The following command copies the XML files out to all nodes as required:

```
sudo /opt/HPCCSystems/sbin/hpcc-push.sh <sourcefile> <destinationfile>
```

See the appendix for more information on using this script.

18.Restart the HPCC system on **every** node. The following command starts the HPCC system on an individual node:

Centos/Red Hat

```
sudo /sbin/service hpcc-init start
```

Ubuntu

```
sudo service hpcc-init start
```

Debian 6 (Squeeze)

```
sudo /etc/init.d/hpcc-init start
```

You may want to create a script to push this command out to every node. A sample script is provided with HPCC. Use the following command to start HPCC on all nodes:

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init start
```

This script can also be used to stop HPCC on all nodes and to stop and start individual components on all nodes. See the appendix for more details.

Starting and Stopping

Start, Stop, Restart the System

Once you have your system environment established , the **init** system can be used to start, stop, or restart components.

The following commands can be used:

To start the system:

Centos/Red Hat

```
sudo /sbin/service hpcc-init start
```

Ubuntu

```
sudo service hpcc-init start
```

Debian 6 (Squeeze)

```
sudo /etc/init.d/hpcc-init start
```

To stop the system:

Centos/Red Hat

```
sudo /sbin/service hpcc-init stop
```

Ubuntu

```
sudo service hpcc-init stop
```

Debian 6 (Squeeze)

```
sudo /etc/init.d/hpcc-init stop
```

You can use a script to start or stop multiple nodes in the system. See *Example Scripts* in the Appendix section for samples.

Start or Stop Single Components

To start or stop a single component, you can use the -c flag in the init system as follows.

Centos/Red Hat

```
sudo /sbin/service hpcc-init -c <component name> <command>
```

Ubuntu

```
sudo service hpcc-init -c <component name> <command>
```

Debian 6 (Squeeze)

```
sudo /etc/init.d/hpcc-init -c <component name> <command>
```

To stop dafilesrv (a helper application), you must use this command: sudo /sbin/service dafilesrv stop. See Helper Applications for details.

Start or Stop Configuration Manager

Configure the system as desired using Configuration Manager.

1. If the system is running, stop the HPCC system, using this command on **every** node:

   Centos/Red Hat

   ```
   sudo /sbin/service hpcc-init stop
   ```

   Ubuntu

   ```
   sudo service hpcc-init stop
   ```

   Debian 6 (Squeeze)

   ```
   sudo /etc/init.d/hpcc-init stop
   ```

2. Start the Configuration Manager service on one node (usually the first node is considered the head node and is used for this task, but this is up to you)

   ```
   sudo /opt/HPCCSystems/sbin/configmgr
   ```

3. Using a web browser, go to the Configuration Manager's interface:

   ```
   http://<ip of installed system>:8015
   ```

Configuring HPCC for Authentication

This section details the steps to configure your HPCC platform to use authentication. There are two ways to use authentication with your HPCC system: simple htpasswd authentication or LDAP.

The htpasswd authentication method is basic password authentication. It only grants or denies access to a user, based upon MD5 encrypted password authentication.

LDAP authentication offers more features and options. LDAP can not only authenticate users, but adds granularity to the authentication. LDAP allows you to control grouped access to features, functions, and files.

You should consider your system needs and decide which of these methods is appropriate for your environment.

When implementing any form of authentication, we strongly recommend that you enable your ESP server to use HTTPS (SSL) and set ALL service bindings to only use HTTPS. This ensures that credentials are passed over the network using SSL encryption. See *Configuring ESP Server to use HTTPS (SSL)* **for details.**

You should not attempt this until you have already deployed, configured, and certified the environment you will use.

Using htpasswd authentication

htpasswd provides basic password authentication to the entire system. This section contains the information to install and implement htpasswd authentication.

Connect to Configuration Manager

In order to change the configuration for HPCC components, connect to the Configuration Manager.

1. Stop all HPCC Components, if they are running.
2. Verify that they are stopped. You can use a single command, such as :

   ```
   sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init status
   ```

3. Start Configuration Manager.

   ```
   sudo /opt/HPCCSystems/sbin/configmgr
   ```

4. Connect your web browser to the Configuration Manager web interface.

   (using the url of http://<*configmgr_IP_Address*>:8015, where <*configmgr_IP_Address*> is the IP address of the node running Configuration Manager)

5. Select the **Advanced View** radio button.
6. Use the drop list to select the XML configuration file.

   Note: Configuration Manager **never** works on the active configuration file. After you finish editing you will have to copy the environment.xml to the active location and push it out to all nodes.

7. Check the **Write Access** box.

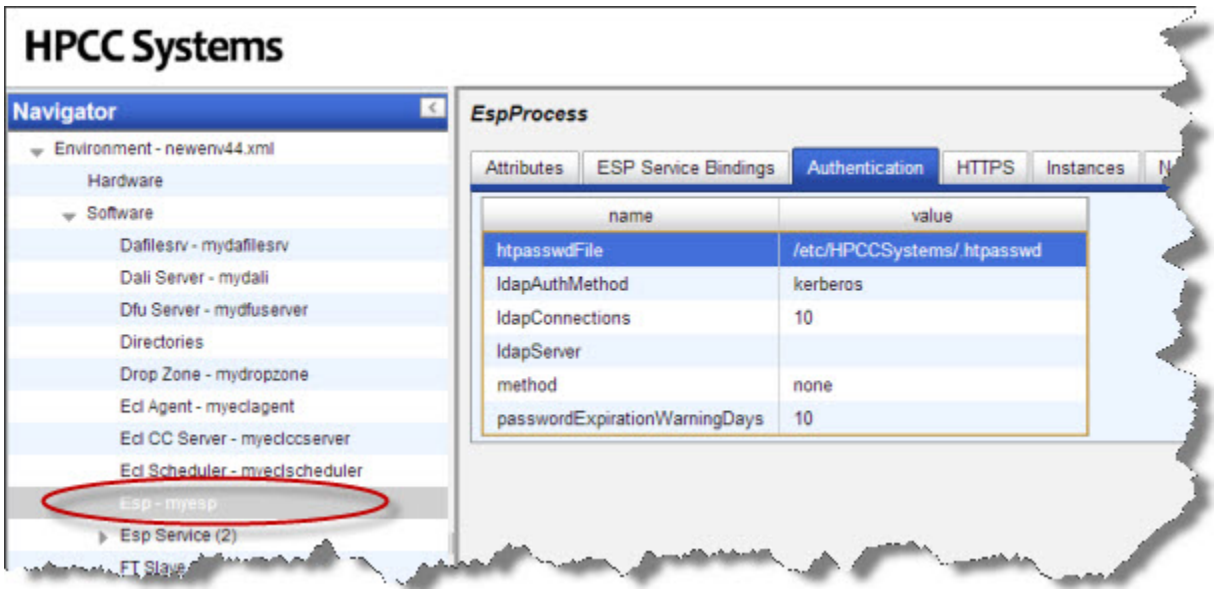
   Default access is read-only. Many options are only available when write-access is enabled.

Enabling htpasswd authentication in HPCC

8. Select **Esp - myesp** in the Navigator panel on the left hand side.

   Note: If you have more than one ESP Server, you would only use one of them for authentication.

9. Select the **Authentication** tab

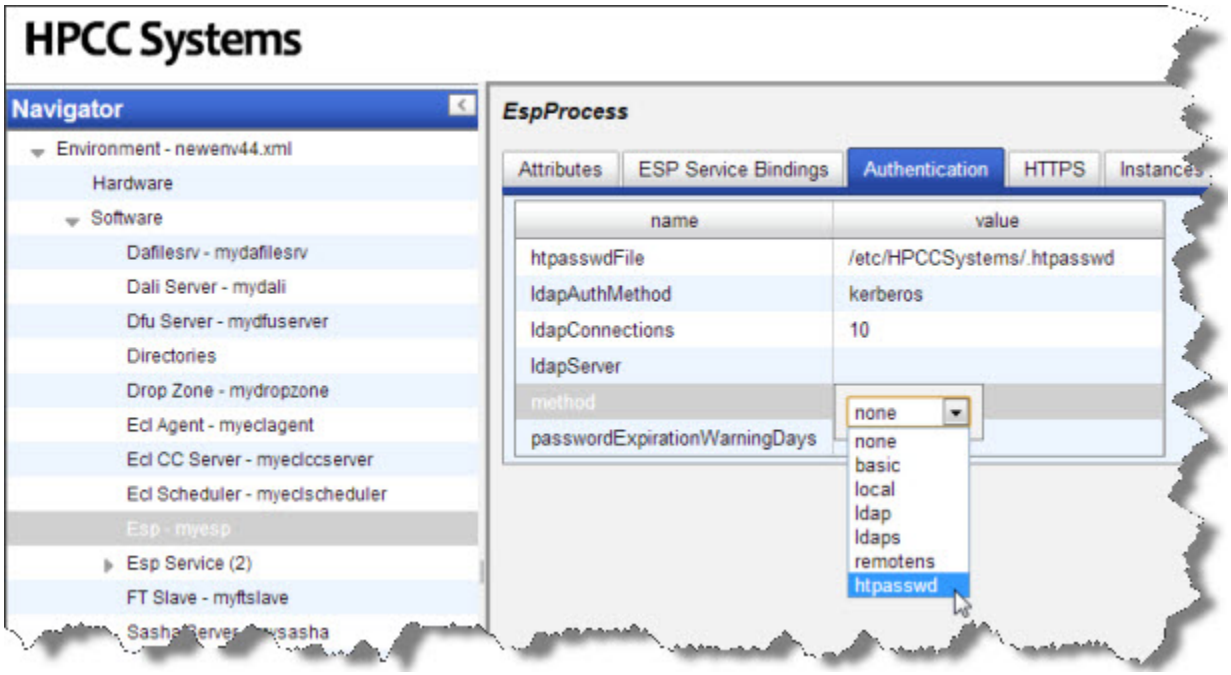


10. Select the **htpasswd File** entry, set the value option to the location of the htpasswd file.

    If the file does not already exist you must create one, see the following section *User administration with htpasswd*.

11. Select the **method** entry.

12. Click on the value column drop list to display the choices for method.



13.Choose **htpasswd** from the drop list.

14.Click on the disk icon to save.

User administration with htpasswd

Users and passwords are kept in the htpasswd file. The htpasswd file needs to exist on the ESP Node that you have enabled authentication. HPCC only recognizes MD5 encrypted passwords.

The default location is: **/etc/HPCCSystems/.htpasswd** on the ESP node that has been configured to authenticate, but it is configurable.

You can use the htpasswd utility to create the .htpasswd file to administer users.

You may already have the htpasswd utility on your system, as it is a part of some Linux distributions. Check your Linux distribution to see if you already have it. If you do not have it you should download the utility for your distribution from The Apache Software Foundation.

For more information about using htpasswd see: http://httpd.apache.org/docs/2.2/programs/htpasswd.html.

Using LDAP Authentication

This section contains the information to install and implement LDAP based authentication. LDAP Authentication provides the most options for securing your system, or parts of your system. In addition to these configuration settings you must run the **initldap** utility to create the appropriate OUs and the default HPCC Admin user on your LDAP server.

Connect to Configuration Manager

In order to change the configuration for HPCC components, connect to the Configuration Manager.

1. Stop all HPCC Components, if they are running.

2. Verify that they are stopped. You can use a single command, such as :

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init status
```

3. Start Configuration Manager.

```
sudo /opt/HPCCSystems/sbin/configmgr
```

4. Connect to the Configuration Manager web interface.

   (using the url of http://<*configmgr_IP_Address*>:8015, where <*configmgr_IP_Address*> is the IP address of the node running Configuration Manager)

5. Select the **Advanced View** radio button.

6. Use the drop list to select the XML configuration file.

Note: Configuration Manager **never** works on the active configuration file. After you finish editing you will have to copy the environment.xml to the active location and push it out to all nodes.

Modifying the configuration

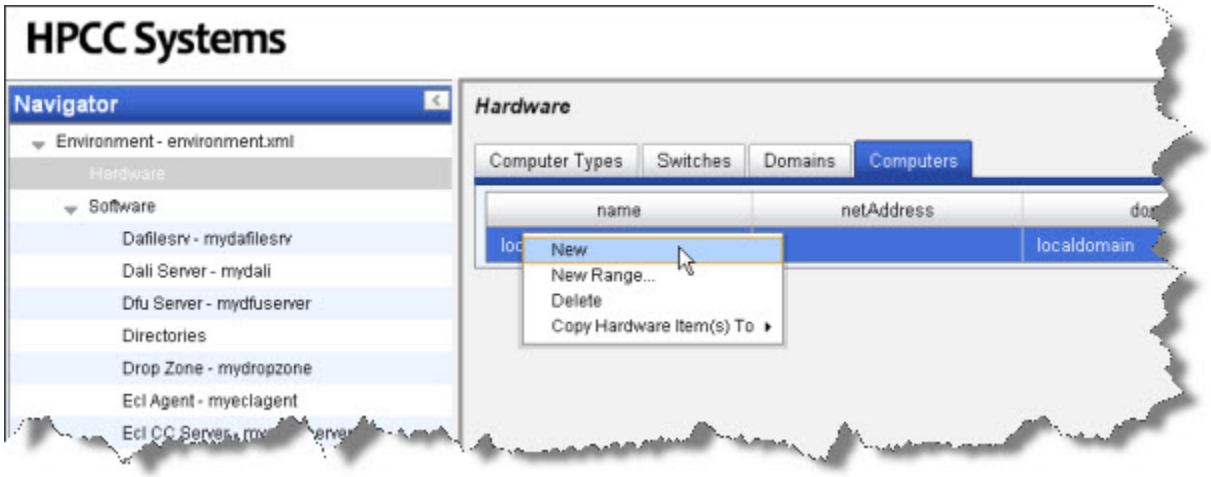
Follow the steps below to modify your configuration.

1. Check the box for **Write Access.**

2. From the **Navigator** pane, select **Hardware**.

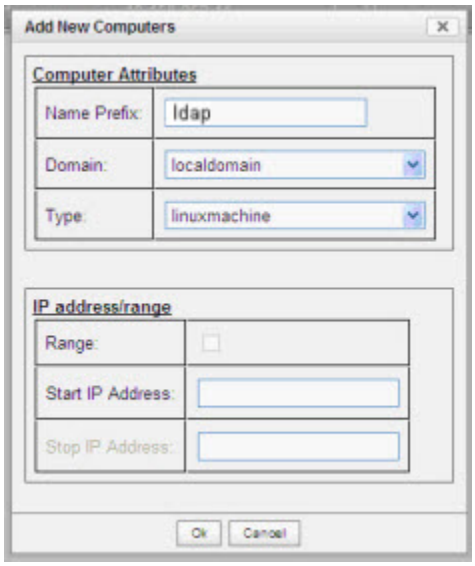
3. Select the **Computers** tab from the panel on the right.

4. Right-click on the table below computers and select **New** from the pop up menu.



The **Add New Computers** dialog displays.

5. Fill in the values for the **Computer Attributes**



a. Provide a **Name Prefix**, for example: ldap.

This helps you to identify it in the list of computers.

b. Fill in **Domain** and **Type** with the values of your domain name, as well as the types of machines you are using.

In the example above, **Domain** is localdomain, and the **Type** is linuxmachine. These should correspond to your domain and type.

If you need to add a new domain or machine type to your system to be able to define an existing LDAP server, you should set these up first in the other two tabs in the hardware section.

c. Add the IP address as appropriate for the LDAP server.

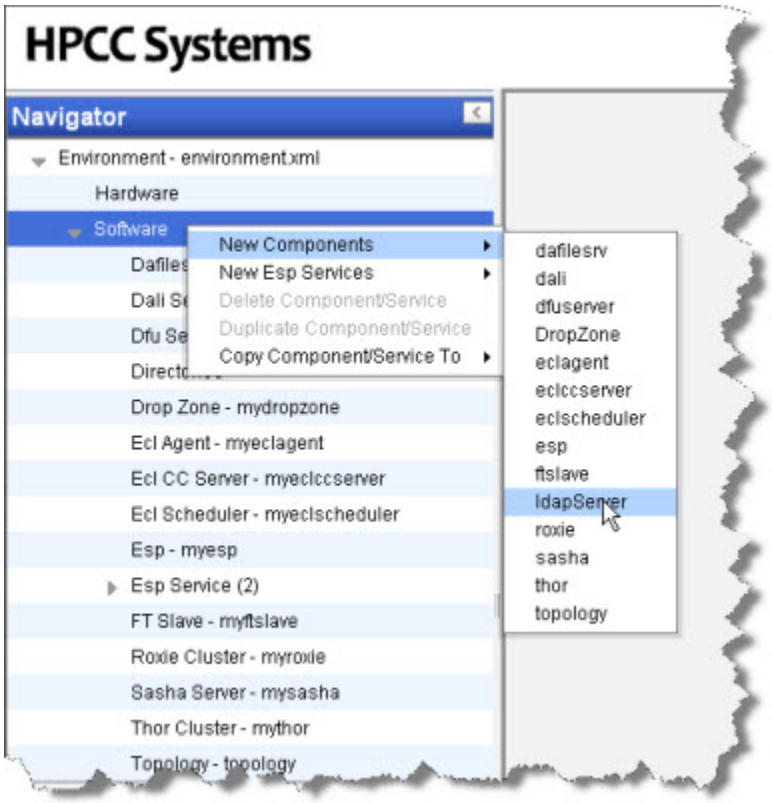
d. Press the **Ok** button.

e. Click on the disk icon to save.

Adding the ldapServer component

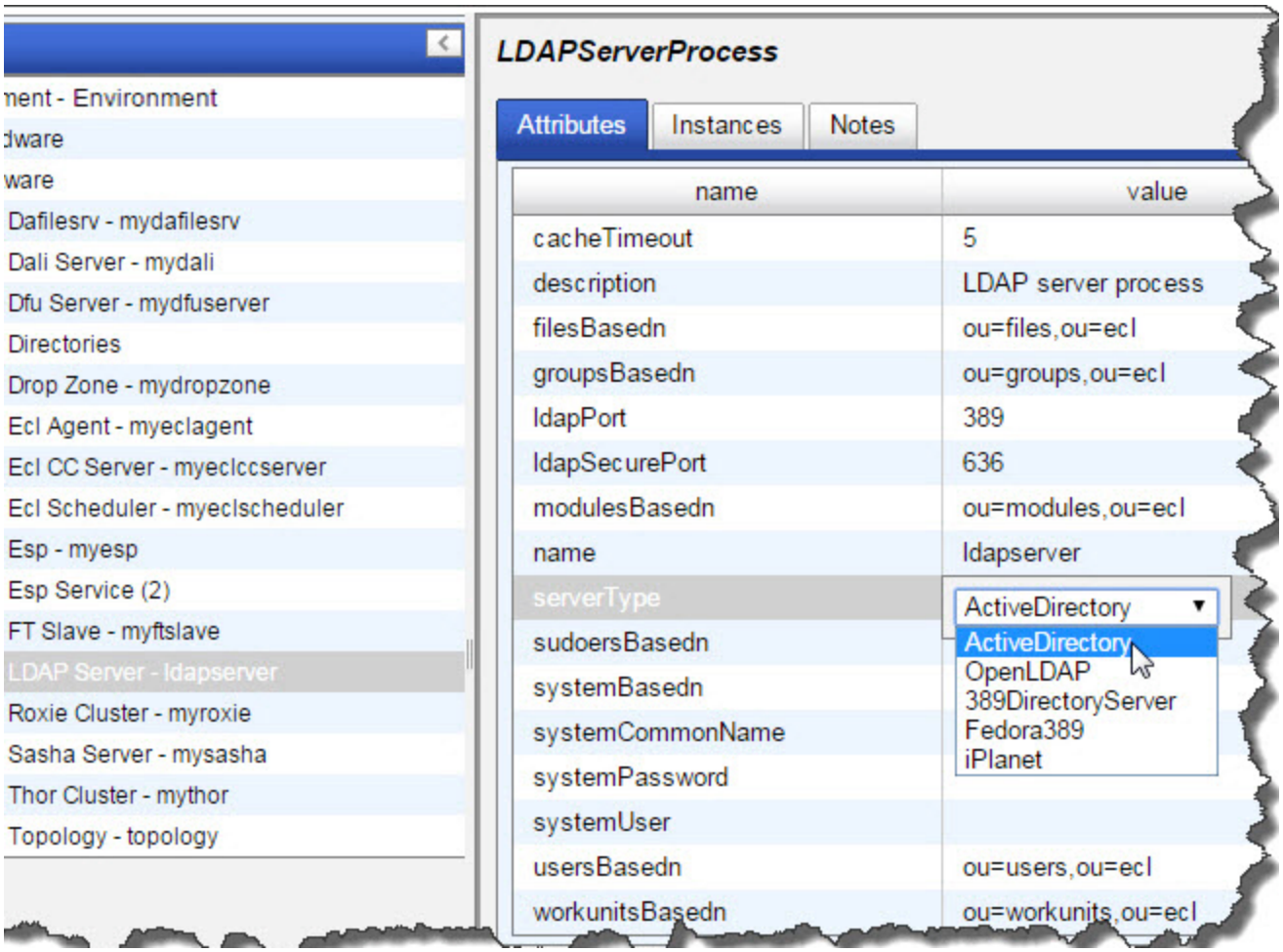
After the LDAP Server node has been added to the Hardware configuration, configure the Software LDAP server definition.

1. Right-click on **Navigator** Pane and choose **New Components** from the pop-up menu, then choose **ldapServer** from the pop-up menu.



	Note: The ldapServer component is merely a definition that specifies an existing LDAP server. It does not install one.

2. Fill in the **LDAP Server Process** properties:



a. On the **Instances** tab, Right-click on the table on the right hand side, choose **Add Instances…**

   The **Select computers** dialog appears.

b. Select the computer to use by checking the box next to it.

   This is the computer you added in the **Hardware** / **Add New Computers** portion earlier.

c. Press the **Ok** button.

d. Fill in the **Attributes** tab with the appropriate settings from your existing LDAP Server.

e. Choose the LDAP server type from the serverType attribute drop box.

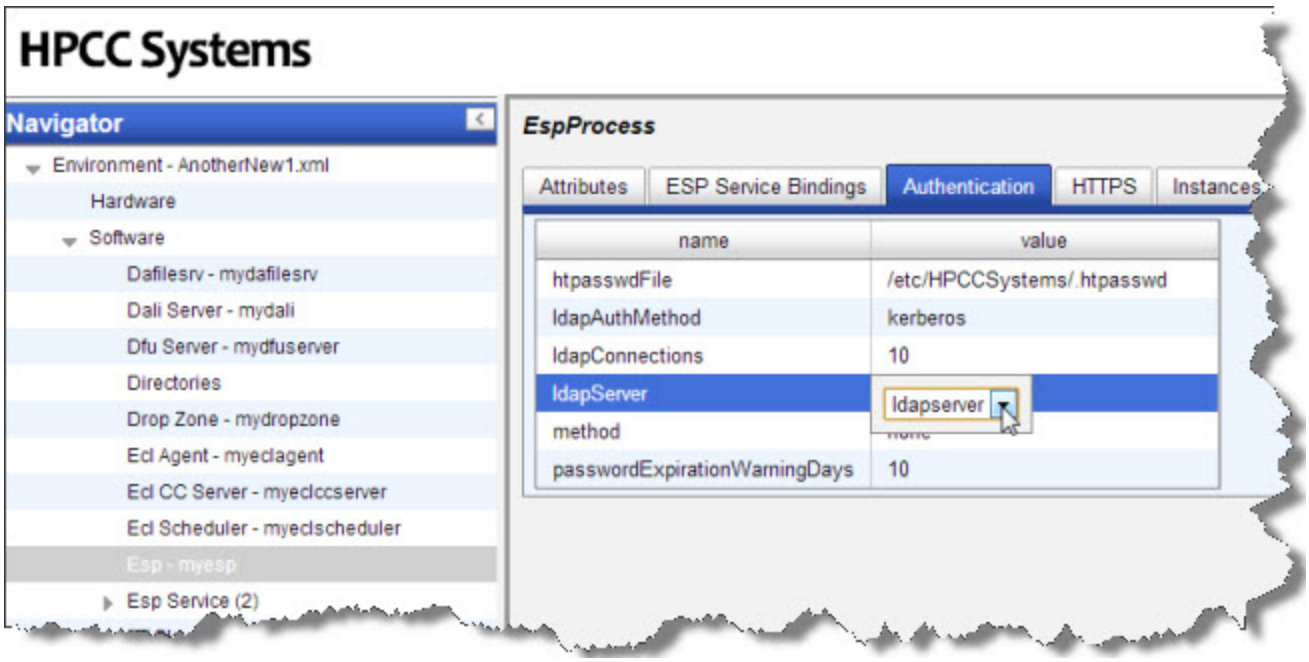
   NOTE: Support for OpenLDAP has been deprecated. The option is included only for legacy purposes.

f. Click on the disk icon to save.

Note: The **cacheTimeout** value is the number of minutes that permissions are cached in ESP. If you change any permissions in LDAP, the new settings will not take effect until ESP and Dali refresh the permissions. This could take as long as the cacheTimeout. Setting this to 0 means no cache, but this has performance overhead so it should not be used in production.

3. In the Navigator pane, click on **ESP – myesp**

4. On the **EspProcess** page on the right hand side, select the **Authentication** tab.



Fill in the appropriate values:

a. Change the **ldapAuthMethod** to simple.

b. Change the **ldapConnections** to the number appropriate for your system (100 is for example only, may not be necessary in your environment).

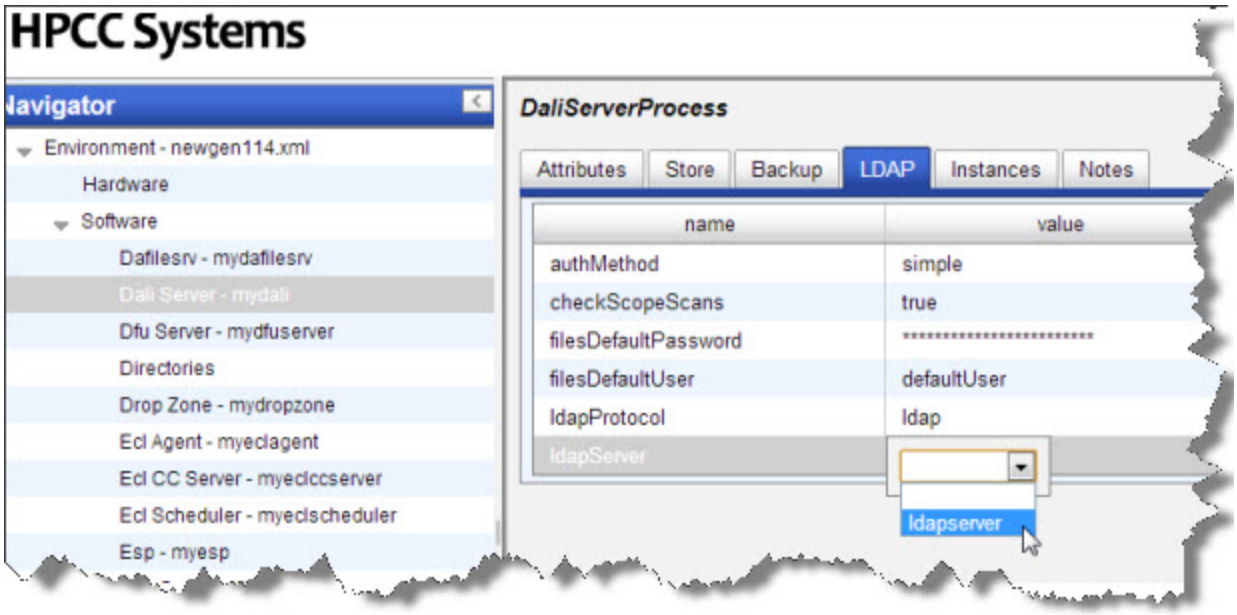
c. Change **ldapServer** value to the name you gave your ldapServer, for example: ldapserver.

d. Change the **method** value to ldap.

e. For the ESP Service bindings, add the **resourcesBasedn** and **workunitsBasedn** to match your LDAP server settings.

f. Click on the disk icon to save.

5. In the Navigator pane, click on the **Dali Server – mydali**



Fill in the values as appropriate:

a. Select the **LDAP** tab.

b. Change the **authMethod** to simple

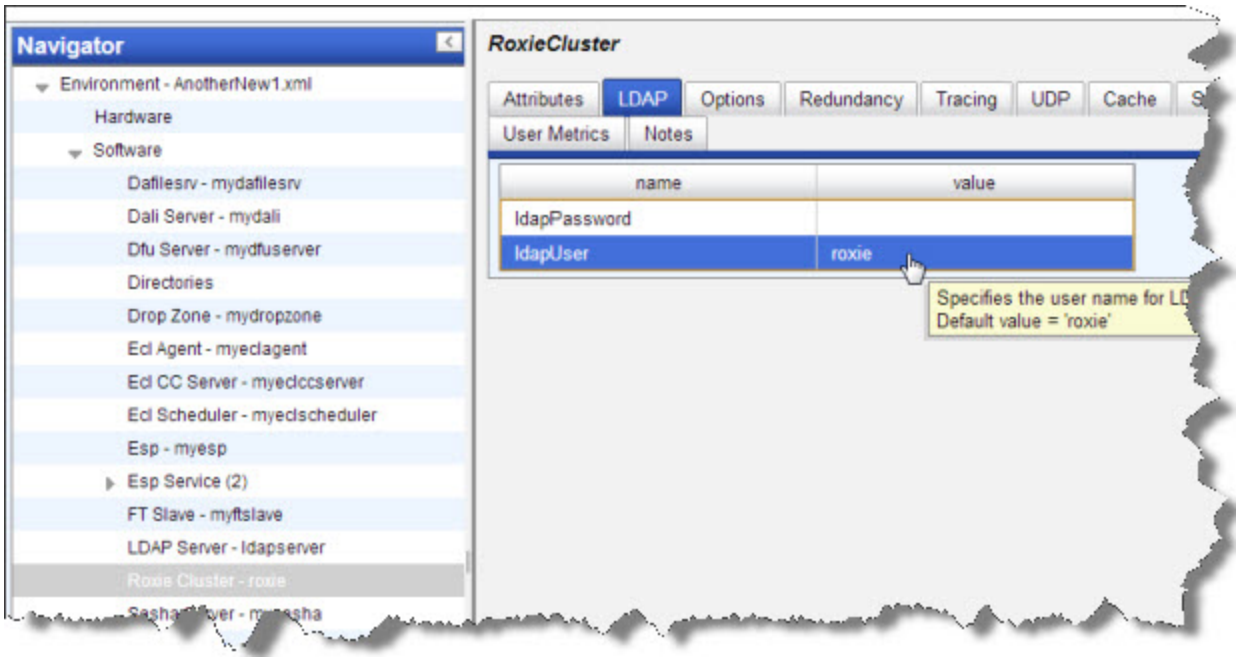
c. Change the LDAP values as appropriate to match the settings in your LDAP server.

   For example: change the **ldapServer** to the value you gave your LDAP Server, in our example it is: *ldapserver*.

   Confirm the change when prompted.

d. Click on the disk icon to save.

6. In the Navigator pane, click on the **Roxie Cluster – myroxie**



a. On the **RoxieCluster** page on the right hand side, select the **LDAP** tab.

b. Locate the **ldapUser** field and verify that there is a "*roxie*" user.

c. You can add password security for Roxie by adding it to the **ldapPassword** field on the same tab.

	In order to run Roxie queries with File Scope security, ensure that the roxie user is created in the list of authenticated users. In the following section, *Adding and editing users*, add "*roxie*" as a user and make sure the password is the same as the one entered in Configuration Manager.

Installing the Default Admin user

After enabling your configuration for LDAP security, you must copy your environment file to the /etc/HPCCSystems directory. See the section *Configuring a Multi-Node System* for more info about configuring your system. With the correct environment.xml file in place, you must then run the **initldap** utility that initializes the security components and the default users.

The initldap Utility

The initldap utility creates the HPCC Administrator's user account and the HPCC OUs for a newly defined LDAP server. The initldap utility extracts these settings from the LDAPServer component(s) in the environment.xml bound to the configured ESPs.

You run the **initldap** utility once you complete your configuration with LDAP components enabled and have distributed your environment.xml file to all nodes.

```
sudo /opt/HPCCSystems/bin/initldap
```

The **initldap** utility prompts you for LDAP Administrator credentials. Enter the appropriate values when prompted.

The following example of initldap for a 389DirectoryServer deployment.

```
Enter the '389DirectoryServer' LDAP Admin User name on '10.123.456.78'...Directory Manager
Enter the LDAP Admin user 'Directory Manager' password...********

Ready to initialize HPCC LDAP Environment, using the following settings
       LDAP Server     : 10.123.456.78
       LDAP Type       : 389DirectoryServer
       HPCC Admin User : HPCCAdmin389
Proceed?  y/n
```

Using the addScopes tool

When a new ESP user account is created, a private “hpccinternal::<user>” file scope is also created granting new users full access to that scope and restricting access to other users. This file scope is used to store temporary HPCC files such as spill files and temp files.

If you are enabling LDAP file scope security and already have user accounts, you should run the addScopes utility program to create the hpccinternal::<user> scope for those existing users.

Users which already have this scope defined are ignored and so it can be used on both new and legacy ESP user accounts safely.

The tool is located in the **/opt/HPCCSystems/bin/** folder and to run it you must pass the location of **daliconf.xml**, for example:

```
/opt/HPCCSystems/bin/addScopes /var/lib/HPCCSystems/mydali/daliconf.xml
```

User Security Maintenance

Configuring an HPCC System to use Active Directory or LDAP-based security allows you to set permissions to control access to Features, File Scopes, and Workunit Scopes.

Introduction

HPCC systems® maintains security in a number of ways. HPCC Systems® can be configured to manage users' security rights by pointing either at Microsoft's Active Directory on a Windows system, or a 389Directory Server on Linux systems.

Using the Permissions interface in ECL Watch, administrators can control access to features in ECL IDE, ECL Watch, ECL Plus, DFU Plus, and the ECL modules within the Attribute Repository. Optionally, you can also implement file and workunit access control by enabling that setting in the Dali server.

Establish permissions by group or by user and define them by association with a particular feature of the HPCC System. Permissions can be defined for each unique combination of group and feature. Permissions are separated into the following categories:

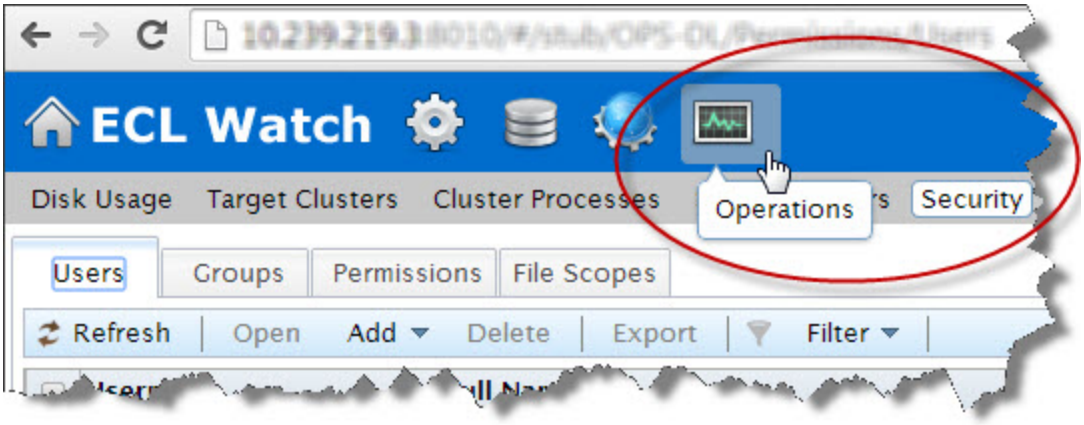
Esp Features for SMC	Controls access to features in ECLWatch and similar features accessed from ECL IDE.
Esp Features for WsEclAccess	Controls access to the WS-ECL web service
Esp Features for EclDirectAccess	Controls access to the ECLDirect web service
File Scopes	Controls access to data files by applying permissions to File scopes
Workunit Scopes	Controls access to Workunits by applying permissions to Workunit scopes
Repository Modules	Controls access to the Attribute Repository and Modules in the repository (legacy)

Security Administration using ECL Watch

Administrator rights are needed to manage permissions. Once you have administrator access rights, open ECL Watch in your browser using the following URL:

- **http://nnn.nnn.nnn.nnn:pppp(where nnn.nnn.nnn.nnn is your ESP Server's IP Address and pppp is the port. The default port is 8010).** For example: http://10.150.51.27:8010/.

Security administration is controlled using the **Security** area of ECL Watch. To access the Security are click on the-**Operations** icon, then click the **Security** link from the navigation sub-menu.



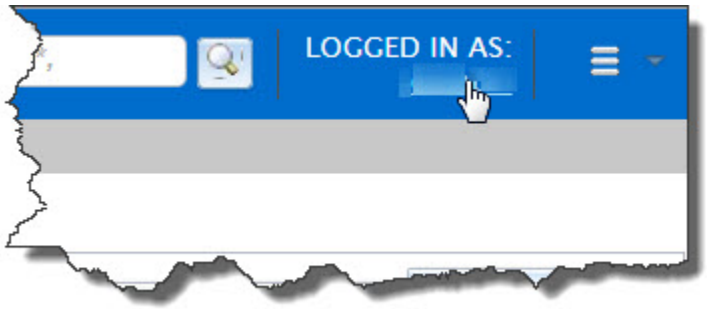
There are three areas where permissions may be set:

- **Users**. Shows all the users currently setup. Use this area to add or delete a user, edit a user's details, set/reset a user's password and view the permissions currently assigned to a user.
- **Groups**. Shows all the groups currently setup. Use this area to add or delete a group, view and edit the members of a group, view and edit the permissions that have been set for a group.
- **Permissions**. Shows the features of the HPCC System where permissions may be set. Use this area to view the permissions currently set for any area of the HPCC System, or to add groups and users and set/modify their permission for a specific feature

	**NOTE:** Use caution when setting any explicit **deny** permission setting. The most restrictive permission always applies.

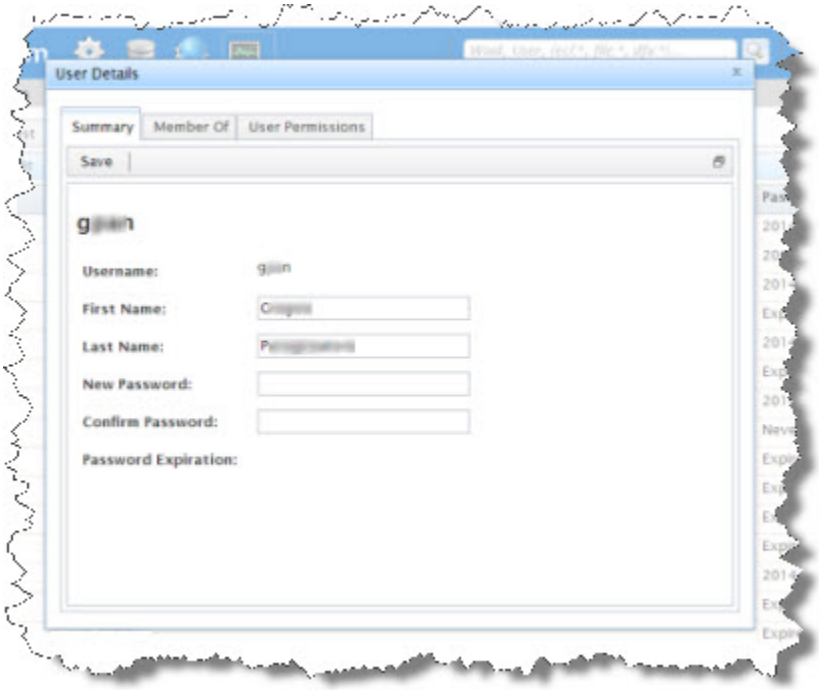
Information about your account

To find out more information about your account, in ECL Watch click on the **Logged In As:** link at the top of the ECL Watch page.



1. Click on the **Logged In As:** link.

   A User Details tab with your account information displays.



2. Confirm the User Name that you are logged in as.

   Note that Administrator rights are needed to manage users and permissions.

   Ensure you are using an account with Administrator rights if you intend to manage users or permissions.

3. Verify the password expiration date, or if password is set to expire.

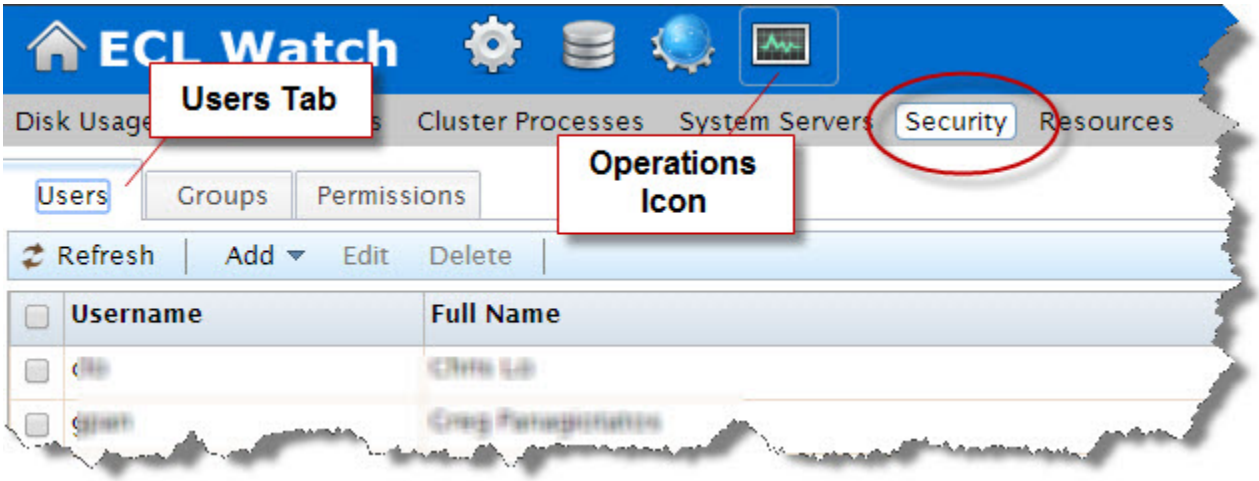
Setting and modifying user permissions

Access to ECL Watch and its features is controlled using a login and password. The **Users** area enables you to control who has access to ECL Watch and the features of your HPCC System to which they have access. Permissions can be set for users based on their individual needs and users can also be added to groups which have already been set up. Use the **Users** menu item to:

- Add a new user (**note**: the username cannot be changed)
- Delete a user
- Add a user to a group
- Change a user's password
- Modify the details/permissions of an individual user

Adding and editing users

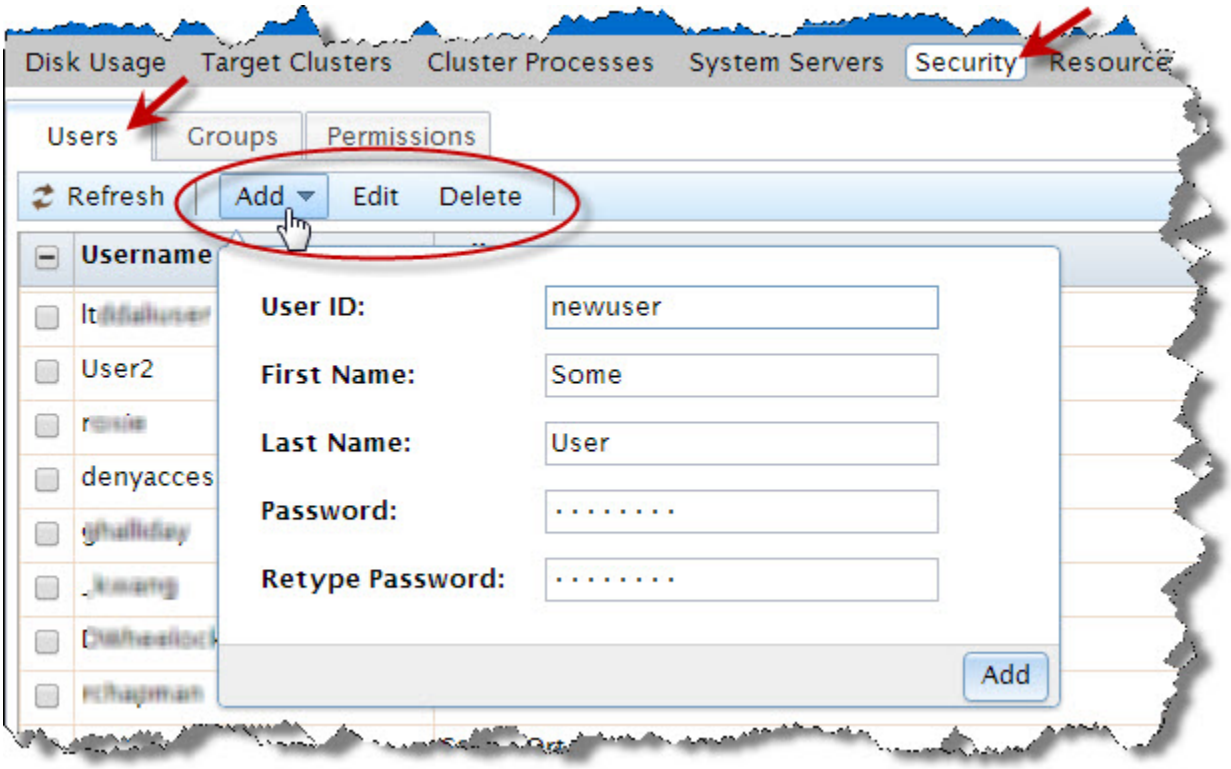
To access the permissions page click on the **Operations** icon, then click the **Security** link from the navigation sub-menu. Click on the **Users** tab to add or edit users.



All current users are identified in the list by their Username and Full Name.

To add a new user to the list of authenticated users:

To add a new user you must have Administrator level access.



1. Press the **Add** button.

   The add user dialog displays.

2. Enter a **Username**.

   This is the login name for using ECL Watch, ECL IDE, WsECL, etc.

3. Enter the **First Name** and **Last Name** of the user.

   This information helps to easily identify the user and is displayed in the **Full Name** field on the main **Users** window.

4. Enter a **Password** for the user and then confirm it in the **Retype Password** field.

5. Press the **Add** button.

   Confirmation of the user request opens a new tab where you can verify the user's information.

6. Press the **Save** button.

Once added, the new user displays in the list and you can modify details and set permissions as required.

To modify a user's details:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

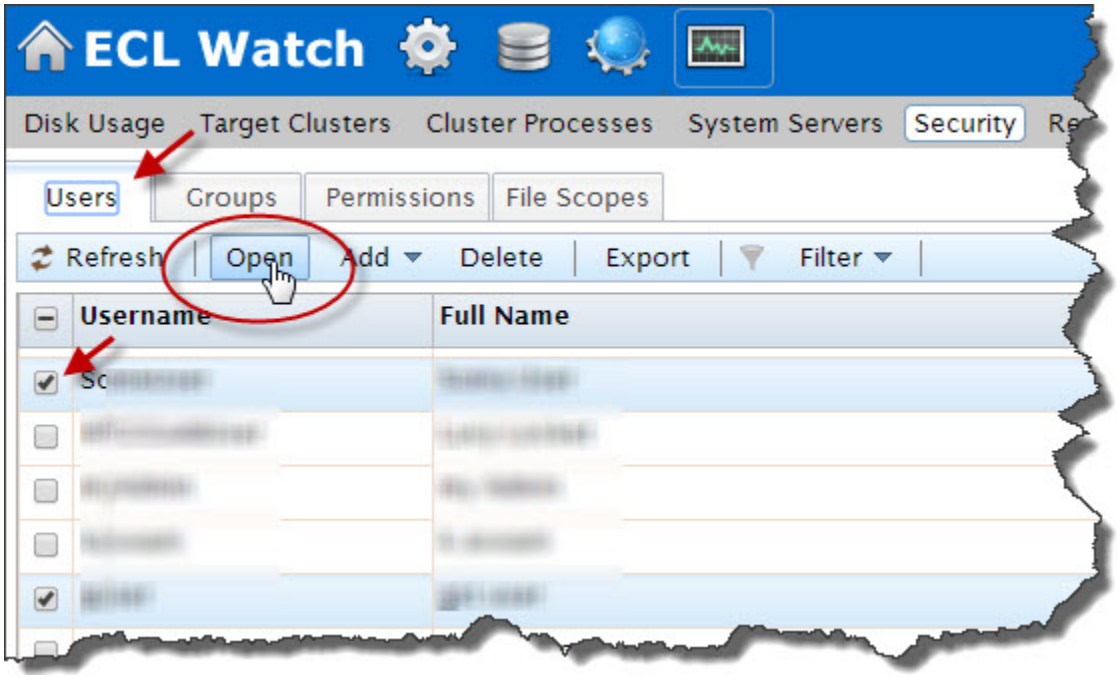
1. Click on the **Users tab**.

   The users display in a list.

2. Select the user (or users) to modify. Check the box next to the Username to select.

   This enables the Users action buttons.

3. Press the **Open** action button.



   A tab opens for each user selected. On that tab there are three sub-tabs.

   The user details are on the **Summary** tab.

4. Modify the user's details as required (if more than one user selected, repeat for each user).

   Note: The **Username** cannot be changed.

5. Press the **Save** button.

   Confirmation message displays.

To add a user to a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

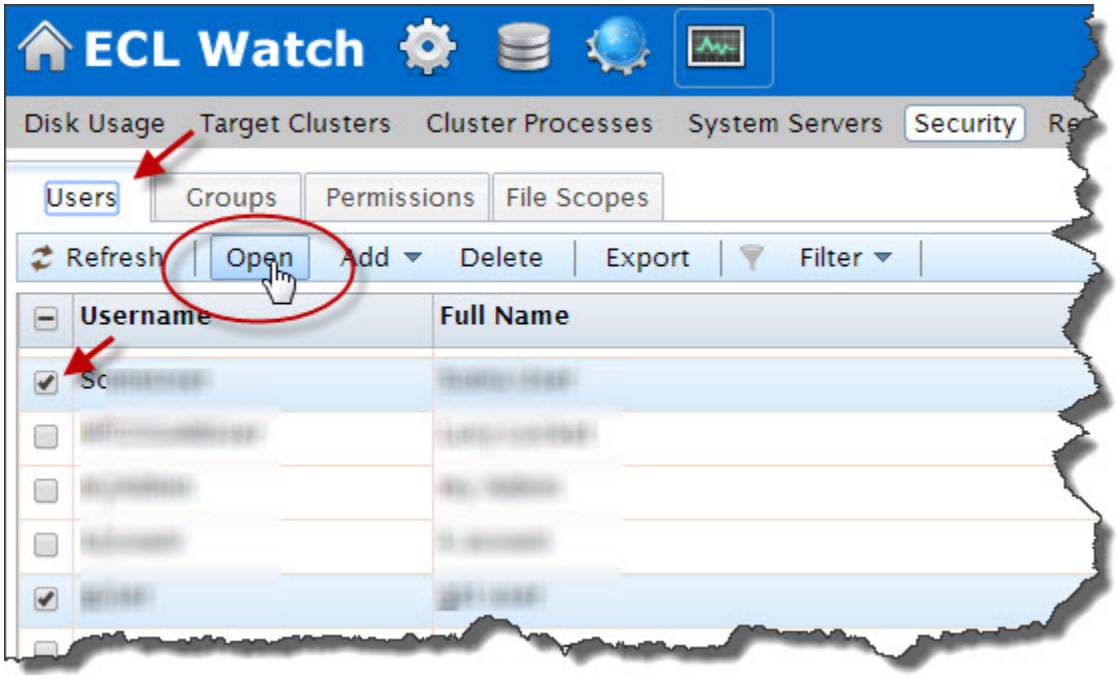
1. Click on the **Users tab**.

   The users display in a list.

2. Select the user (or users) to modify. Check the box next to the Username.

   This enables the user action buttons.

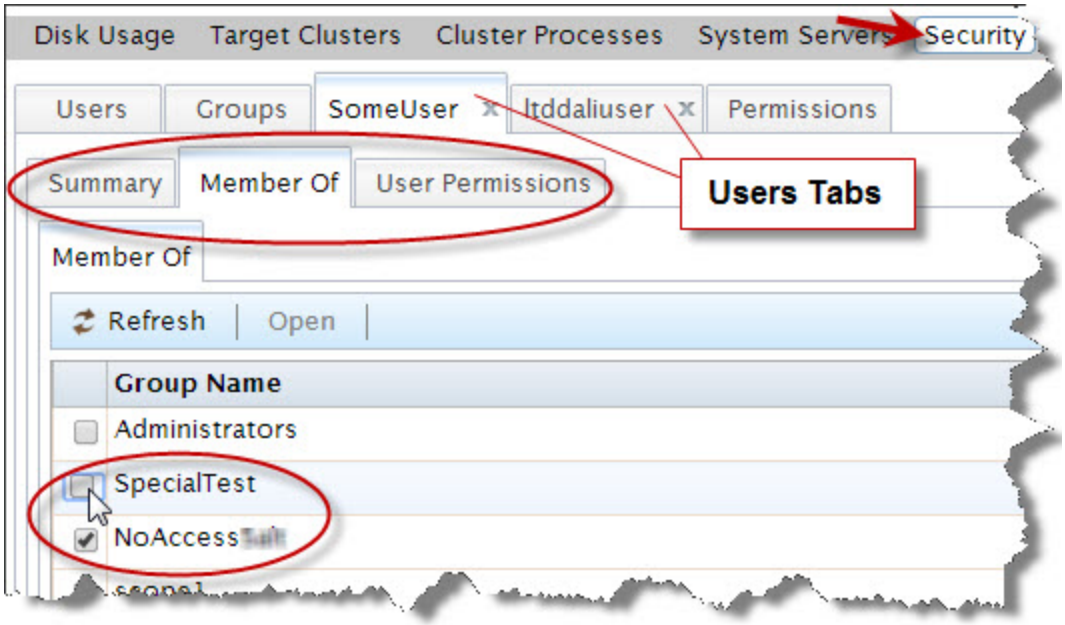
3. Press the **Open** action button.



   A new tab opens for each user selected. On that tab there are three sub-tabs.

4. Click on the tab for the user to modify (if more than one user selected, repeat for each user).

   On the user's tab there are three sub-tabs.



Click on the **Member Of** sub-tab to modify that user's groups.

5. On the **Member Of** tab for that user, a list of the available groups display.

   There is a check in the box next to each group that user belongs to.

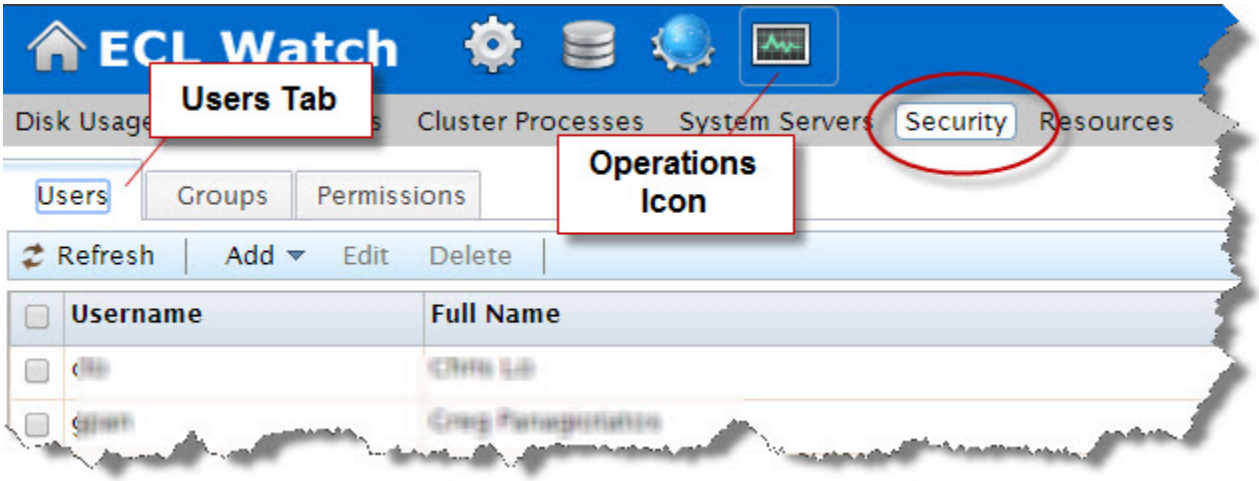
   To add that user to a group, check the box next to the desired group.

6. The changes are automatically saved. Close the tab.

To promote a user to an Administrator

To modify a users credentials you must have Administrator level access. To promote a user to an HPCC Administrator, add the user to the **Administrators** group.

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.



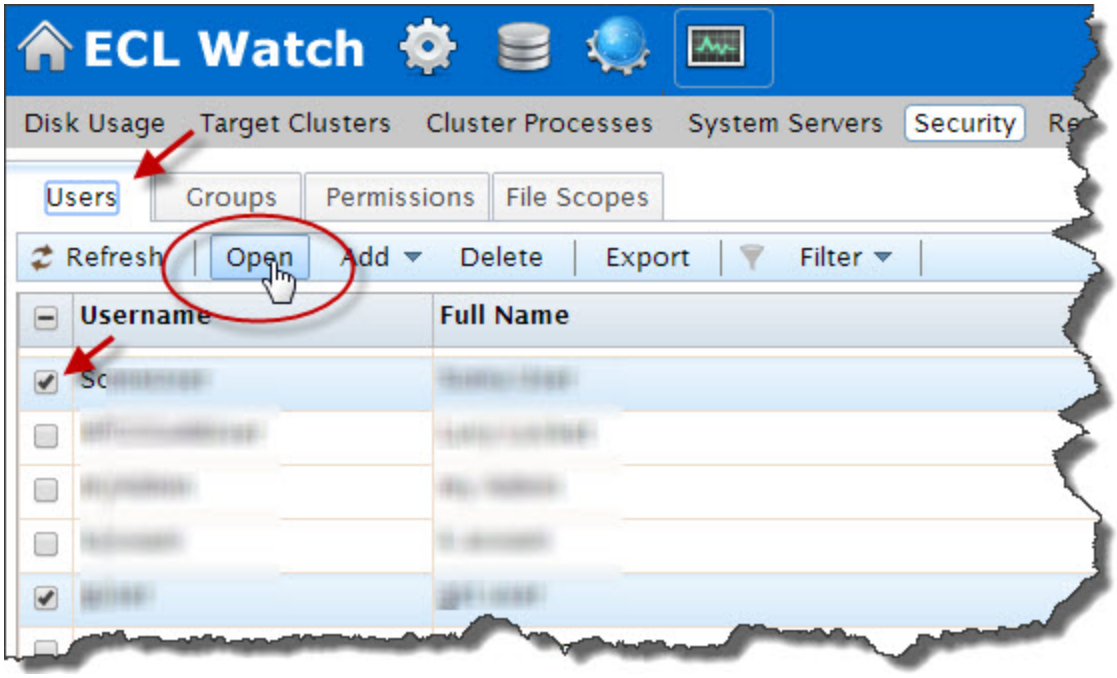
1. Click on the **Users tab**.

The users display in a list.

2. Select the user (or users) to promote. Check the box next to the Username to select.

   This enables the Users action buttons.

3. Press the **Open** action button.

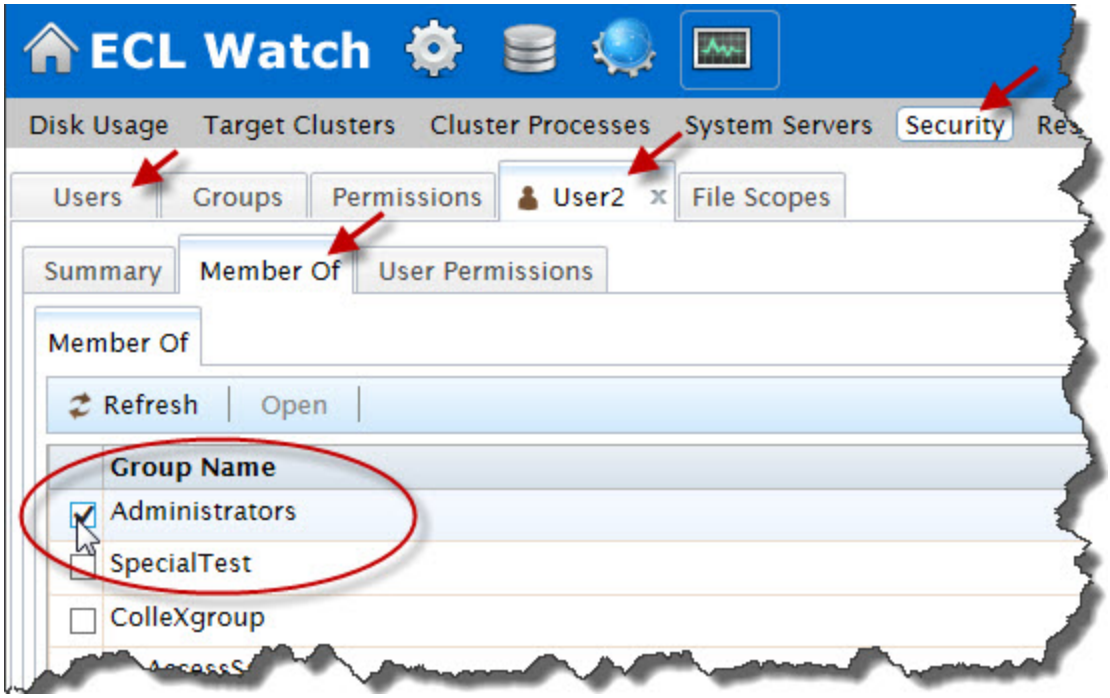


   A tab opens for each user selected. On that tab there are three sub-tabs.

4. Click on the tab for the user to modify (if more than one user selected, repeat for each user).

   On the user's tab there are three sub-tabs.

   Click on the **Member Of** sub-tab.



5. Select **Administrators** by placing a check in box.

   NOTE: The name of the default Administrator group could vary. For example, in Active Directory, it is "Administrators", in LDAP it is "Directory Administrators".

6. The changes are automatically saved. Close the tab(s).

To delete a user from a group:

To delete a user you must have Administrator level access.

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users tab**.

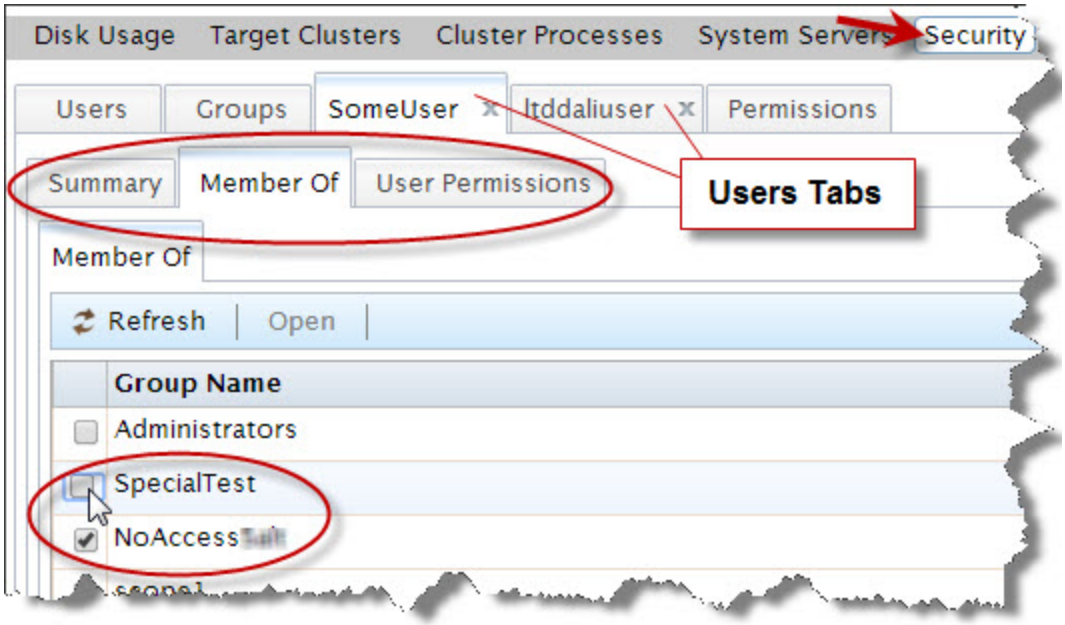
   The users display in a list.

2. Select the user (or users) to modify. Check the box next to the Username.

   This enables the user action buttons. Press the **Edit** action button to modify settings for that user.

3. Click on the tab for the user to modify (if multiple users selected, repeat for each user).

   On the user's tab there are three sub-tabs.



Click on the **Member Of** sub-tab to modify that user's groups.

4. On the **Member Of** tab for that user, there is a list of the available groups.

   There is a check in the box next to each group that user belongs to.

   To remove that user from a group, uncheck the box next to the desired group.

5. The changes are automatically saved. Close the tab.

To change a user's password:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

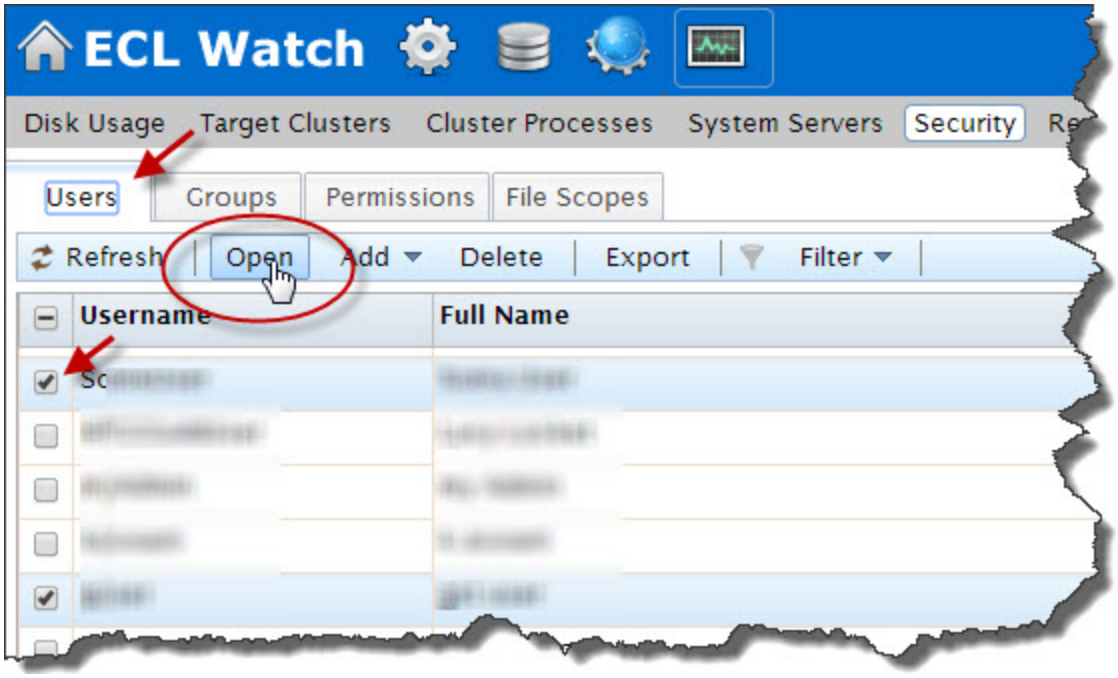
1. Click on the **Users tab**.

   The users display in a list.

2. Select the user (or users) to modify. Check the box next to the Username to select.

   This enables the Users action buttons.

3. Press the **Open** action button.



A tab opens for each user selected. On that tab there are three sub-tabs.

The user details are on the **Summary** tab.

4. Change the password in the **Password** and **Retype New Password** fields as required on the User details summary tab (if multiple users selected, repeat for each user).

   Note: The **Username** cannot be changed.

5. Press the **Save** button.

   A confirmation message displays.

To delete a user from the list of authenticated users:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users** tab.

   The users display in a list.

2. Check the box to the left of the user(s) you want to remove.

   Note: These users will no longer have access to ECL Watch.

3. Press the **Delete** button.

   Confirmation displays.

Setting permissions for an individual user

There may be occasions when you need to modify the permissions for individual users. For example, users may have individual security needs that are not completely covered in any group or, there may be occasions when a user requires

temporary access to an HPCC feature. Permissions set in this area of ECL Watch only affect the user you choose. Most individual permissions you set here overwrite ones set in any group to which the user belongs, except in the case of an explicit deny.

To set permissions for an individual user:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users tab**.

   The users display in a list.

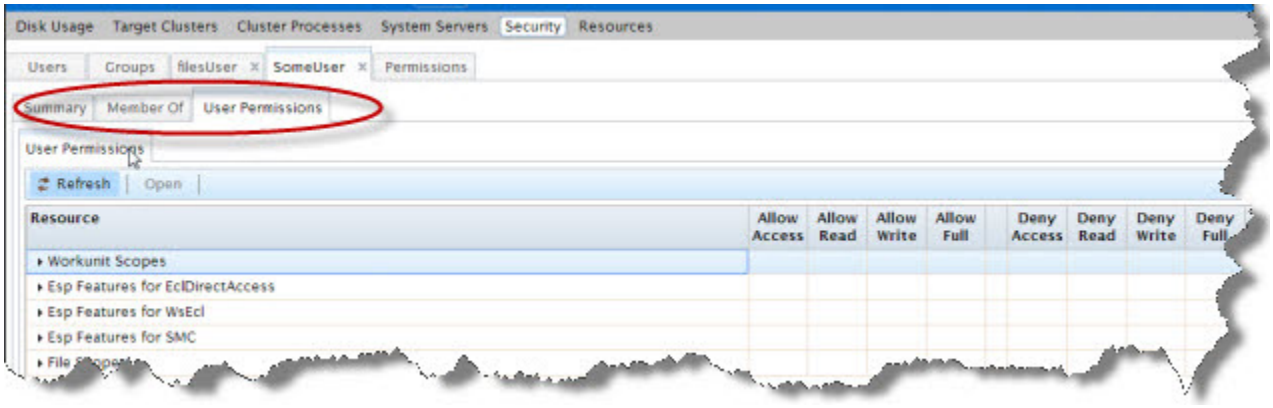
2. Select the user (or users) to modify. Check the box next to the Username to select.

   This enables the Users action buttons.

3. Press the **Open** button.

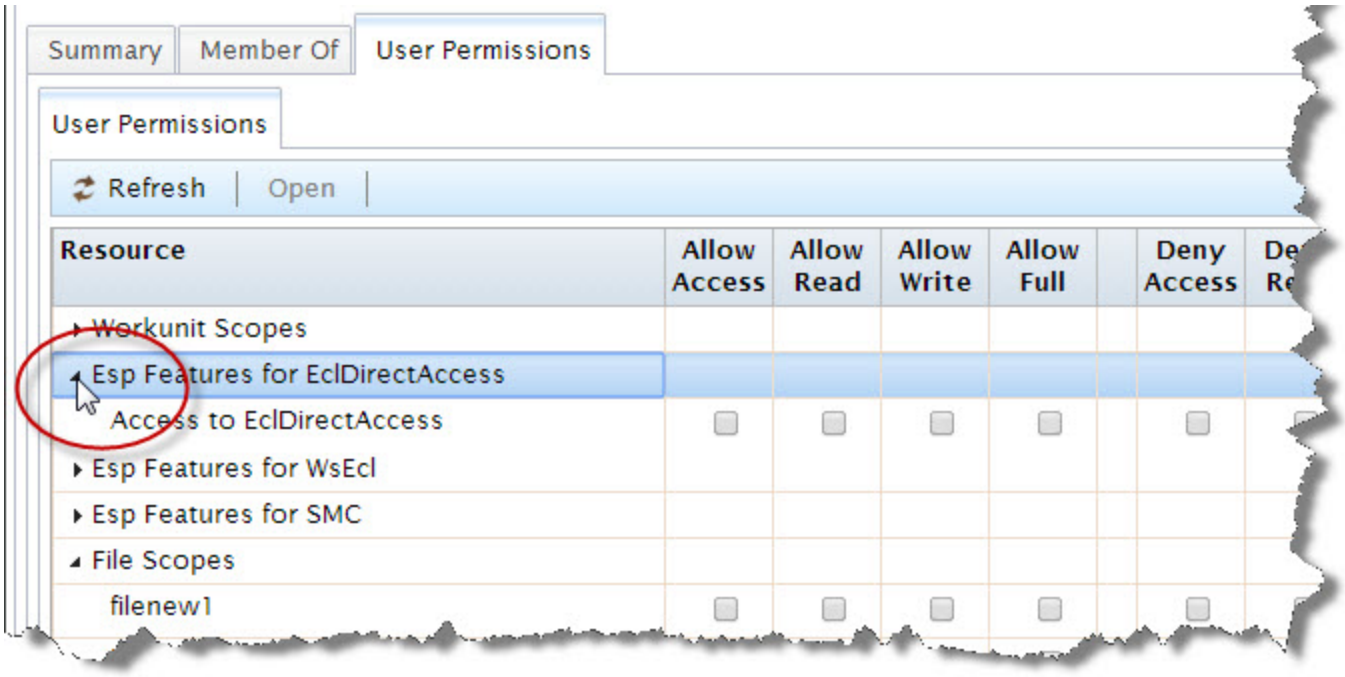
4. Click on the tab for the username to modify (if multiple users selected, repeat for each user).

   On the user's tab there are three sub-tabs.



   Click on the **User Permissions** sub-tab to modify that user's permissions.

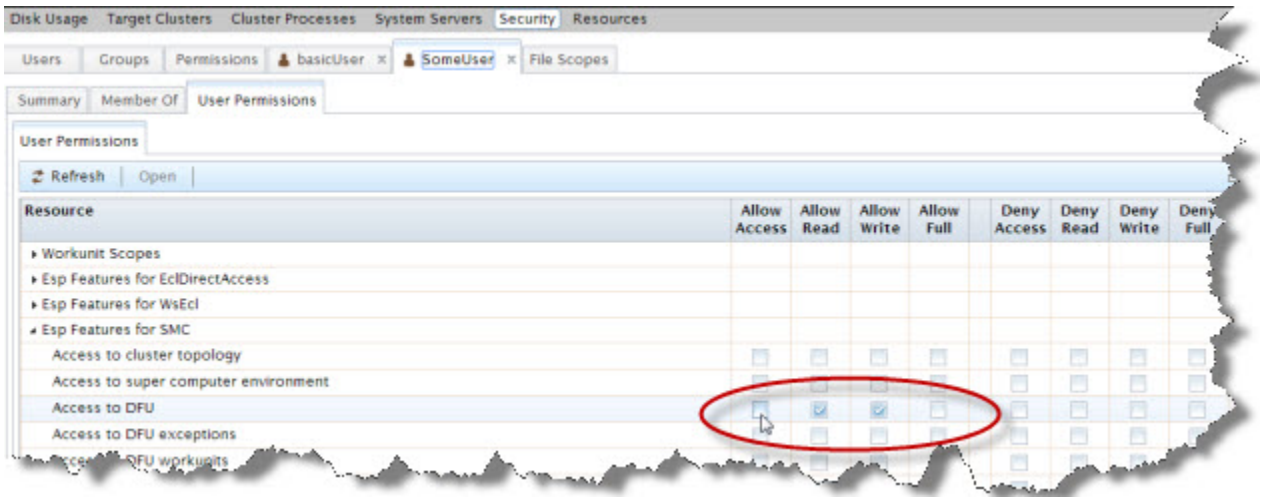
5. Click on the arrow next to the resource to display the permissions for that resource.



The list of permission groups currently set for this user and the ones the user has inherited are also listed. Click the arrow to allow setting the individual resource settings.

6. There may be more than one resource setting available in each group, be sure to set the permissions for each setting as required.

7. Check the boxes that **allow** and **deny** access as required for the user.



	**NOTE:** Use caution when setting any explicit **deny** permission setting. The most restrictive permission always applies.

8. The changes are automatically saved. Close the tab.

Setting and modifying group permissions

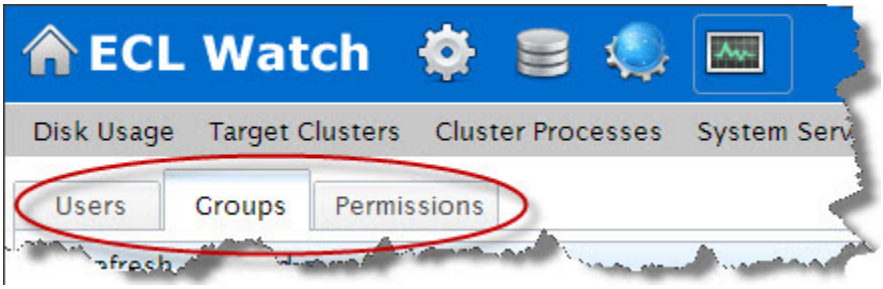
Setting up groups ensures that all users with the same permission needs have the same permission settings. You can give users the access they require to the feature areas of HPCC that they need. There is no limit to the number of groups you can create. You can create as many groups as you need to control access for all your users regardless of their tasks.

Use the **Groups** menu item to:

- Add a new group.
- Delete a group.
- Add members to a group.
- Modify the permissions for a group.

Adding and editing groups

When adding or changing the permissions for a group, all members of that group are given those permission settings. So it is important to be sure that you are giving or denying access to features appropriate for the members of that group. If you need to make a change for a single user (or small number of users), it is probably better to make that change for each individual user as illustrated in the previous sections.

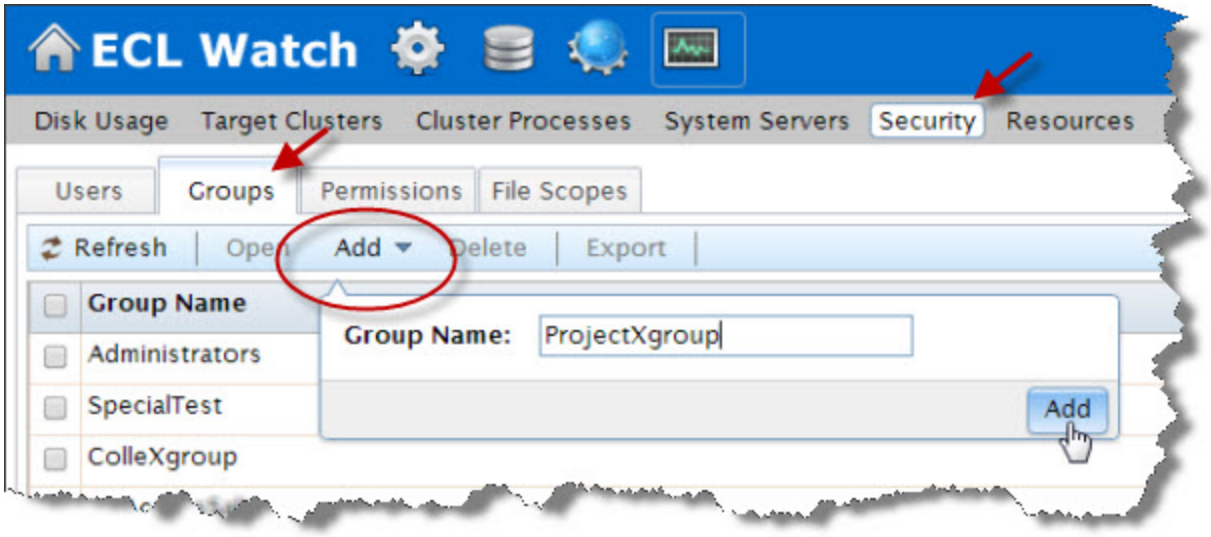


To modify groups, click on the **Operations** icon, then click the **Security** link from the navigation sub-menu. Click on the **Groups** tab.

To add a new group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Press the **Add** action button button.



This opens a dialog where you can enter the name for the group.

3. Enter a **Group Name**.
4. Press the **Add** button.

   This opens a **Summary** tab for this new group.

   You can set the permissions and add members to this group from the respective sub-tabs on that group tab.

To delete a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Locate the group in the list and check the checkbox next to it.
3. Press the **Delete** action button.
4. Press the **OK** confirmation button.

   The group no longer displays in the list.

To add new members to a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Locate the group in the list and check the box next to it.
3. Press **Edit** action button.

   This opens a new tab for the group.

   Three sub-tabs display: **Summary**, **Members**, and **Group Permissions**.

4. Select the **Members** tab.

The members tab displays a list of all users on the system. The users that belong to the selected group have a check in the box next to them.

5. Check the box(es) to the left for all the users you want to add to the group.
6. The changes are automatically saved. Close the tab.

To delete members from a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Locate the group in the list and check the box next to it.
3. Press the **Open** action button.

   This opens a new tab for the group.

   The Groups tab has three sub-tabs: **Summary**, **Members**, and **Group Permissions**.
4. Select the **Members** tab.

   The Members tab displays a list of all users on the system. The users that belong to the selected group have a check in the box next to them.
5. Uncheck the box(es) to the left for all users you want to delete from the group.
6. The changes are automatically saved. Close the tab.

Setting permissions for a group

By default, all users are members of the **Authenticated Users** group. The **Authenticated Users** group has access rights to almost all resources. To set up more restricted controls, you should create specific groups with more restricted permissions.

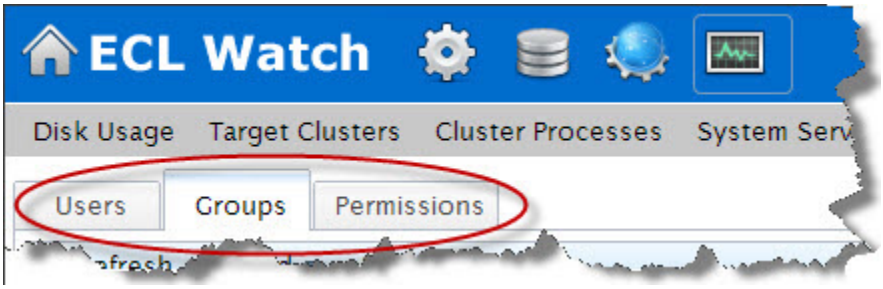
You can then create groups with only those access rights you wish to grant. This approach allows the most flexibility since a single User ID can have multiple group memberships.

As a best practice, you should use **Allow** instead of **Deny** to control access. Denies should be used only as an exception, when possible. If you wish to deny a user access to some specific control, a good practice would be to create a group for that, place the user(s) in that group, then you can deny access to that group.

Remember the most restrictive control takes precedence. For example, if a user is in a group that has deny permission to file access, and the user is in another group where file access is allowed, that user will still not have file access.

To set permissions for a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

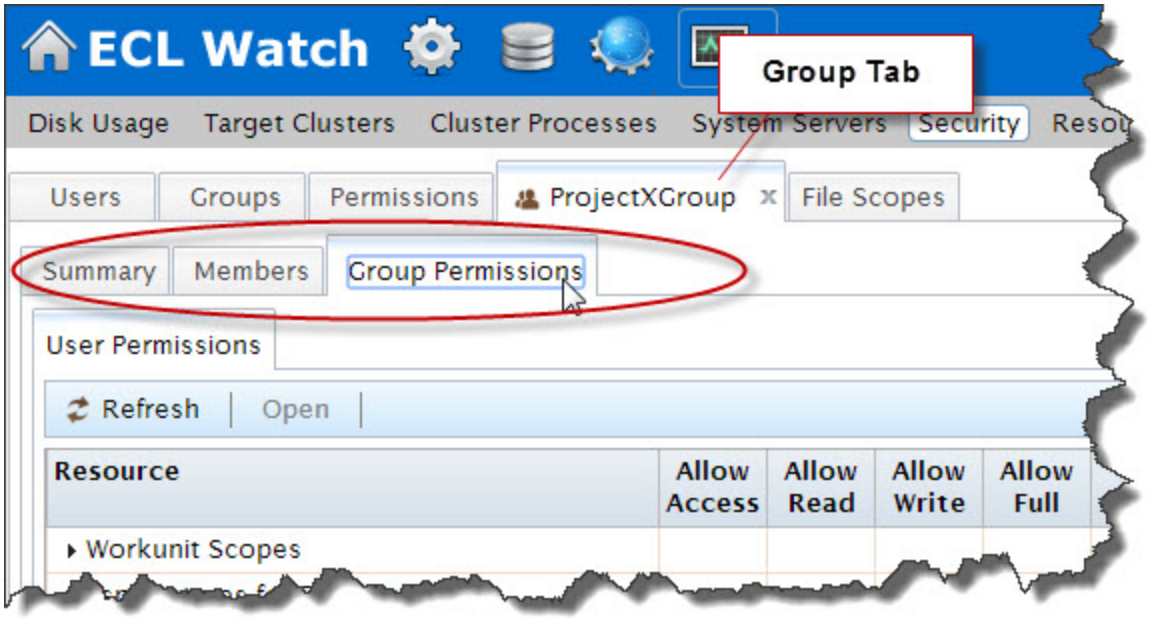


1. Click the **Groups** tab.

2. Locate the group in the list and check the box next to it.

3. Press the **Open** action button.

   This opens a new tab for the group.



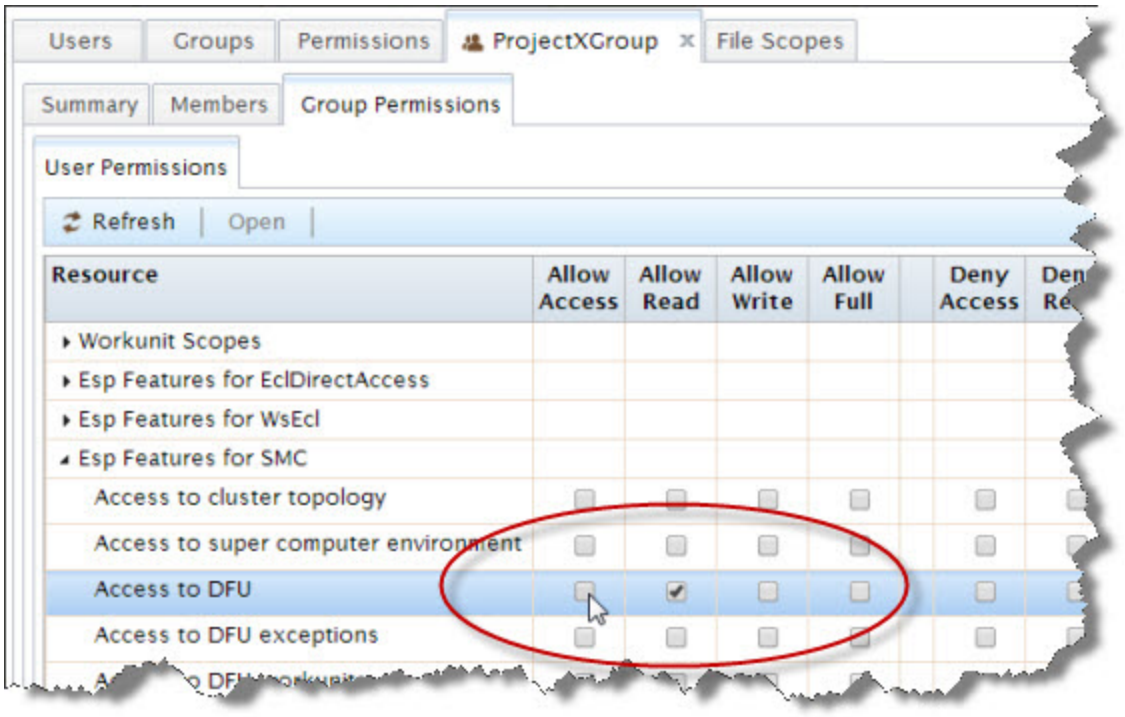
   The group tab displays three sub-tabs: **Summary**, **Members**, and **Group Permissions**.

4. Select the **Group Permissions** tab.

5. Click on the arrow to the left of the resource to display the permissions for that resource. The permission groups currently set for this group and the inherited ones display.

6. There may be more than one resource setting available in each group, be sure to set the permissions for each setting as required.

7. Check the boxes for **allow** and **deny** as required for the group.



	**NOTE:** Use caution when setting any explicit **deny** permission setting. The most restrictive permission always applies.

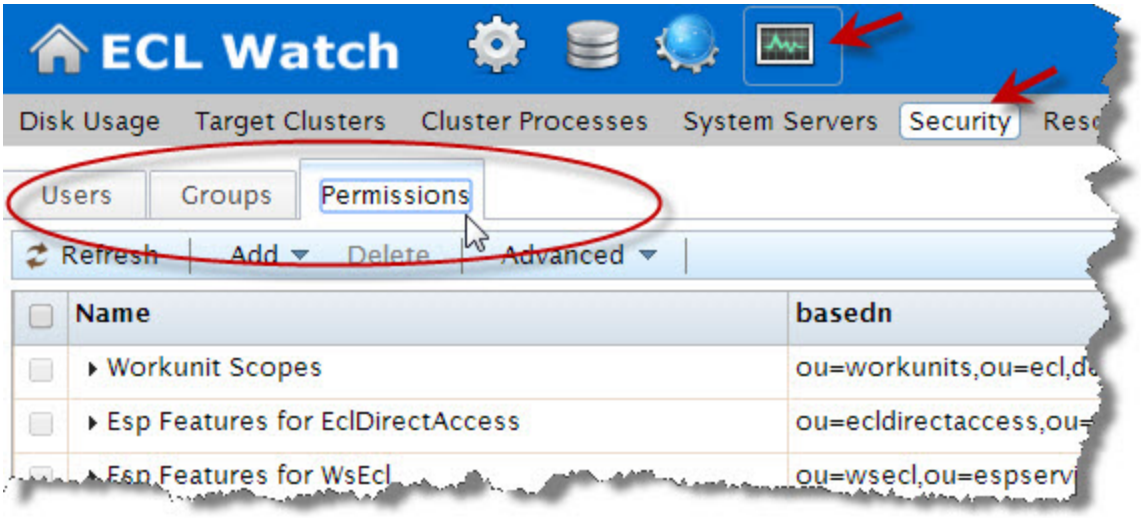
8. There may be more than one resource setting available, select the resource(s) you require from the drop list.

   Repeat for each applicable resource.

9. The changes are automatically saved. Close the tab.

Feature level access control

Access to the feature permissions is available through ECL Watch. To modify feature permissions you must have Administrator level access. To access the feature permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

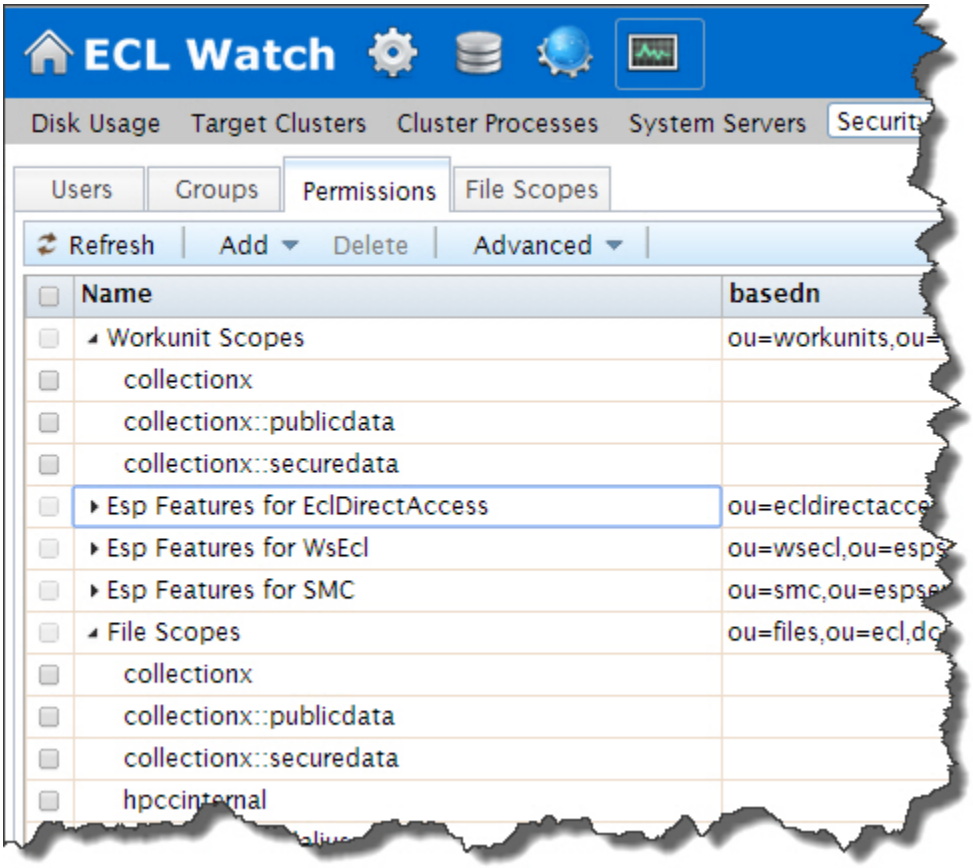


Use the feature level controls on the **Permissions** tab to:

- View the features and permissions for any resource
- Edit the permissions for any feature
- Update the permissions for users and groups for a specific resource

Feature resources

There are three types of features for which you can set up access control in HPCC. Access to features of the HPCC system is controlled by via the **ESP Features for SMC** category. Access to features of WsECL web service are controlled by the **ESP Features for WsEclAccess** category. Access to features of the ECLDirect web service are controlled by the **ESP Features for EclDirectAccess** category.



These features are listed as **Resources** when setting permissions using ECL Watch.

ECL Watch feature permission settings that are not listed are not relevant and should not be used.

Modify permissions for a feature resource:

To use the feature permissions, you must apply them to a user or group(s). To access the feature permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click the **Permissions** tab.

   The resources are listed.

2. Identify the user(s) or group(s) which you want to modify the feature permissions.

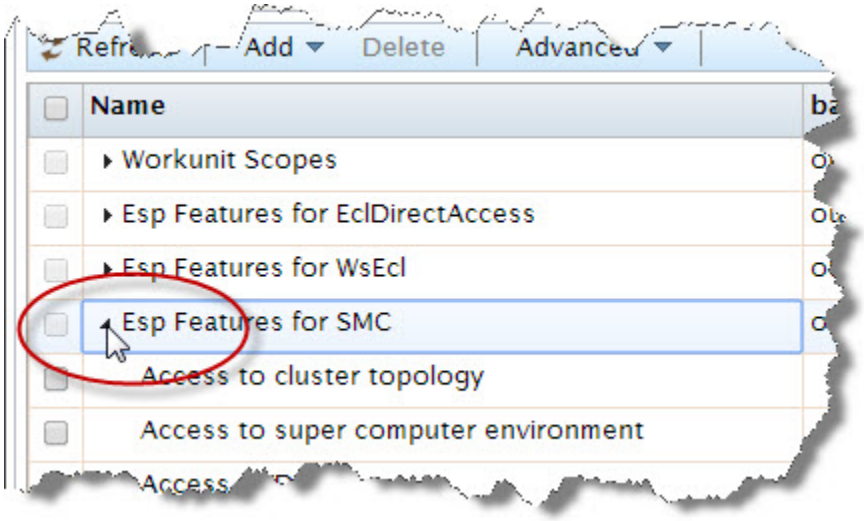
   Select the appropriate tab. (Users or Groups)

3. Check the checkbox(es) next to the user(s) or group(s) to modify.

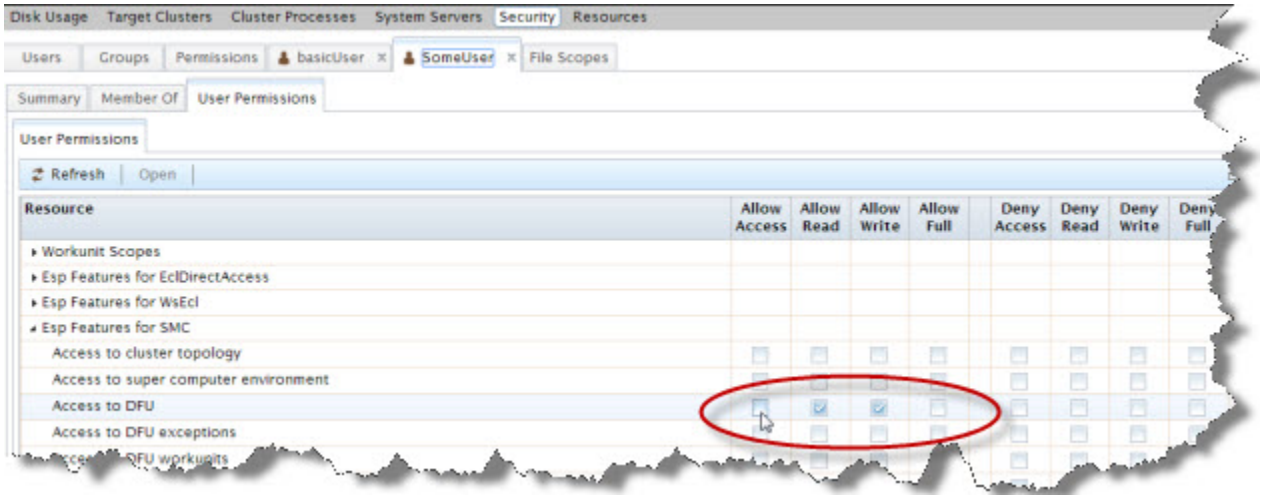
4. Press the **Open** action button. A tab for each user or group selected opens.

5. Click the **User Permissions** sub-tab.

6. Click on the arrow to the left of the resource to display the features of that resource.



7. Locate the feature resource(s) you want to update.



8. Click the checkbox(es) in the **allow** and **deny** columns as appropriate.

9. The changes are automatically saved. Close the tab(s).

   Note: You must follow this process for each user or group(s) separately.

Feature Permissions

The following sections show the level of access required to be able to use ECL Watch features:

Login

SMCAccess is required by all users to be able to successfully login to ECL Watch.

Name	Description	Access
SmcAccess	Root Access to SMC Service	Read

Clusters

Users may be given access to the thor queue which can be manipulated by promoting/demoting queued workunits according to priority. The thor queue can also be paused or cleared and users can view thor usage statistics.

From this page, users can also click on workunit IDs to view details about the workunit. Depending on the level of access given, they can view, modify and delete their own, or others workunits.

Name	Description	Access
ThorQueueAccess	Access to Thor Job Queue Control	Full
RoxieControlAccess	Access to Roxie Process Cluster Control	Full

ECL Workunits

Workunits can also be viewed using this feature of ECL Watch. The contents of the workunits list reflects whether a user has the permission to view their own and others workunits.

Name	Description	Access
OwnWorkunitsAccess	Access to View Own Workunit	Read
	Access to Create or Modify Own Workunit	Write
	Access to Delete Own Workunits	Full
OtherWorkunitsAccess	Access to View Other User's Workunits	Read
	Access to Modify or Resubmit User's Workunits	Write
	Access to Delete Other User's Workunits	Full

Topology

This section shows details about the clusters and other HPCC System components. Preflight provides diagnostic information including disc space, CPU usage and access to logs as well as the ability to swap faulty nodes out of the cluster.

Name	Description	Access
ClusterTopologyAccess	Access to Cluster Topology	Read
	Set Machine Status	Write
	Swap Node	Full
MachineInfoAccess	Access to machine/Preflight Information	Read
MetricsAccess	Access to SNMP Metrics Information (Roxie Metrics)	Read
ExecuteAccess	Access to Remote Execution in ECL Watch	Full

DFU Workunits

A user must have permission to view DFU Workunits and requires other permissions to be able to manipulate them.

Name	Description	Access
DfuWorkunitsAccess	Access to View DFU Workunits	Read
	Access to Create, Delete, Update, Submit, and Abort DFU Workunits	Write

DFU Files

Users need permission to see files on the dropzone and also to put files there. They need further permissions to be able to spray and copy files from the dropzone to their cluster and also to despray files from the cluster back to the dropzone.

XREF is used for monitoring files on the cluster(s). Reports generated show where housekeeping is required on the cluster(s) and users require additional permission to use this feature.

Name	Description	Access
DfuAccess	Access to DFU Logical Files	Read
	Delete Files, add to superfiles	Write
DfuExceptions	Access to DFU Exceptions	Read
DfuWorkunitsAccess	Access to View DFU Workunits	Read
	Access to Create, Delete, Update, Submit, and Abort DFU Workunits	Write
DfuXrefAccess	Access to DFU XREF	Read
	Clean directory	Write
	Make changes and generate XREF Reports	Full
FileDesprayAccess	Access to De-Spraying Files	Write
FileSprayAccess	Access to Spraying and Copying	Read
	Rename files	Write
	Delete from Drop zone	Full
FileIO	Access to read files in Drop zone	Read
	Access to write to files in Drop zone	Write

On a large system, we suggest limiting the number of users who can Generate XREF reports by setting DfuXrefAccess access to FULL for only those users.

Roxie Queries

Additional permission is required to view roxie queries in ECL Watch.

Name	Description	Access
RoxieQueryAccess	Access to Roxie Queries	Read

Users/Permissions

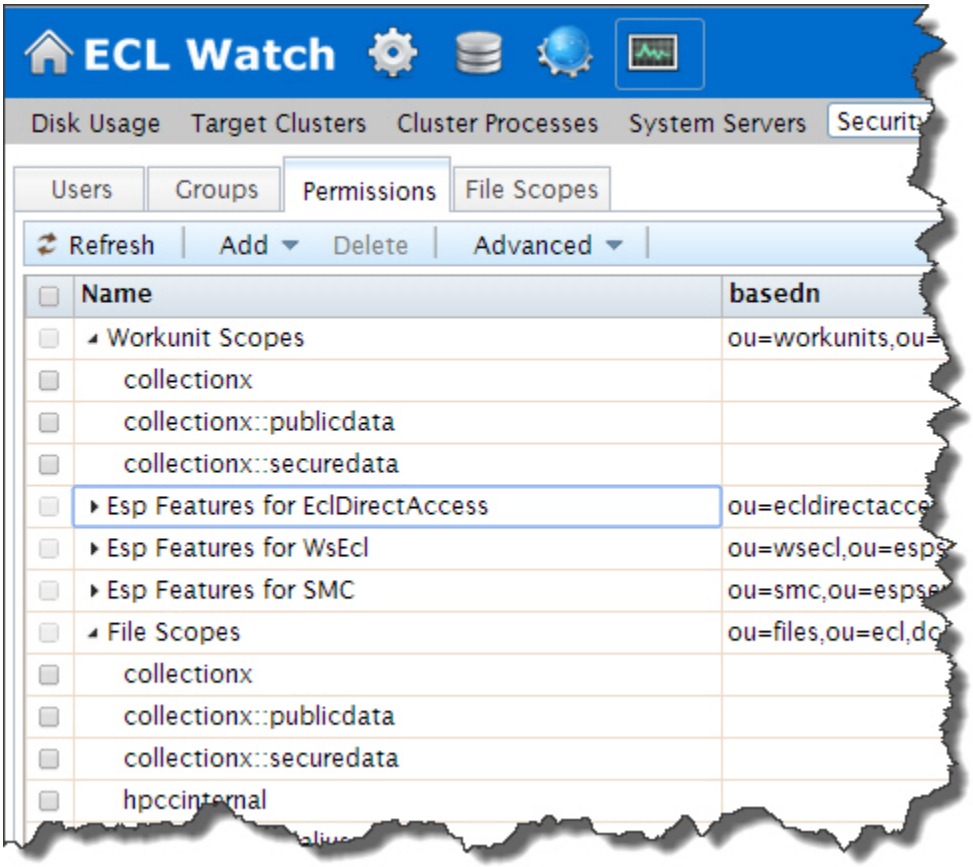
To be able to view the **Users/Permissions** area in ECL Watch, a user must be a member of the Administrators (or similarly named) group with the appropriate permissions on the LDAP or Active Directory server.

File Access Control

The HPCC's LDAP **Dali Server** technology provides the ability to set secure access permissions to data file folders (or file scopes). This is controlled by the use of file scope resources.

An OU called **Files** is automatically created when the Dali server starts. To secure data folders, create a file scope for that folder and apply rights to each scope.

Figure 14. File Scopes Permissions



For example, below **Files** there is a unit (OU) representing the cluster, such as **thor** (or the name that you set up for your cluster). Furthering the example, below that could be a unit named **collectionx** which contains two units, **publicdata** and **securedata.** The **publicdata** folder has rights granted to a large group of users and the **securedata** folder has limited access granted. This allows you to prevent unauthorized users from any access to files in the **securedata** folder.

The structure described above corresponds to this logical structure:

collectionx::securedata

Which corresponds to this physical structure:

/var/lib/HPCCSystems/hpcc-data/thor/collectionx/securedata

All HPCC components and tools respect LDAP file access security. The following exceptions are assumed to be system level or for administrative users:

- Network file access using UNCs, Terminal Services, or SSH.
- Administrative utilities

Attempting to access a file in a folder for which access is not granted will result in one of the following errors:

```
DFS Exception: 4 Create access denied for scope <filepath>
```

or

```
DFS Exception: 3 Lookup access denied for scope <filepath>
```

(where <filepath> is the full logical file scope path)

Creating file scopes

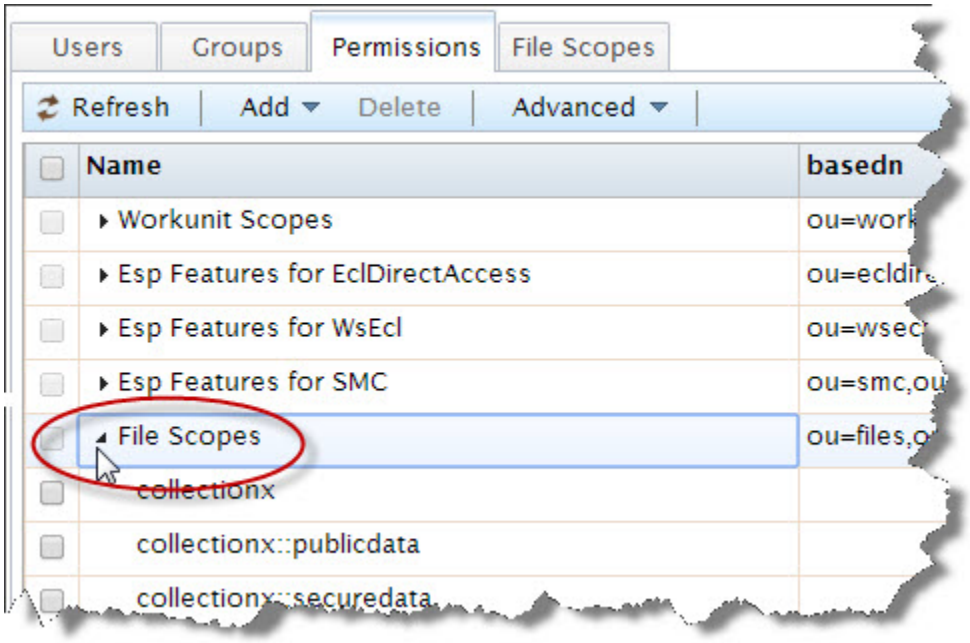
To apply permissions to a file scope, you must first create the file scope(s).

To create file scope(s) click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click the **Permissions** tab.

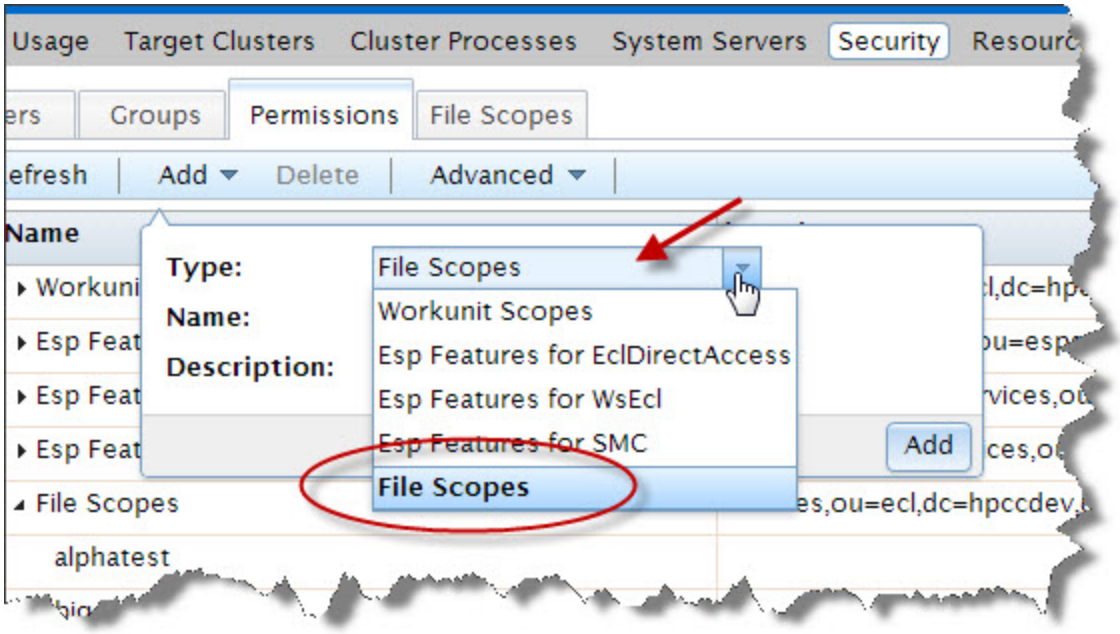
   The feature resources display.

2. Click on the arrow to the left of the **File Scopes** resource to display the file scopes.

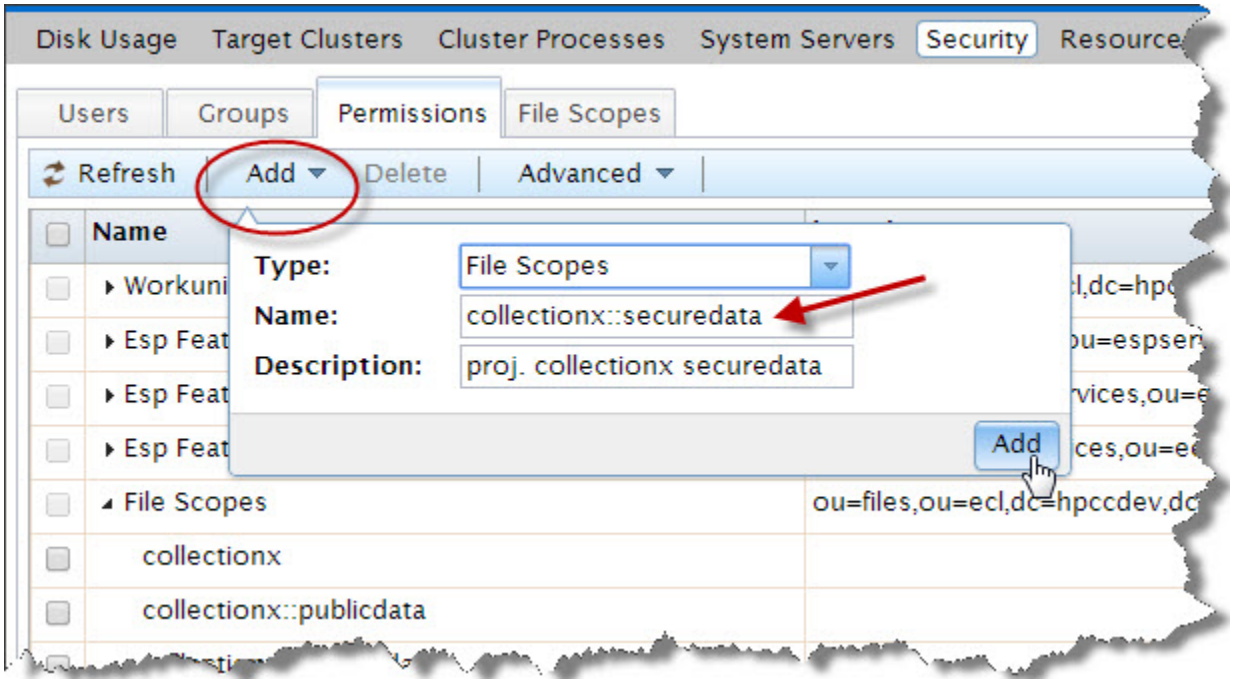


3. Press the **Add** button.

4. Choose **File Scopes** from the drop list.



5. Enter the exact name of the scope you want to add in the **Name** field.



Enter a short description in the **Description field.**

6. Press the **Add** button.

The new scope displays in the list.

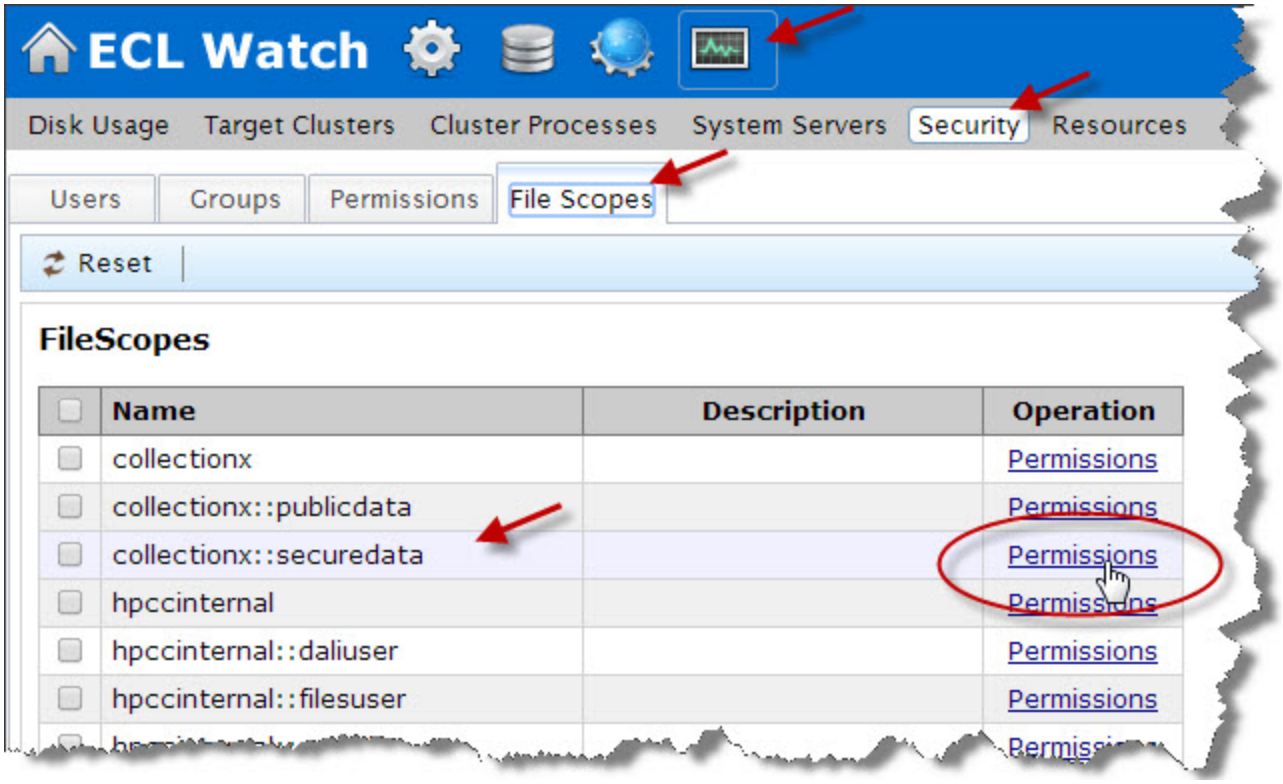
Setting permissions for file scopes

You must apply permissions for file scopes to users or group(s). If you want to apply the scope to a new group, create the group(s) as required.

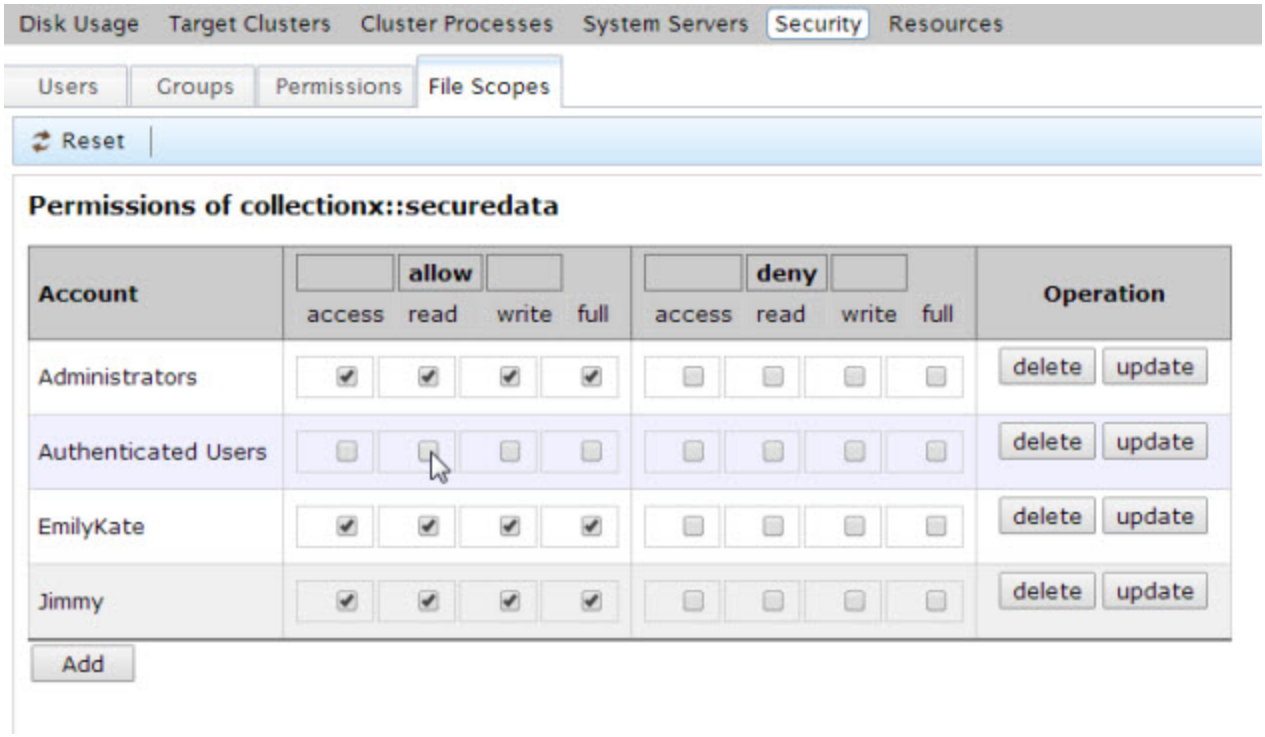
To set the file scope permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Select the **File Scopes** tab.

2. Choose the scope to modify. Click the **Permissions** link for that scope.



3. The permissions defined for users and groups for that scope display.

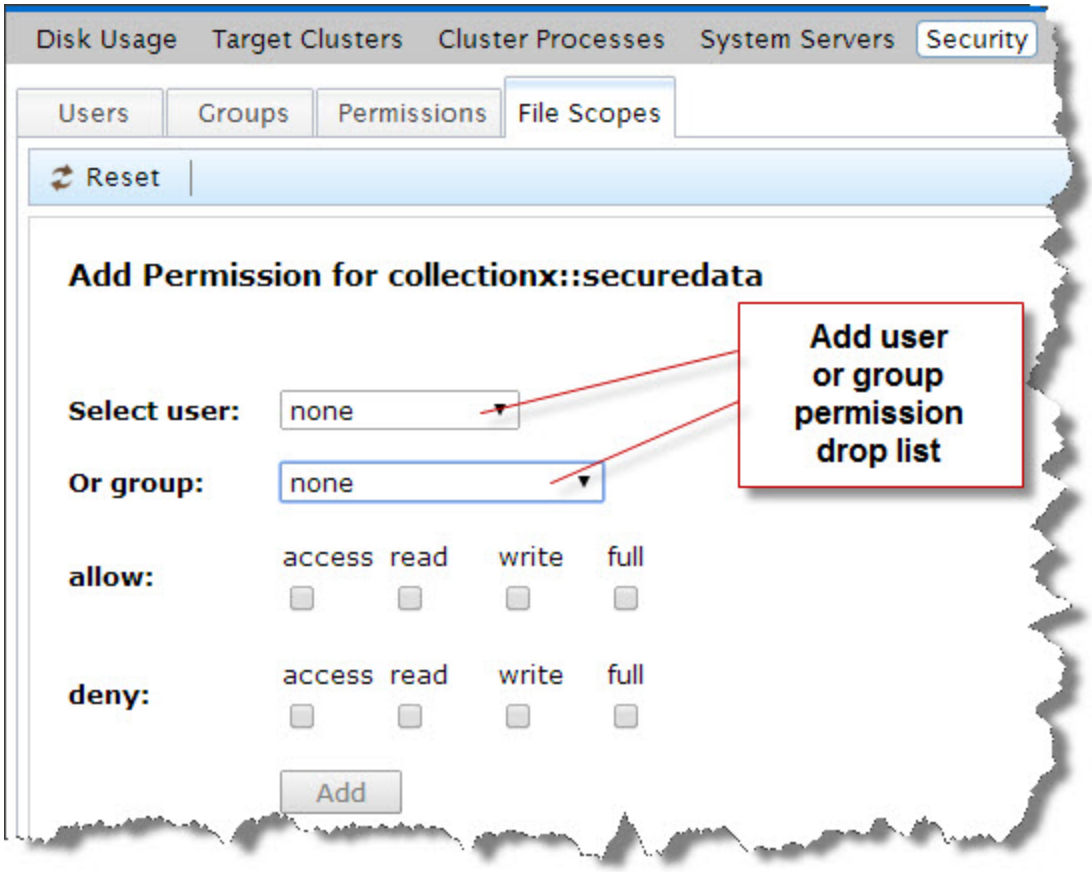


4. Check (or clear) the checkbox(es) in the **allow** and **deny** columns as appropriate for the users or groups displayed.

5. To add users or groups to the scope, press the **Add** button.

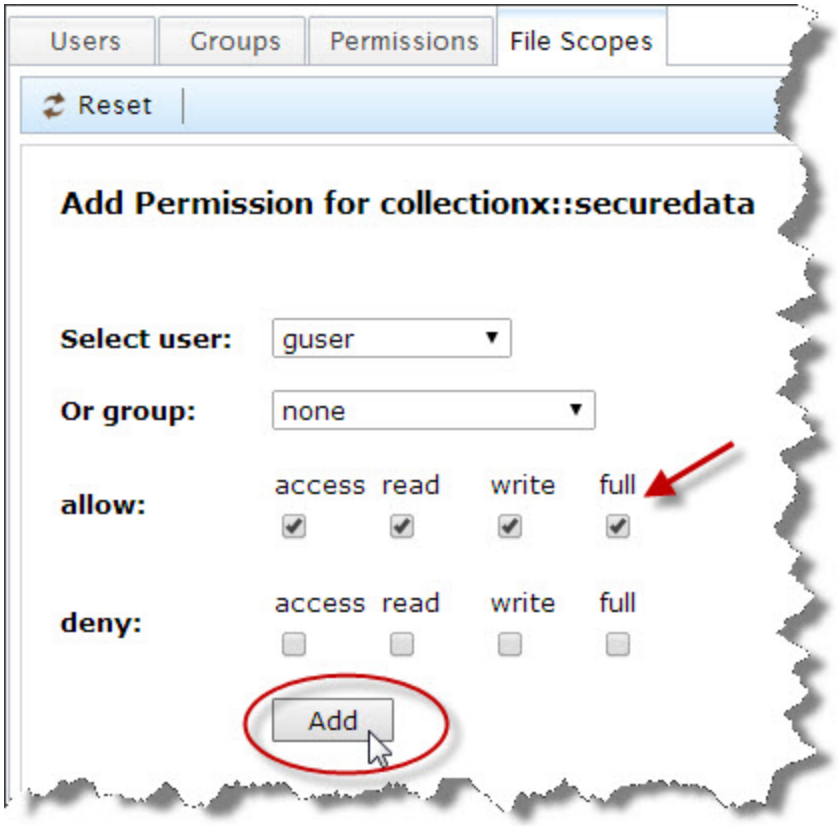
   The Add Permission dialog displays.

6. Select the user or the group to add from the drop list(s).



Once a user or group is selected, the Add button and the allow and deny checkboxes are active

7. Check the boxes for allow and deny as appropriate to set the permissions for this scope.



8. Press the **Add** button.

9. The changes are automatically saved. Close the tab(s).

File scope features

Below the List of File Scopes, there are buttons that allow you to:

- Reset **Default Permissions** to selected file(s)

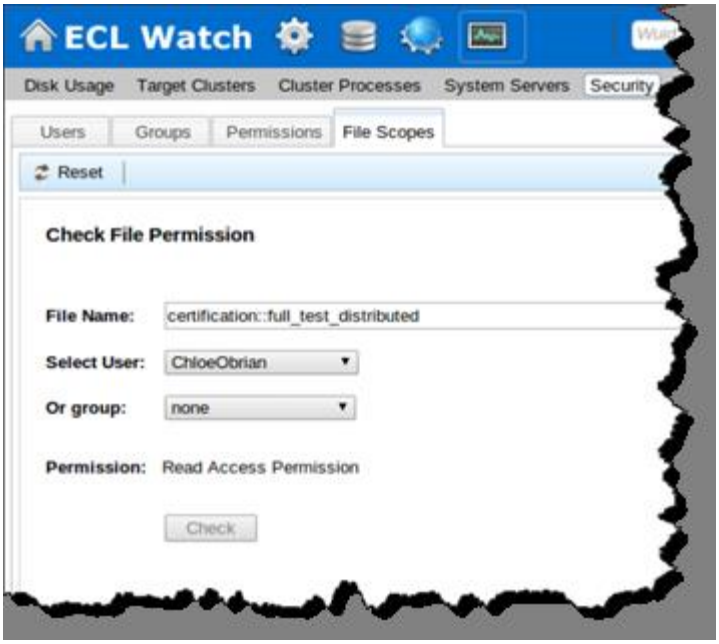
  This allows you to quickly remove any added permission settings for a file and reset to the default access.

- Allow or Deny Access to physical files on Landing Zone

  This provides a way to grant or deny access to the top level file scope. By default, only administrators have access to this scope.

- Check File Permissions for a user or group

  This provides a way to check a user or group's access to a logical file.



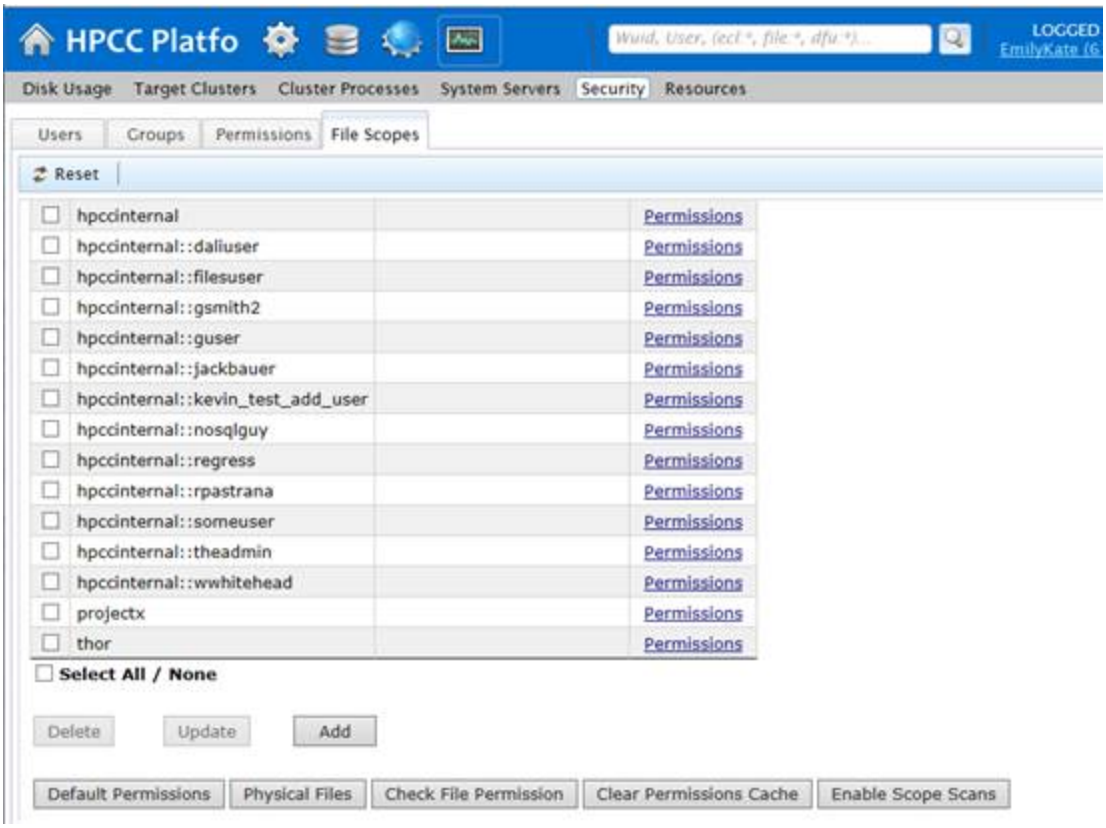
- Clear the Permissions Cache

  This clears the permissions cache and allowing any new permission settings to take effect immediately.

- Enable/Disable Scope Scans

  This provides a means to enable or disable Scope scans. Enable scope scans to check permissions for users to access scopes. This will impact performance. Disable scope scans ignores any scope permissions and removes all access control, but improves performance. Disabling access control is not recommended.

  Changing this setting through ECL Watch, as described here, is only a temporary override. When Dali restarts this setting will revert to what is defined in the configuration environment.xml.



Workunit Access Control

There are 2 aspects of workunit (WU) security:

- Feature Authentication for workunits allows you to set permissions to control whether users can view their own WUs and/or other users' WUs.
- Workunit Scope security provides the ability to set permissions for individual WU scopes. All new workunits have a scope value.

Both methods are valid to use (either separately or together), and the strictest restriction always applys.

In other words, if someone is granted permission to see WUs in the scope *johndoe* but is denied permission to see other users' WUs in the Feature Authentication permissions, this user would be denied access to see the WUs in the *johndoe* scope.

Conversely, if the user is allowed access to see other people's WUs but is denied access to the *johndoe* WU scope, this user will be able to see other WUs in that scope.

Note: If you do not have access to a WU, you will never be able to view it or even know of its existence.

By default, a submitted WU has a scope of the user's ID. For example, a WU JohnDoe submits has *scope=johndoe* in the WU. This value in a WU allows ESP and its services to use LDAP to check for permissions and enforce those permissions.

You can override the default scope using ECL Code:

```
#workunit('scope','MyScopeValue');
```

Securing workunit scopes

ESP (on startup) automatically creates an LDAP OU called **Workunits** (unless it already exists). If this OU is automatically created, the OU is made with full permissions granted to all authenticated users. All WU scopes are below the *workunits* OU either implicitly or explicitly.

If a specific scope OU does not exist in LDAP (e.g., the scope johndoe used in earlier example), then the parent OU's permissions are used. In other words, the scope of *johndoe* is implicitly under the *workunits* OU even though it might not be explicitly listed in the LDAP structure and therefore it would use the permissions granted for the parent, *workunits* .

Workunits feature permissions

Using the **Workunit Scopes** feature in the **Permissions** area of ECL Watch the permissions for any scope can be reset to the default permissions settings for your system. Permission settings for Workunit Scopes may be set as follows:

Description	Access
View WUs in that scope	Read
Create/modify a WU in that scope	Write
Delete a WU in that scope	Full

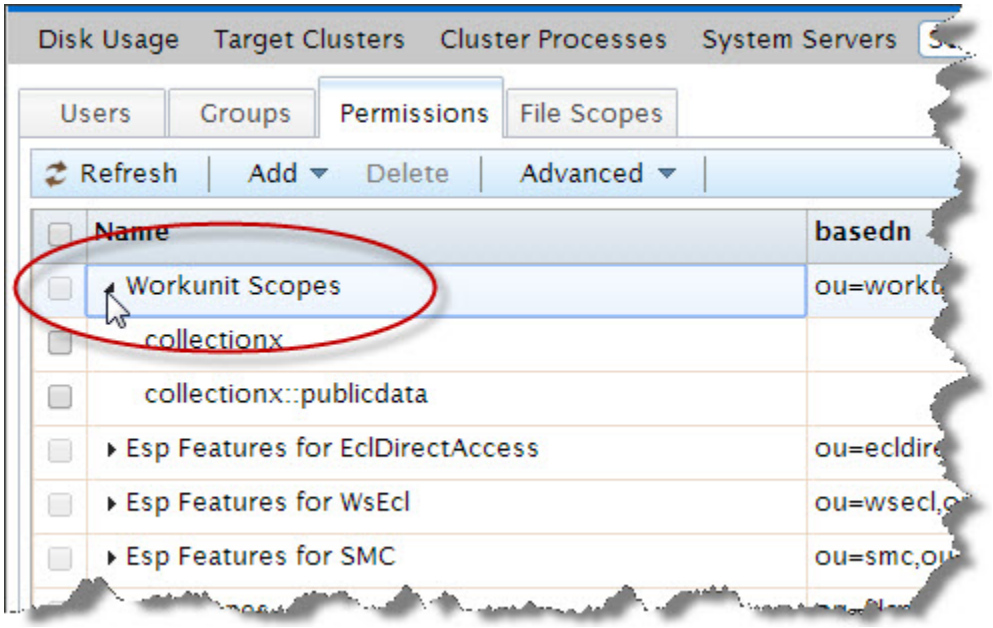
Adding workunit scopes

To add workunit scope permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click the **Permissions** tab.

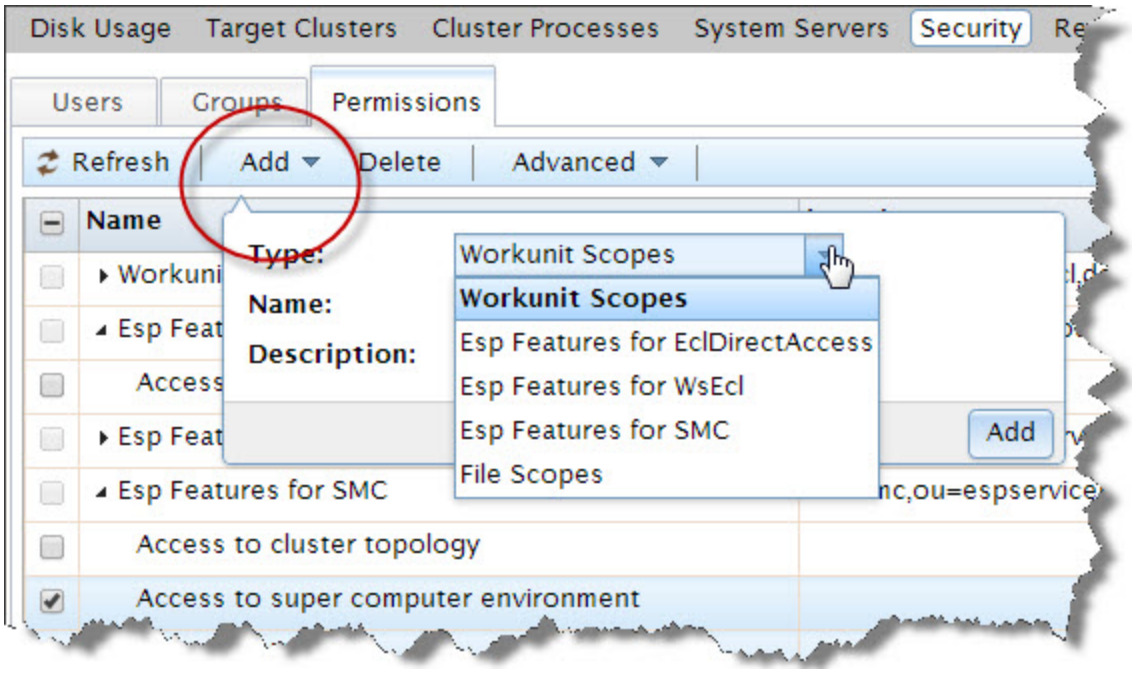
   The feature resources display.

2. Click on the arrow to the left of the **Workunit Scopes** resource to display the file scopes.

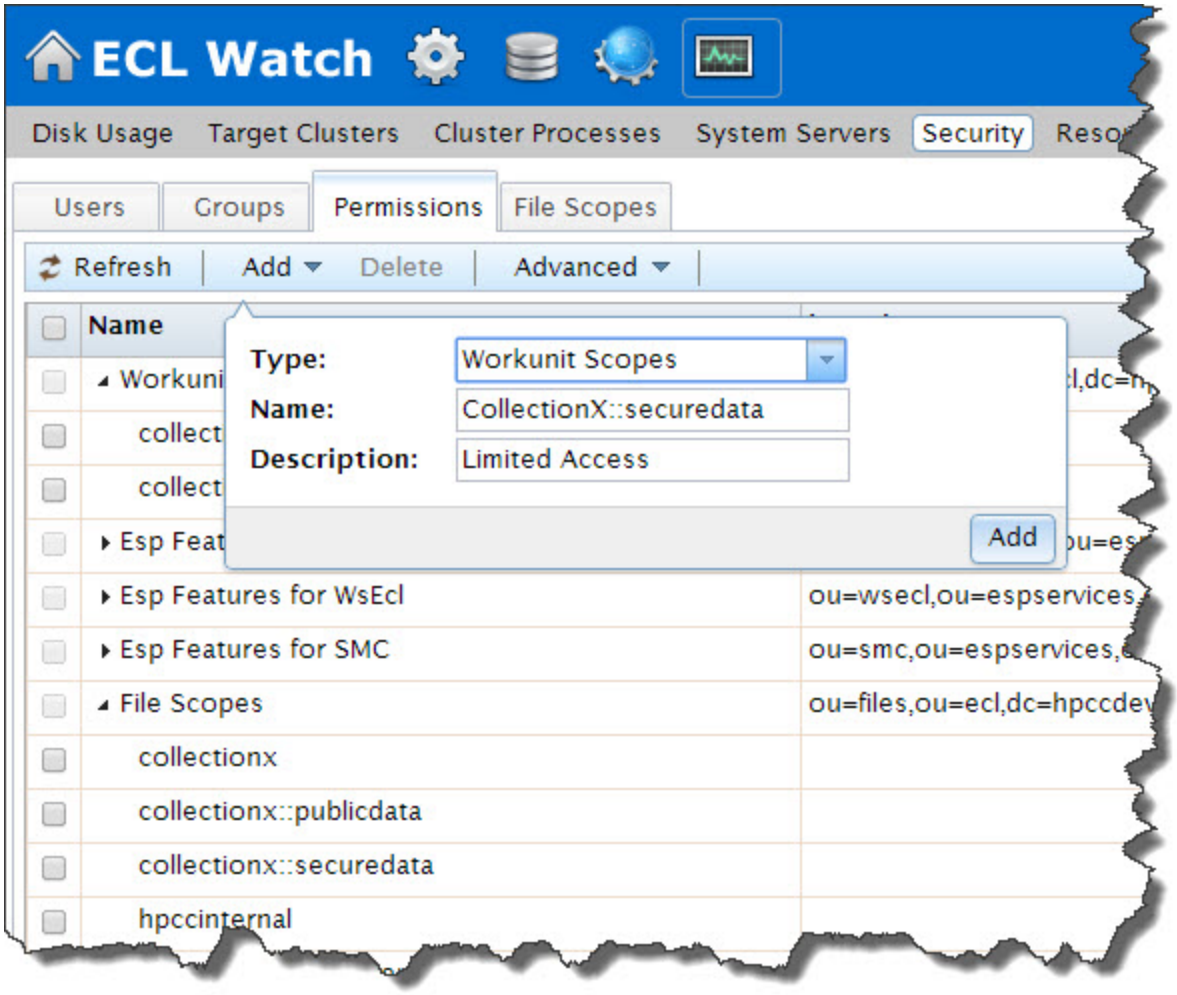


3. Press the **Add** button.

4. Choose **Workunit Scopes** from the drop list.



5. Enter the exact name of the scope you want to add in the **Name** field.



Enter a short description in the **Description field.**

6. Press the **Add** button.

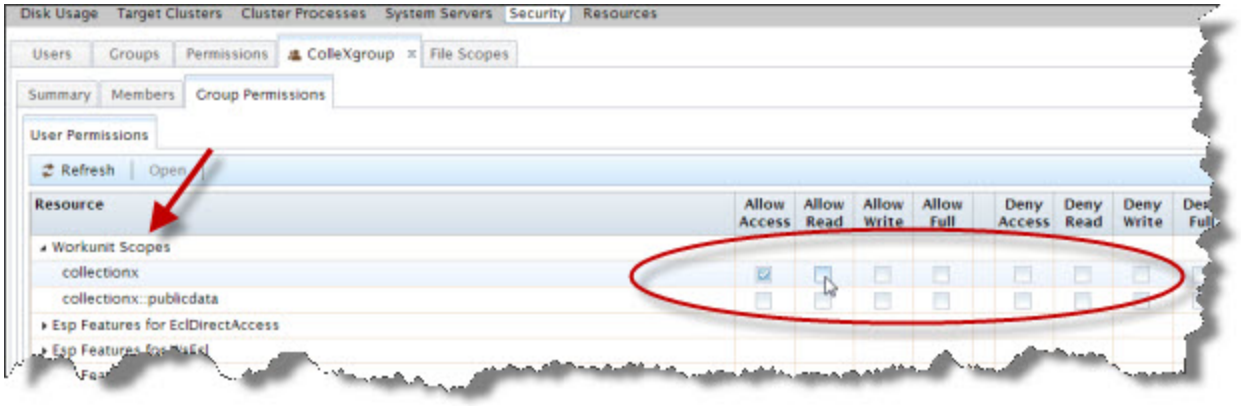
   The new scope displays in the list.

Set permissions to the scope.

You apply the workunit scopes to a group. If you want to apply the scope to a new group, create the group(s) as required.

1. Go to the **Groups** tab.
2. Select a group to apply the scope to by checking the box next to the group name.

   Press the **Open** action button. You can select multiple groups, a tab opens for each group.
3. Select the **Group Permissions** tab of that group. (if multiple groups selected, you must repeat for each group)
4. Click on the arrow to the left of the Workunit Scopes to display the available scopes.



The Workunit scopes display. Check the boxes as appropriate to set the permissions for this scope.

5. To set permissions in this scope for another group, open and go to that groups tab.
6. To set permissions in this scope for a user, select the tab.
7. Select the user and press the Edit action button.

   A new tab for that user opens.
8. On that tab, click on the **User Permissions** sub-tab.
9. Locate the new scope listed under the appropriate Resource.

   Set the access permissions as appropriate for that user.
10. The changes are automatically saved. Close the tab(s).

Permission Caching

When you change a permission in ECL Watch, the settings are cached in the ESP server and stored in the Dali server. The information in the cache is updated at a configurable interval. This value can be set in the Configuration Manager under the LDAP Server settings Attributes tab. The default cacheTimeout is 5 minutes.

When you want a permission change to take effect immediately, you can clear the cache and force Dali to update the permission settings by pressing the **Clear Permissions Cache** button. This action transfers the settings when you press the button. Use this feature judiciously as overall system performance is affected temporarily while the LDAP settings in the Dali System Data Store repopulate.

Configuring ESP Server to use HTTPS (SSL)

The HPCC Enterprise Services Platform server (ESP) supports Secure Sockets Layer (SSL), a protocol used to send and receive private data or documents.

SSL works by using a private key to encrypt and decrypt data transferred over the SSL connection. By convention, URLs using an SSL connection start with HTTPS instead of HTTP.

The SSL option in the ESP Server allows secure and encrypted communication between a browser or SOAP client application and the HPCC platform.

SSL capabilities are configured in the Configuration Manager, but require a certificate be installed on the ESP server. The OpenSSL libraries provide a means to create the necessary certificate files in one of two ways.

- You can use the OpenSSL libraries to create a private key and a Certificate Signing Request (CSR) to purchase a certificate from a Certificate Issuing Authority (such as, VeriSign).
- You can use that CSR to generate your own self-signed certificate and then install the certificate and private key to your ESP Server.

In either case, once installed and configured, the network traffic is encrypted and secure. The Public and Private Keys use 1024-bit RSA encryption.

Generate an RSA Private Key

Use the OpenSSL toolkit to generate an RSA Private Key and a Certificate Signing Request (CSR). This can also be the basis for a self-signed certificate. Self-signed certificates are useful for internal use or testing.

In our example, we create a 1024-bit RSA Private Key which is encrypted using Triple-DES encryption and stored in Privacy Enhanced Mail (PEM) format.

```
openssl genrsa -des3 -out server.key 1024
```

When prompted, provide a passphrase. This is used as the basis for the encryption.

Remember this passphrase as you will need to enter it into the Configuration Manager later.

Generate a CSR (Certificate Signing Request)

After you have a private key, you can use it to create a Certificate Signing Request (CSR). You can use your CSR to request a signed certificate from a Certificate Authority (such as Verisign or Network Solutions). You can also use the CSR to create a self-signed certificate.

```
openssl req -new -key server.key -out server.csr
```

Answer the questions when prompted:

Country Name (2 letter code):	
State or Province Name (full name):	
Locality Name (eg, city) :	
Organization Name (eg, company) :	
Organizational Unit Name (eg, section) :	
Common Name (e.g., server's hostname):	
Email Address :	
A challenge password (optional):	
An optional company name (optional):	

Generate a Self-Signed Certificate

To generate a temporary certificate, which is good for up to 365 days, issue the following command:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

When prompted, enter the passphrase you used earlier when creating your CSR.

Installing the Private Key and Certificate to your ESP Server

You must install the certificate and private key on all ESP server node(s) that will host a service binding using SSL.

Your PrivateKey and certificate must be copied to /var/lib/HPCCSystems/myesp/.

```
# For example:
sudo cp server.crt /var/lib/HPCCSystems/myesp/certificate.cer
sudo cp server.key /var/lib/HPCCSystems/myesp/privatekey.cer
```

Configure HTTPS on your ESP Server

Start Configuration Manager in Advanced Mode

1. Start the Configuration Manager Service on one node (usually the first node is considered the head node and is used for this task, but this is up to you).

   ```
   sudo /opt/HPCCSystems/sbin/configmgr
   ```

2. Using a Web browser, go to the Configuration Manager's interface.

   Use the url of http://nnn.nnn.nnn.nnn:pppp, where nnn.nnn.nnn.nnn is the IP address of the node running Configuration Manager and pppp is the port (default is 8015).

   The Configuration Manager startup wizard displays.

3. Select **Advanced View**.

4. Select an XML file from the drop list.

   This list is populated from versions of an environment XML file in your server's **/etc/HPCCSystems/source/** directory.

   Tip: The XML file that matches the active environment.xml is highlighted.

5. Press the **Next** button.

   The Configuration Manager Advanced View interface displays.

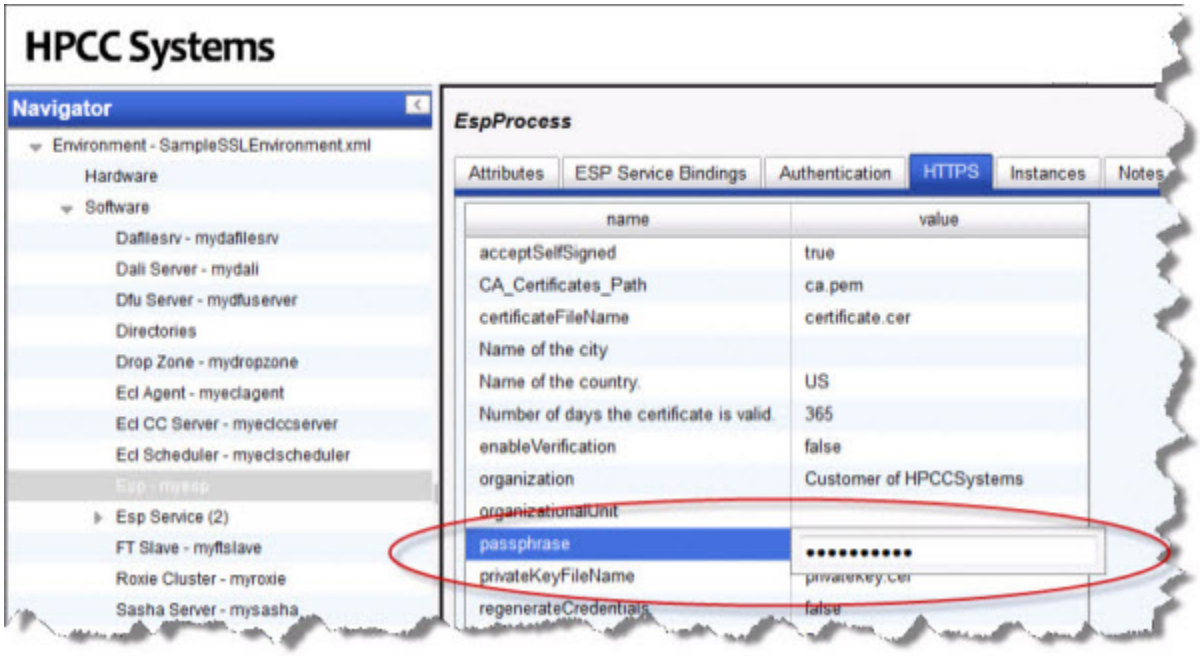
6. Check the **Write Access** box at the top of the page.

Configure ESP

1. Select ESP - MyEsp in the Navigator panel on the left side.

2. Select the **HTTPS** tab.

Figure 15. Select HTTPS Tab



3. In the **passphrase** entry control, enter the passphrase you used earlier when you created the private key.

4. When prompted, provide the passphrase again.

5. Click the disk icon to save.

Configure one or more SSL-Enabled Service Bindings

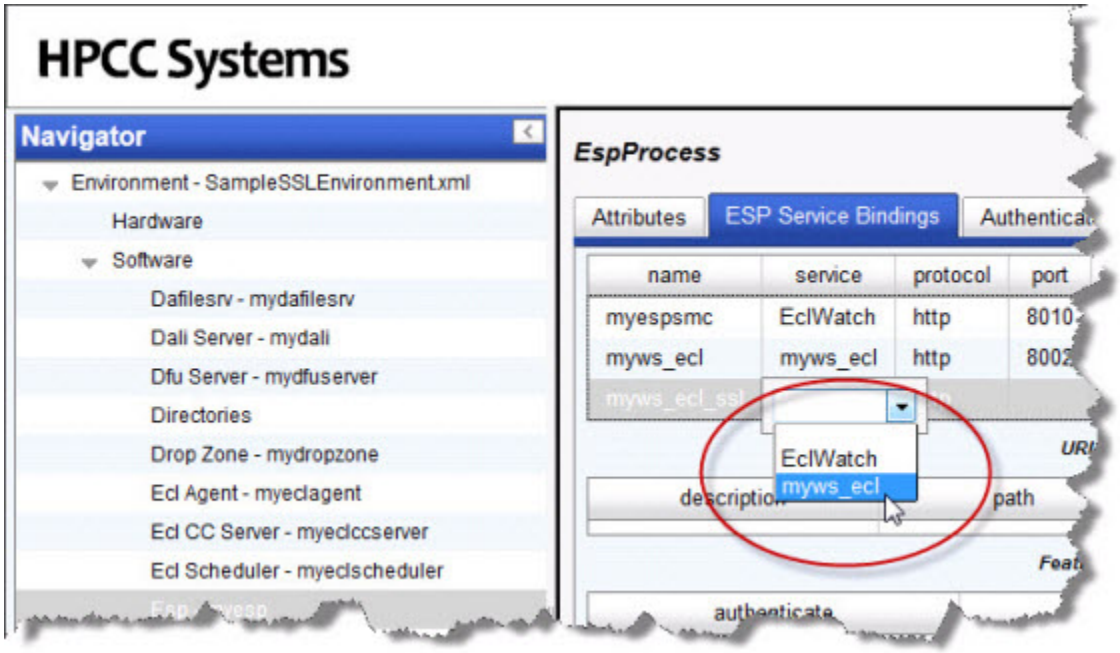
1. Select the ESP Service Bindings tab.

2. Right-click on the list of services, then select **Add**.

3. Provide a name for the binding (e.g., myws_ecl_ssl)

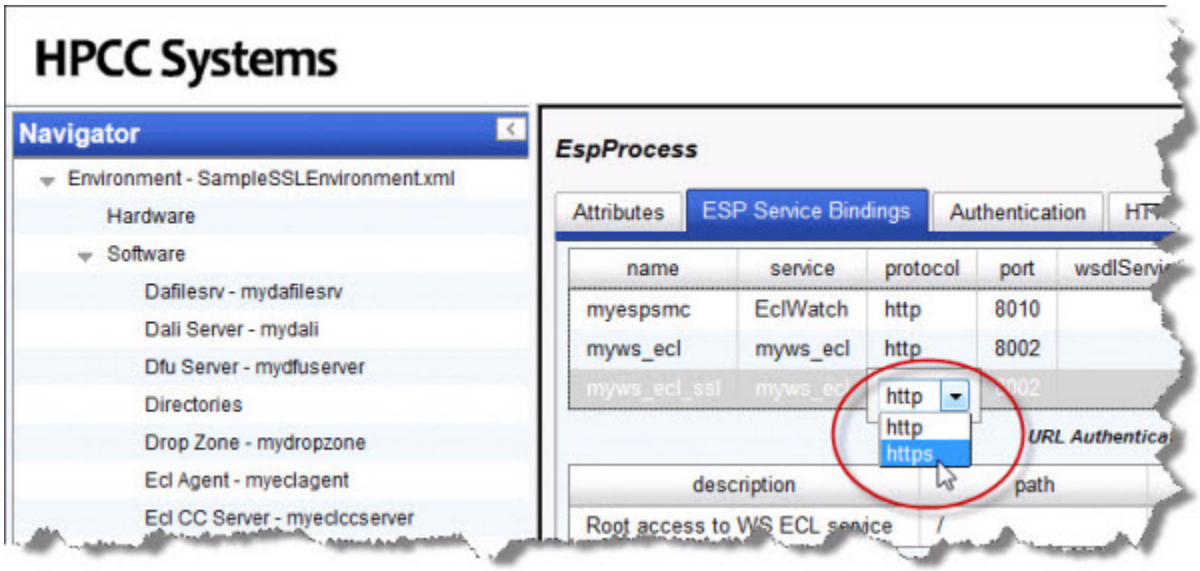
4. Select myws_ecl from the service drop-list.

Figure 16. myws_ecl



5. Select https from the protocol drop-list.

Figure 17. Select HTTPS



Note: If you have not previously edited the port, the change from http to https triggers Configuration Manager to automatically change the port to the default port for https (18002). It only updates automatically if the port has not been edited.

6. Click the disk icon to save

Distribute the environment configuration file to all nodes, Restart, and Certify

Once your environment is set up as desired, you must copy the configuration file out to the other nodes.

1. If it is running, stop the system.

   Make sure system is stopped before attempting to move the environment.xml file.

2. Back up the original environment.xml file

   ```
   # for example
   sudo cp /etc/HPCCSystems/environment.xml /etc/HPCCSystems/environment.bak
   ```

   Note: the "live" environment.xml file is located in your **/etc/HPCCSystems/** directory. ConfigManager works on files in **/etc/HPCCSystems/source** directory. You must copy the XML file from this location to make an environment.xml file active.

3. Copy the NewEnvironment.xml file from the source directory to the /etc/HPCCSystems and rename the file to environment.xml

   ```
   # for example
   sudo cp /etc/HPCCSystems/source/NewEnvironment.xml /etc/HPCCSystems/environment.xml
   ```

4. Copy the **/etc/HPCCSystems/environment.xml** to the **/etc/HPCCSystems/** on every node.

   You might prefer to use a script to automate this step, especially if you have many nodes. See the Example Scripts section in the Appendix of the Installing and Running the HPCCPlatform manual.

5. Restart the HPCC system and certify the components as usual.

More Examples

This section contains additional ECL examples you can use on your HPCC cluster. You can run these on a single-node system or a larger multi-node cluster.

ECL Example: Anagram1

This example takes a STRING and produces every possible anagram from it. This code is the basis for a second example which evaluates which of these are actual words using a word list data file.

1. Open the ECL IDE (Start >> All Programs >> HPCC Systems >> ECL IDE) and login to your HPCC.

2. Open a new **Builder Window** (CTRL+N) and write the following code:

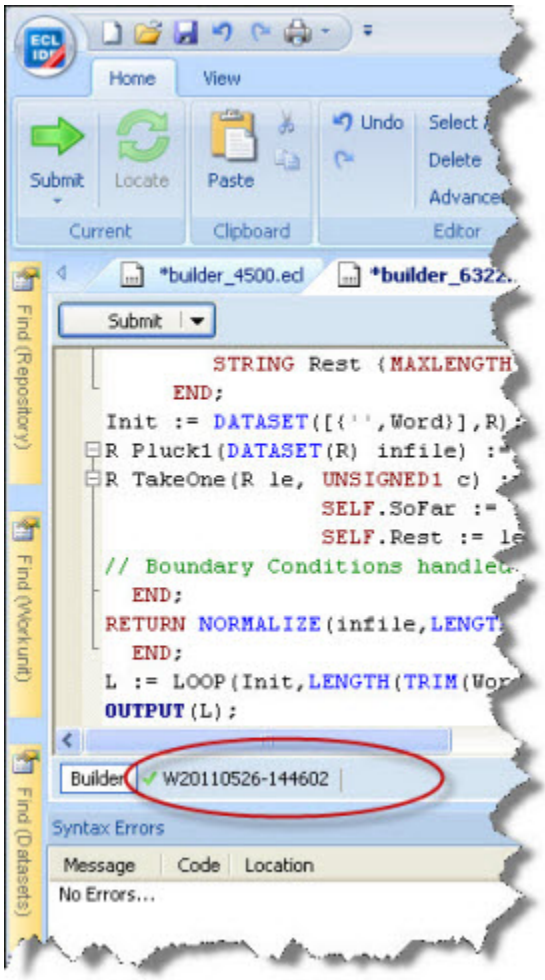
```
STRING Word := 'FRED' :STORED('Word');
R := RECORD
        STRING SoFar {MAXLENGTH(200)};
        STRING Rest {MAXLENGTH(200)};
     END;
Init := DATASET([{'',Word}],R);
R Pluck1(DATASET(R) infile) := FUNCTION
R TakeOne(R le, UNSIGNED1 c) := TRANSFORM
                SELF.SoFar := le.SoFar + le.Rest[c];
                SELF.Rest := le.Rest[..c-1]+le.Rest[c+1..];
// Boundary Conditions handled automatically
  END;
RETURN NORMALIZE(infile,LENGTH(LEFT.Rest),TakeOne(LEFT,COUNTER));
  END;
L := LOOP(Init,LENGTH(TRIM(Word)),Pluck1(ROWS(LEFT)));
OUTPUT(L);
```

3. Select **thor** as your target cluster.

4. Press the syntax check button on the main toolbar (or press F7)

5. Press the **Submit** button (or press ctrl+enter).

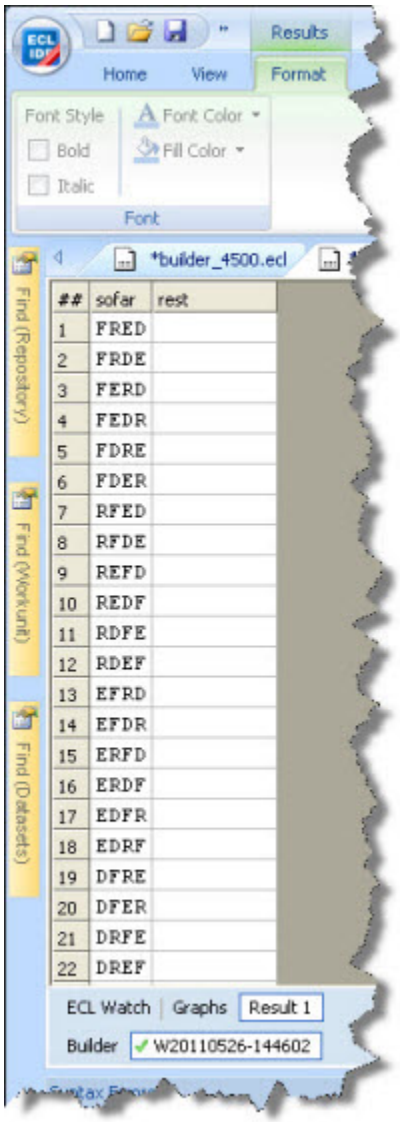
Figure 18. Completed job



The green check mark indicates successful completion.

6. Click on the workunit number tab and then on the Result 1 tab to see the output.

Figure 19. Completed job output



Roxie Example: Anagram2

In this example, we will download an open source data file of dictionary words, spray that file to our Thor cluster, then validate our anagrams against that file so that we determine which are valid words. The validation step uses a JOIN of the anagram list to the dictionary file. Using an index and a keyed join would be more efficient, but this serves as a simple example.

Download the word list

We will download the word list from http://wordlist.sourceforge.net/

1. Download the *Official 12 Dicts* Package. The files are available in tar.gz or ZIP format.

2. Extract the **2of12.txt** file to a folder on your local machine.

Load the Dictionary File to your Landing Zone

In this step, you will copy the data files to a location from which it can be sprayed to your HPCC cluster. A Landing Zone is a storage location attached to your HPCC. It has a utility running to facilitate file spraying to a cluster.

For smaller data files, maximum of 2GB, you can use the upload/download file utility in ECL Watch. This data file is only ~400 kb.

Next you will distribute (or Spray) the dataset to all the nodes in the HPCC cluster. The power of the HPCC comes from its ability to assign multiple processors to work on different portions of the data file in parallel. Even though the VM Edition only has a single node, the data must be sprayed to the cluster.

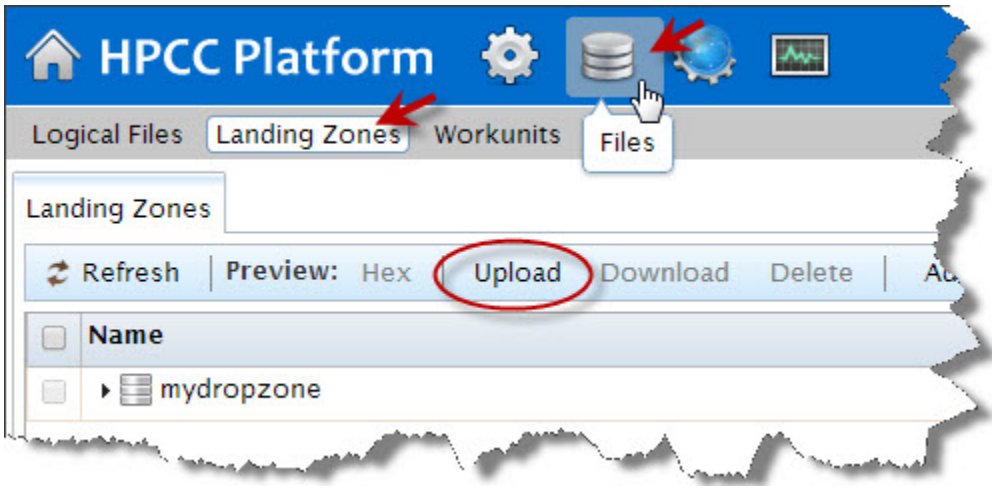
1. In your browser, go to the **ECL Watch** URL. For example, http://nnn.nnn.nnn.nnn:8010, where nnn.nnn.nnn.nnn is your ESP Server's IP address.

	Your IP address could be different from the ones provided in the example images. Please use the IP address provided by **your** installation.

2. From ECL Watch click on the **Files** icon, then click the **Landing Zones** link from the navigation sub-menu.

   Press the **Upload** action button.

Figure 20. Upload



3. A dialog opens. **Browse** your local machine select the file to upload and then press the **Open** button.

Figure 21. File Uploader

   The file you selected should appear in the **File Name** field. The data file is named: **2of12.txt**.

4. Press the **Start** button to complete the file upload.

Spray the Data File to your *Data Refinery (Thor) Cluster*

To use the data file in our HPCC system, we must "spray" it to all the nodes. A *spray* or *import* is the relocation of a data file from one location (such as a Landing Zone) to multiple file parts on nodes in a cluster.

The distributed or sprayed file is given a *logical-file-name* as follows**: ~thor::word_list_csv** The system maintains a list of logical files and the corresponding physical file locations of the file parts.

1. Open ECL Watch using the following URL:

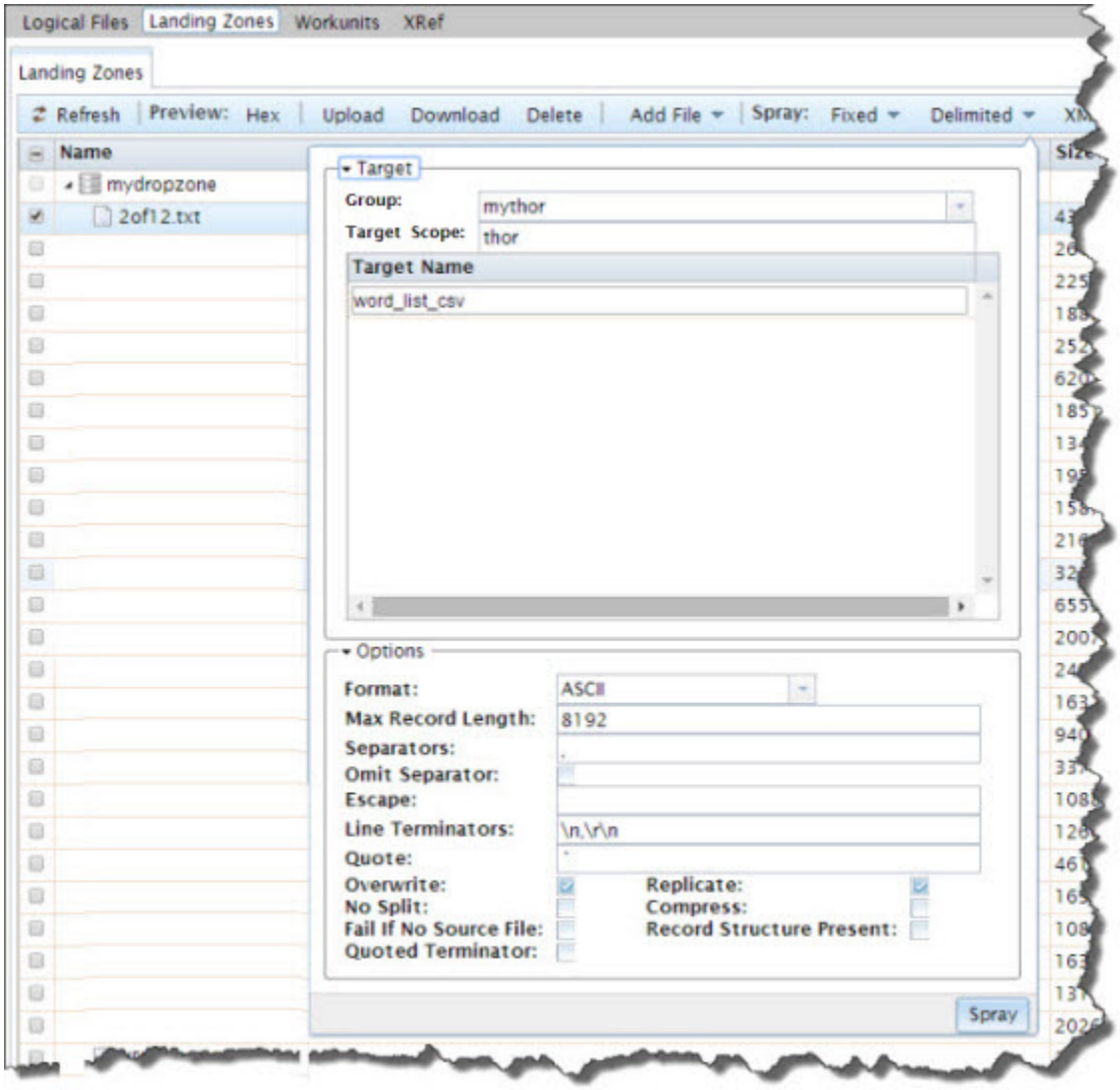
   http://nnn.nnn.nnn.nnn:pppp(where nnn.nnn.nnn.nnn is your ESP Server's IP Address and pppp is the port. The default port is 8010)

2. Click on the **Files** icon, then click the **Landing Zones** link from the navigation sub-menu. Select the appropriate landing zone (if there are more than one landing zones). Click the arrow to the left of your landing zone to expand it.

3. Select the file from your drop zone by checking the box next to it.

4. Check the box next to 2of12.txt, then press the **Delimited** button.

Figure 22. Spray Delimited



The **DFU Spray Delimited** page displays.

5. Select mythor in the Target Group drop list.

6. Complete the Target Scope as *thor*.

7. Fill in the rest of the parameters (if they are not filled in already).

   - Max Record Length 8192
   - Separator \,
   - Line Terminator \n,\r\n
   - Quote: '

8. Fill in the Target Name using the rest of the Logical File name desired: word_list_csv

9. Make sure the **Overwrite** box is checked.

   If available, make sure the **Replicate** box is checked. (The Replicate option is only available on systems where replication has been enabled.)

10. Press the **Spray** button.

    A tab displays the DFU Workunit where you can see the progress of the spray.

Run the query on Thor

1. Open a new **Builder Window** (CTRL+N) and write the following code:

```
IMPORT Std;
layout_word_list := record
  string word;
end;
File_Word_List := dataset('~thor::word_list_csv', layout_word_list,
                          CSV(heading(1),separator(','),quote('')));
STRING Word := 'teacher' :STORED('Word');
STRING SortString(STRING input) := FUNCTION
  OneChar := RECORD
    STRING c;
  END;
  OneChar MakeSingle(OneChar L, unsigned pos) := TRANSFORM
    SELF.c := L.c[pos];
  END;
  Split := NORMALIZE(DATASET([input],OneChar), LENGTH(input),
  MakeSingle(LEFT,COUNTER));
  SortedSplit := SORT(Split, c);
  OneChar Recombine(OneChar L, OneChar R) := TRANSFORM
    SELF.c := L.c+R.c;
  END;
  Recombined := ROLLUP(SortedSplit, Recombine(LEFT, RIGHT),ALL);
  RETURN Recombined[1].c;
END;

STRING CleanedWord := SortString(TRIM(Std.Str.ToUpperCase(Word)));

R := RECORD
  STRING SoFar {MAXLENGTH(200)};
  STRING Rest {MAXLENGTH(200)};
END;
Init := DATASET([{'',CleanedWord}],R);
R Pluck1(DATASET(R) infile) := FUNCTION
  R TakeOne(R le, UNSIGNED1 c) := TRANSFORM
    SELF.SoFar := le.SoFar + le.Rest[c];
```

```
    SELF.Rest := le.Rest[..c-1]+le.Rest[c+1..];
    // Boundary Conditions
    // handled automatically
  END;
  RETURN DEDUP(NORMALIZE(infile,LENGTH(LEFT.Rest),TakeOne(LEFT,COUNTER)));
END;
L := LOOP(Init,LENGTH(CleanedWord),Pluck1(ROWS(LEFT)));
ValidWords := JOIN(L,File_Word_List,
LEFT.SoFar=Std.Str.ToUpperCase(RIGHT.Word),TRANSFORM(LEFT));
OUTPUT(CleanedWord);
COUNT(ValidWords);
OUTPUT(ValidWords)
```

2. Select **thor** as your target cluster.

3. Press the syntax check button on the main toolbar (or press F7)

4. Press the **Submit** button.

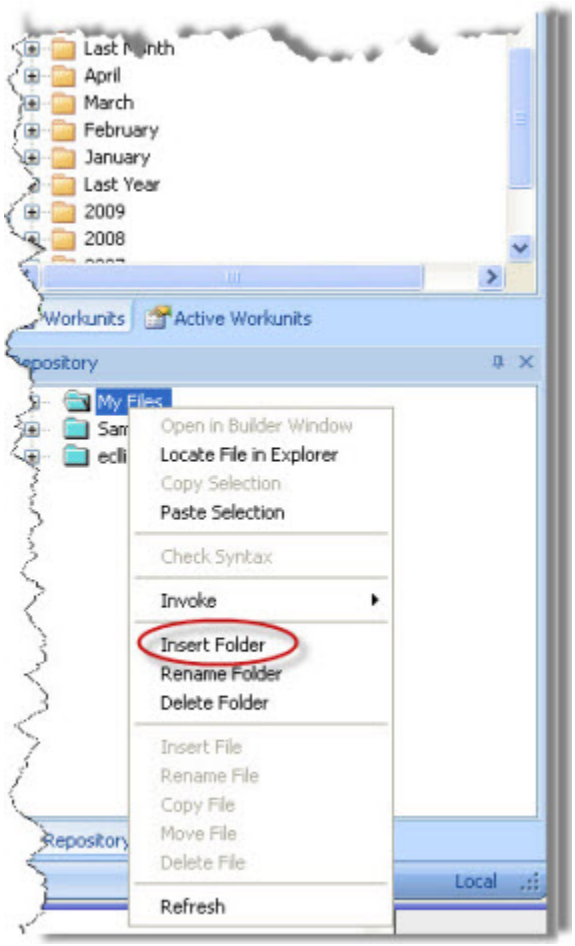
5. When it completes, select the Workunit tab, then select the Result tab.

6. Examine the result.

Compile and Publish the query to Roxie

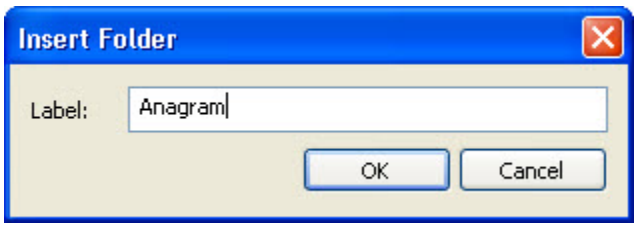
1. Right-click on the **My Files** folder in the Repository window, and select **Insert Folder** from the pop-up menu.

Figure 23. Insert Folder



2. Enter **Anagram** for the label, then press the OK button.

Figure 24. Enter Folder Label

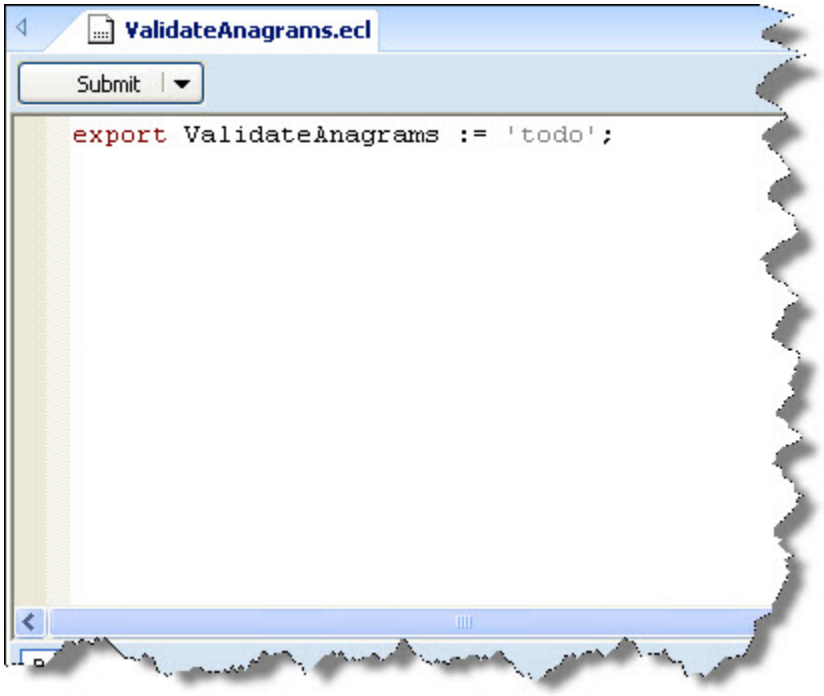


3. Right-click on the **Anagram** Folder, and select **Insert File** from the pop-up menu.

4. Enter **ValidateAnagrams** for the label, then press the OK button.

   A Builder Window opens.

Figure 25. Builder Window



5. Write the following code (you can copy the code from the other builder window):

```
IMPORT Std;
layout_word_list := record
  string word;
end;
File_Word_List := dataset('~thor::word_list_csv', layout_word_list,
                          CSV(heading(1),separator(','),quote('')));
STRING Word := 'teacher' :STORED('Word');
STRING SortString(STRING input) := FUNCTION
  OneChar := RECORD
    STRING c;
  END;
  OneChar MakeSingle(OneChar L, unsigned pos) := TRANSFORM
    SELF.c := L.c[pos];
  END;
  Split := NORMALIZE(DATASET([input],OneChar), LENGTH(input),
  MakeSingle(LEFT,COUNTER));
  SortedSplit := SORT(Split, c);
  OneChar Recombine(OneChar L, OneChar R) := TRANSFORM
    SELF.c := L.c+R.c;
  END;
  Recombined := ROLLUP(SortedSplit, Recombine(LEFT, RIGHT),ALL);
  RETURN Recombined[1].c;
END;

STRING CleanedWord := SortString(TRIM(Std.Str.ToUpperCase(Word)));

R := RECORD
  STRING SoFar {MAXLENGTH(200)};
```

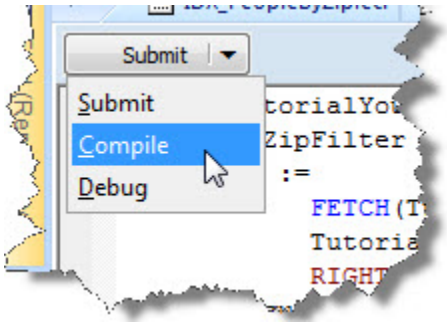
```
  STRING Rest {MAXLENGTH(200)};
END;
Init := DATASET([{'',CleanedWord}],R);
R Pluck1(DATASET(R) infile) := FUNCTION
  R TakeOne(R le, UNSIGNED1 c) := TRANSFORM
    SELF.SoFar := le.SoFar + le.Rest[c];
    SELF.Rest := le.Rest[..c-1]+le.Rest[c+1..];
    // Boundary Conditions
    // handled automatically
  END;
  RETURN DEDUP(NORMALIZE(infile,LENGTH(LEFT.Rest),TakeOne(LEFT,COUNTER)));
END;
L := LOOP(Init,LENGTH(CleanedWord),Pluck1(ROWS(LEFT)));
ValidWords := JOIN(L,File_Word_List,
LEFT.SoFar=Std.Str.ToUpperCase(RIGHT.Word),TRANSFORM(LEFT));
OUTPUT(CleanedWord);
COUNT(ValidWords);
OUTPUT(ValidWords)
```

6. Select **Roxie** as your target cluster.

7. Press the syntax check button on the main toolbar (or press F7)

8. In the Builder window, in the upper left corner the **Submit** button has a drop down arrow next to it. Select the arrow to expose the **Compile** option.

Figure 26. Compile

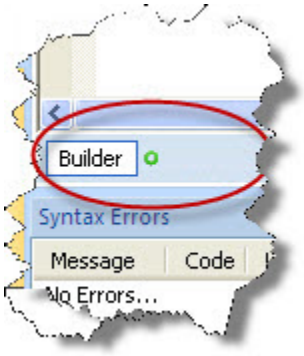


9. Select **Compile**

10. When it completes, select the Workunit tab, then select the Result tab.

11. When the workunit finishes, it will display a green circle indicating it has compiled.

Figure 27. Compiled



Publish the Roxie query

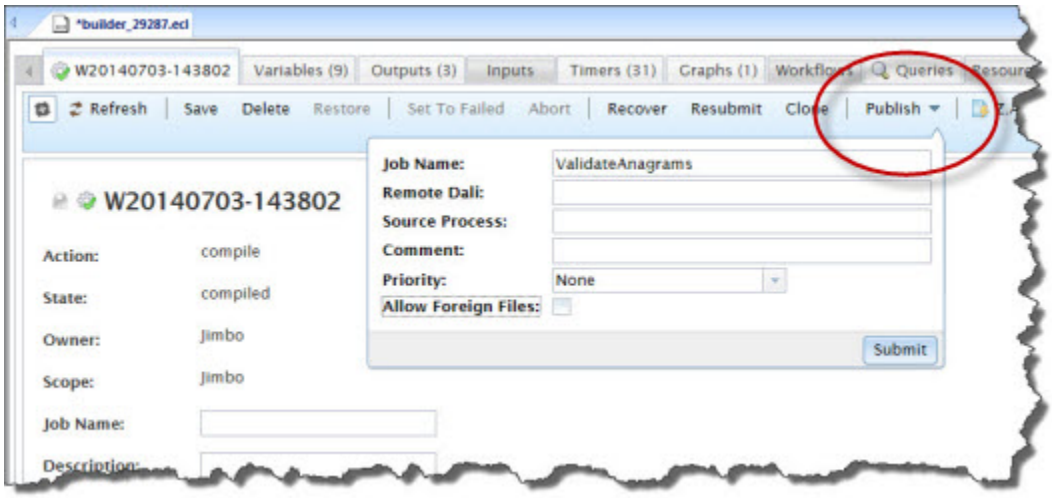
Next we will publish the query to a Roxie Cluster.

1. Select the workunit tab for the ValidateAnagrams that you just compiled.

2. Select the ECL Watch tab.

3. Press the **Publish** button, complete the dialog, and press **Submit**.

Figure 28. Publish Query



When it successfully publishes, a confirmation message displays.

Run the Roxie Query in WsECL

Now that the query is published to a Roxie cluster, we can run it using the WsECL service. WsECL is a web-based interface to queries on an HPCC platform. Use the following URL:

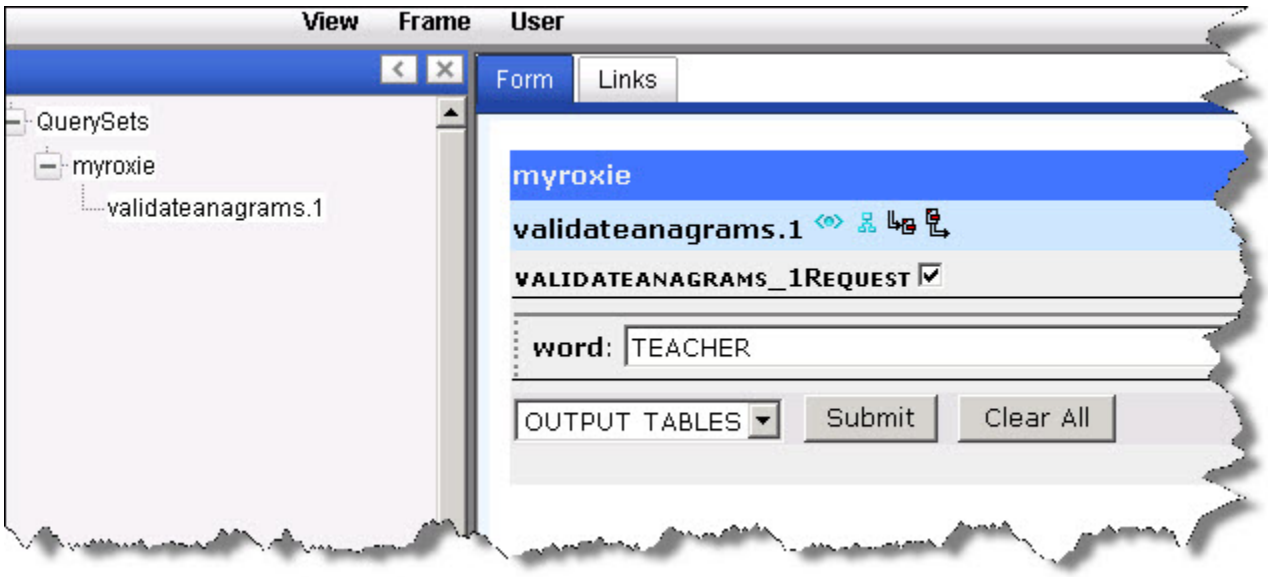
http://nnn.nnn.nnn.nnn:pppp (where nnn.nnn.nnn.nnn is your ESP Server's IP address and pppp is the port. The default port is 8002)

1. Click on the + sign next to **myroxie** to expand the tree.

2. Click on the **ValidateAnagrams.1** hyperlink.

   The form for the service displays.

Figure 29. RoxieECL

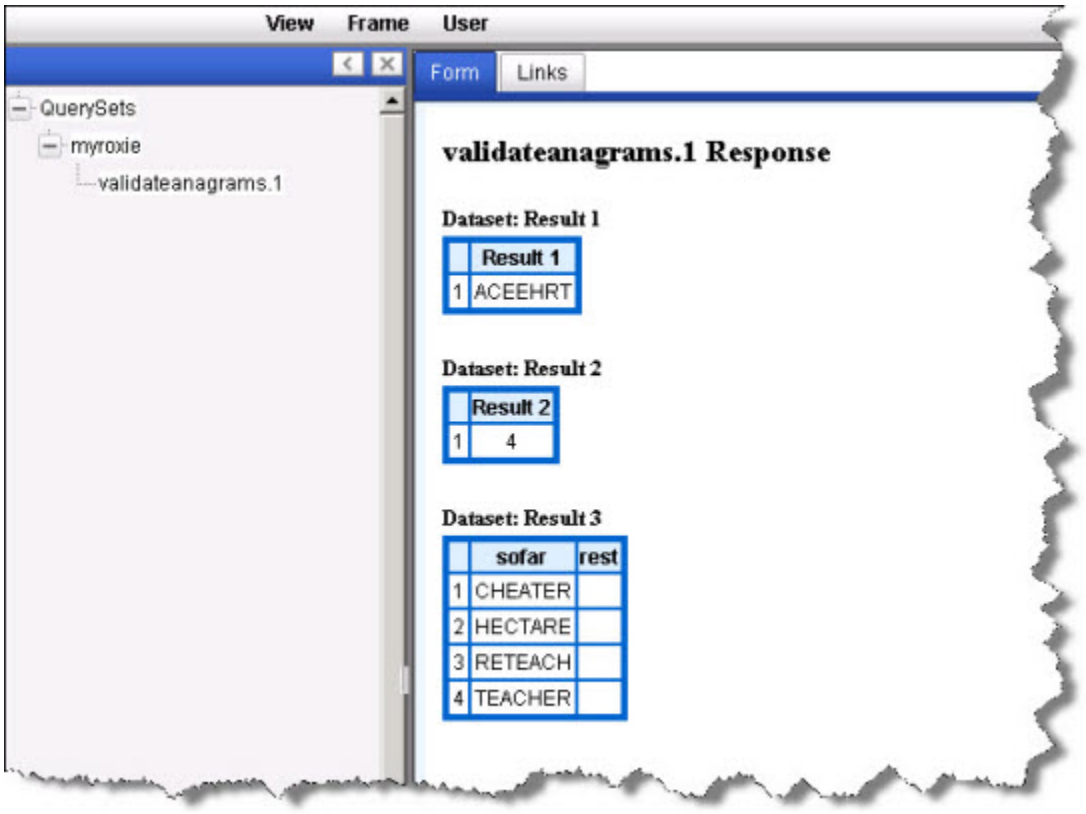


3. Select Output Tables in the drop list.

4. Provide a word to make anagrams from (e.g., TEACHER), then press the Submit button.

   The results display.

Figure 30. RoxieResults

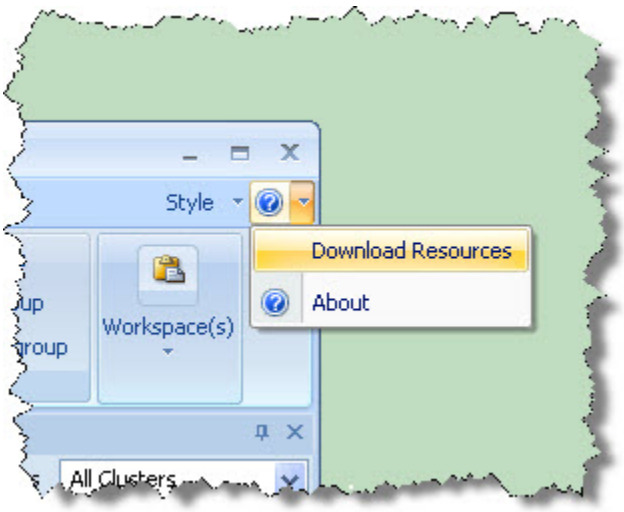


Next Steps

Available from the menu in the ECL IDE there are several documents which provide details on various aspects of the HPCC.

You can access them from the help menu: Help >> Documentation.

Figure 31. Help Menu



You can also find these from the **Start** menu :

Start >> All Programs >> HPCC Systems >> ECL IDE >> Docs

To familiarize yourself with what your system can do we recommend following the steps in

- The **HPCC Data Tutorial**
- **The Six Degrees of Kevin Bacon** example
- Read **Using Config Manager** to learn how to configure an HPCC platform using Advanced View.
- Use your new skills to process your own massive dataset!

The HPCC Systems® Portal is also a valuable resource for more information including:

- Video Tutorials
- Additional examples
- White Papers
- Documentation

Appendix

Example Scripts

For a multi-node configuration, you must install the packages on each node. You can install each one manually or use scripts to copy and install the packages. On a large system where you have many nodes copying and installing on every node is not practical, therefore we provide some scripts you can use or to serve as examples to give you a start in making your own.

Scripts are installed to the **/opt/HPCCSystems/sbin** directory. Scripts should be run as sudo or as a user with appropriate privileges on all nodes. The scripts have the ability to multi-thread.

Make sure that you have the sufficient privileges to sudo as an administrator to use the install-cluster.sh script. To use the hpcc-push.sh or hpcc-run.sh scripts, you must sudo as user **hpcc**.

install-cluster.sh

install-cluster.sh [-k | -p <directory>] [-n <value>] <package-name>

<package-name>	Name of the HPCC package to install. Required
-h	Help. Optional.
-k, --newkey	When specified, the script generates and distributes ssh keys to all hosts. Optional.
-p, --pushkeydir	Push existing ssh key to remote machine. Optional. Use either -k or -p, not both.
-n, --concurrent	When specified, denotes the number of concurrent executions. Default is 5. Optional.

You can run this script as any user with sufficient permissions to execute it; however, when prompted for username/password, you must provide credentials for a user with sufficient sudo rights to run commands as an administrator on all nodes.

Before you can use this script, you must have already defined and generated an environment.xml file (using ConfigMgr's wizard or advanced mode). This script:

- reads the active environment.xml file and gathers a list of nodes upon which to act.
- installs the HPCC platform package(s) on all nodes specified.
- pushes out and deploys the environment file (environment.xml) to all nodes specified.
- optionally, if you specify the -k option it also generates the required ssh keys and deploys them as required to all nodes specified.
- optionally, if you specify the -p option it pushes out the existing ssh keys to all nodes specified. Use either the -k or the -p option, but not both.
- optionally, if you specify the -n <value> option it spawns that many concurrent executions. Default is 5.

Examples:

This example installs the HPCC Platform packages to remaining nodes and pushes out the active environment.xml file to those nodes.:

```
sudo /opt/HPCCSystems/sbin/install-cluster.sh hpccsystems-platform-xxxx-n.n.nnnn
```

(where *n.n.nnnn* is the build number)

This example installs the HPCC Platform packages to all nodes and pushes out the active environment.xml file to those nodes. It also generates ssh keys and pushes them out to all nodes.

```
sudo /opt/HPCCSystems/sbin/install-cluster.sh -k hpccsystems-platform-xxxx-n.n.nnnn
```

(where *n.n.nnnn* is the build number)

This example installs the HPCC Platform packages and pushes out the active environment.xml file to 8 concurrent nodes.:

```
sudo /opt/HPCCSystems/sbin/install-cluster.sh -n 8 hpccsystems-platform-xxxx-n.n.nnnn
```

(where *n.n.nnnn* is the build number)

deploy-java-files.sh

deploy-java-files.sh [-c] [-e] [-H <value>] [-n <value>] [-r] [-s <value>] [-t <value>] [-u <value>] [-x]

-c	When specified, this option adds the target directory or jar file path to classpath in environment.conf.
-e	When specified, this denotes the target is to be removed from the classpath.
-H	Host IP list. When specified, will target the IP addresses specified, one IP address per line. If this option is not used will run on the IP list generated from the environment.xml
-n	When specified, denotes the number of concurrent execution threads. Default is 5. You must have python installed, otherwise this option will be ignored and the action will run on each host sequentially.
-r	Reset classpath. When specified, will reset the classpath to <install_directory>/classes. If used in conjunction with the -t adds the new entries to the classpath after reset.
-s	Source file or directory.
-t	Target directory. The default is <install_directory>/classes. If it is only for adding to class-path, the value can be the full path of the java jar file.
-u	The username to use for ssh access to remote system. Provide this option when the specified user does not use a password to run ssh/scp. Without specifying this option you will be prompted to supply a username and password. We strongly recommend not using <hpcc user> to avoid security issues.
-x	When specified, this option excludes execution on the current host.

The **deploy-java-files.sh** script, is used to deploy java files (source) to HPCC cluster hosts and update the classpath variable in environment.conf.

This script runs a command on all IP addresses or host names in the active environment.xml. The IP addresses are defined when editing the environment in ConfigMgr.

This script writes to a log file:

/var/log/HPCCSystems/cluster/se_<action>_<commnd>_<pid>_yyyymmdd_HHMMSS.log

Examples:

To deploy java files from /home/hpcc/development/java/ on local system to /home/hpcc/java/ on all hosts in cluster and update classpath with 10 concurrent executions:

```
./deploy-java-files.sh -s /home/hpcc/development/java/* -t /home/hpcc/java/ -c -n 10
```

To deploy java files from /home/hpcc/java/ on local system to /home/hpcc/java on all hosts in cluster except local system:

```
./deploy-java-files.sh -s /home/hpcc/java/* -t /home/hpcc/java -x
```

To update classpath for a cluster:

```
./deploy-java-files.sh -c -t /home/hpcc/develop/java:/home/hpcc/test/java/
```

To To deploy java files to a list of hosts :

```
./deploy-java-files.sh -H /home/hpcc/hosts.txt -s /home/hpcc/java/* -t /home/hpcc/java/
```

hpcc-push.sh

hpcc-push.sh [-s <source>] [-t <target>] [-n <concurrent>] [-x]

-s	Source file or directory.
-t	Target file or directory.
-n, --concurrent	When specified, denotes the number of concurrent executions. Default is 5. Optional.
-x	When specified, this option excludes execution on the current host.

This script "pushes" files from the source filename and path to the destination filename and path for all IP addresses in the active environment.xml.

To use this script, the ssh keys need to be properly configured on all nodes, and you must use sudo.

The IP addresses were defined when editing the environment in ConfigMgr.

```
sudo /opt/HPCCSystems/sbin/hpcc-push.sh -s <sourcefile> -t <destinationfile>
```

For example:

```
sudo /opt/HPCCSystems/sbin/hpcc-push.sh -x \
     -s /etc/HPCCSystems/environment.xml -t /etc/HPCCSystems/environment.xml
```

hpcc-run.sh

hpcc-run.sh [-c component] [-a {hpcc-init|dafilesrv}] [-n <value>] [-s] [-S] {start|stop|restart|status|setup}

-a	HPCC service name. Either hpcc-init (default) of dafilesrv.
-c	HPCC component. For example, mydali, myroxie, mythor, etc.
-n, --concurrent	When specified, denotes the number of concurrent instances to run. The default is 5. Optional.
-S	When specified, the command runs sequentially, one host at a time.
-s	When specified, saves the result to a file named <ip address>.

To use this script, the ssh keys need to be properly configured on all nodes, and you must sudo as user hpcc.

This script runs a command on all IP addresses in the active environment.xml.

The IP addresses were defined when editing the environment in ConfigMgr. This script supports all the parameters of hpcc-init and dafilesrv.

Examples:

This example starts all components on the nodes

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init start
```

This example starts all components on all the nodes, using 8 concurrent executions

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init start -n 8
```

This example starts all components of the esp type on the nodes

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -c esp -a hpcc-init start
```

This example starts all components with a component name myesp on the nodes

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -c myesp -a hpcc-init start
```

This example starts the dafilesrv helper application

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a dafilesrv start
```

Uninstalling the HPCC Platform

To uninstall the HPCC platform, issue the appropriate commands for your system. If necessary, do so on each node that it is installed on.

Centos/Red Hat

```
sudo rpm -e hpccsystems-platform
```

Ubuntu/Debian

```
sudo dpkg -r hpccsystems-platform
```

Helper Applications

There is a helper applications that runs on all nodes that you may need to stop or start manually.

Normally, this process is started automatically the first time the hpcc-init service executes.

Enter the following commands to stop or start the helper application:

- dafilesrv

```
sudo /sbin/service dafilesrv stop

sudo /sbin/service dafilesrv start
```

hpcc-init

sbin/service hpcc-init [*option*] *command*

option

- -c *componentname*, --component=*componentname*

  Specifies the component upon which to execute the command. If omitted, the default is **all** components on the machine.

  -c *componenttype*, --component=*componenttype*

  Specifies the component type upon which to execute the command. If more than one of this type is configured, all will be acted upon. If omitted, the default is **all** components on the machine.

- --componentlist

  Provides a list of all component names on the current node as specified in the environment file.

- --typelist

  Provides a list of all component types on the current node as specified in the environment file.

- -h, --help

  Displays a help page

command

- start:

  Starts component(s)

- stop

  Stops component(s)

- status

  Displays component(s) status

- restart

  Restarts component(s)

- force-reload

  Deletes all local configuration files, data files, log files, and then restarts component(s). BE CAREFUL using this command.

- setup

  Initializes component configuration files but does not start the component(s).

The **hpcc-init** function is used to start, stop, restart, setup, or check the status of any or all HPCC components.

Examples:

```
sudo /sbin/service hpcc-init start
sudo /sbin/service hpcc-init stop

sudo /sbin/service hpcc-init -c myeclserver start
sudo /sbin/service hpcc-init --component=myeclserver start

sudo /sbin/service hpcc-init -c esp start
```

Unity Launcher Icon

The HPCC platform supports an Ubuntu Unity Launcher icon.

This allows you to start, stop, restart, or query the status of an installed single node system from an icon on the Unity Launcher of a desktop version of Ubuntu.

Note: This is only useful on a single-node system at this time. Future versions may operate in a different manner and support multi-node HPCC systems®.

To add the icon:

1. Use the search on Dash Home to find the HPCC Systems® application icon.

Figure 32. HPCC Application Icon

2. Click and Drag it to the Unity Launcher bar.

Figure 33. Unity Launcher

3. Drop it on the bar.

Note: In Ubuntu 12.04 or later, you can move the to any position on the bar by dragging and dropping to the desired position.

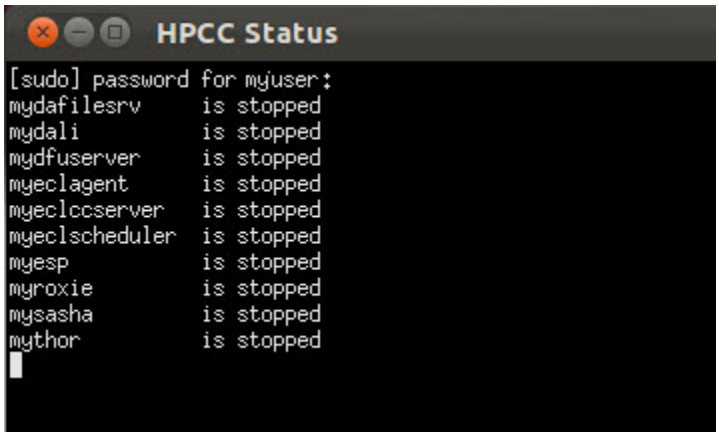
To use the icon:

1. Right-click on the icon, then select the desired action from the menu.

Figure 34. Context Menu

2. The result displays in a Terminal window.

Figure 35. Results



3. Close the window when you are done.

Running the ECL IDE under WINE

To run the ECL IDE under WINE in Linux, follow these steps.

1. Install wine1.2 (this corresponds to Wine version 1.1.31) and its dependencies.
2. Download msxml3.msi from Microsoft (Service Pack 7 or later).

   http://www.microsoft.com/en-us/download/details.aspx?id=3988
3. Install msxml3.msi in Wine (Double-click the msi file and Wine will install it).
4. Open Configure Wine (Applications/Wine/Configure Wine):
5. Select the Libraries tab.
6. In the New override for library drop list, select *msxml3*, then press the add button.
7. Select *msxml3* in the Existing overrides list and press Edit.
8. Select the *Native (Windows)* option and press the OK button.
9. Press the OK button to close the Wine Configuration window.
10. Install the HPCC ECL IDE (Double-click the setup.msi file and Wine will install it).

External Language Support

This section covers the steps to add external language support to the HPCC platform. HPCC offers support for several programming languages, some have additional dependencies that must be installed. External language support is included with the platform installation package, however there are RPM-based HPCC Platform installation packages that explicitly state **with plug-ins**.

RPM-based systems:

If you are interested in using external languages for RPM-based systems (CentOS/Red Hat), you need to download and install the appropriate platform installation distribution **with plug-ins** option from the downloads site.

For RPM based systems, there are two different installation packages available. One package includes the optional plug-ins to support embedded code from other languages. If you want support for other languages, choose the package for your distro that begins with:

```
hpccsystems-platform_community-with-plugins-
```

Debian-based systems:

Optional plug-in downloads are NOT needed for the Debian-based systems (Ubuntu) installation package, as the plug-ins are included in all the Debian installation packages.

The external languages currently supported include:

- C++ (full support is already built-in)
- Java
- JavaScript
- Python
- R

The following sections detail what is required to utilize these languages in your HPCC platform.

In addition to these languages, you can add support for additional languages by creating your own plug-in. This is not very difficult to do. For example the JavaScript plug-in is about 500 lines of C++ code. You can use that as a template to write your own and, if desired, you can contribute it back to the open source initiative.

Java

You can run external Java code on the HPCC platform. Compiled Java can be used either as a .class (or a .jar) and called from ECL just like any other ECL function.

To extract the JNI signatures:

```
javap -s
```

To set up Java to integrate with the HPCC platform:

1. Install a Java development package, such as OpenJDK or Oracle Java SE Development Kit (JDK) on the server.

2. Set the Java CLASSPATH

   You can set the classpath several ways:

   - In your profile.
   - In your environment.
   - in your JVM Profile.
   - using classpath value in environment.conf

     The default configuration file for the HPCC platform is **/etc/HPCCSystems/environment.conf** you will need to edit this file to point to your Java build directory.

     For example (on a Linux system):

     ```
     classpath=/opt/HPCCSystems/classes:/home/username/workspace/StreamAPI/bin
     ```

     The classpath should point to your Java build directory.

3. Start the HPCC Systems® platform (restart if it is already running) in order to read the new configuration.

   For example :

   ```
   sudo service hpcc-init start
   ```

   or

   ```
   sudo service hpcc-init restart
   ```

   For more information see the Starting-and-stopping the HPCC System in the *Installing and Running The HPCC Platform* document.

4. Test the Java integration.

   The HPCC Systems® platform comes with a Java example class. You can execute some Java code either in your ECL IDE or the ECL Playground.

   For example:

```
IMPORT java;

integer add1(integer val) := IMPORT(java, 'JavaCat.add1:(I)I');

add1(10);
```

   If this successfully executes, you have correctly set up Java to work with your HPCC platform.

   If you get a "unable to load libjvm.so" error you should reinstall or try a different Java package.

You can call Java from ECL just like any other ECL function. Java static functions can be easily prototyped using ECL types.

Additional examples of Java for HPCC can be found at:

https://github.com/hpcc-systems/HPCC-Platform/tree/master/testing/ecl/embedjava.ecl

JavaScript

To enable JavaScript support within the HPCC Systems® Platform:

1. Install the appropriate dependencies for your platform.

   RPM-based systems:

   JavaScript support is available for CentOS 6.x or later (not available for CentOS 5.x).

   On an RPM-based system (CentOS/Red Hat) install **v8embed.**

   Debian-based systems:

   For a Debian-based system (Ubuntu) install the **libv8-dev** package.

2. Test the JavaScript integration.

   JavaScript does multi-thread, as a result this can be the fastest of the currently supported embedded languages.

   You can now execute some JavaScript code either in your ECL IDE or the ECL Playground.

   For example:

```
//nothor

IMPORT javascript;

javascript.Language.syntaxcheck('1+2');

integer add1(integer val) := EMBED(javascript) val+1; ENDEMBED;

data testData(data val) := EMBED(javascript) val[0] = val[0] + 1; val; ENDEMBED;
set of integer testSet(set of integer val) := EMBED(javascript)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

add1(10);
```

   If this successfully executes, you have correctly set up JavaScript to work with your HPCC platform.

   Additional examples of HPCC code can be found at:

   https://github.com/hpcc-systems/HPCC-Platform/tree/master/testing/ecl/embedjava.ecl

Python

To enable Python support within the HPCC Systems® Platform:

1. Install Python, if not already installed. Many distributions come with Python already installed.

   Python 2.6 or 2.7 depending on your distribution's default version.

2. You can embed Python natively inside an ECL Program, much like BEGINC++

3. Call Python from ECL as you would any other ECL function.

   Python does not multi-thread efficiently (Global Interpreter Lock). Effectively only one thread can be in the python code at once. Scripts are compiled every call (but with caching of most recent, per thread). The IMPORT case will avoid recompiles.

4. Test the Python integration.

   You can now execute some Python code either in your ECL IDE or the ECL Playground.

   For example:

```
IMPORT Python;

SET OF STRING split_words(STRING val) := EMBED(Python)
  return val.split()
ENDEMBED;

split_words('Once upon a time');
```

   If this successfully executes, you have correctly set up Python to work with your HPCC platform. You can now embed Python anywhere you would use ECL within with your HPCC System.

   Additional examples of HPCC code can be found at:

   https://github.com/hpcc-systems/HPCC-Platform/tree/master/testing/ecl/embedjava.ecl

R

To enable R support within The HPCC Systems® Platform:

1. Install R-core distribution of your choice:

   RPM-based systems:

   On an RPM-based system (CentOS/Red Hat) use **R-core** and **R-core-devel**

   Debian-based systems:

   For a Debian-based (Ubuntu) system use **r-base-core**.

2. Install the **Rcpp** and the **RInside** libraries.

   You will need both Rcpp and RInside libraries in order for the R-embedding to work properly. The library installation packages are named with the version number appended so you should search for Rcpp_ and RInside_ to obtain the latest version for your system.

   For all distros except Ubuntu 14.04:

   ```
   wget http://cran.r-project.org/src/contrib/Rcpp_0.10.4.tar.gz
   wget http://cran.r-project.org/src/contrib/RInside_0.2.10.tar.gz
   ```

   To Install:

   ```
   sudo R CMD INSTALL Rcpp_0.10.4.tar.gz
   sudo R CMD INSTALL RInside_0.2.10.tar.gz
   ```

   For Ubuntu 14.04:

   ```
   wget http://cran.r-project.org/src/contrib/Rcpp_0.11.0.tar.gz
   ```

   To Install:

   ```
   sudo R CMD INSTALL Rcpp_0.11.0.tar.gz
   ```

   Note: The version of library files to install on your system must be 0.2.10 for all distros except Ubuntu 14.04. For Ubuntu 14.04, use 0.2.11.x.

   These libraries are maintained by the R project and can be found on the site http://cran.r-project.org/src/contrib/ along with more information. Use the version number appropriate for your system. If a version number is unsupported, you must use an earlier version.

3. Install the **RInside** library.

   You will also need RInside library in order for the R-embedding to work properly. The version must match the

   For all distros:

   ```
   wget http://cran.r-project.org/src/contrib/Archive/RInside_0.2.10.tar.gz
   ```

   To Install:

   ```
   sudo R CMD INSTALL RInside_0.2.10.tar.gz
   ```

   These libraries are maintained by the R project and can be found on the site http://cran.r-project.org/src/contrib/ and http://cran.r-project.org/src/contrib/archive/ along with more information.

4. Test the R integration.

   R is not multi-thread aware, so the plug-in has to wrap all calls to R for critical sections. Scripts are compiled with every call to R.

   You can now execute some R code either in your ECL IDE or the ECL Playground.

   For example:

```
IMPORT R;

integer add1(integer val) := EMBED(R)
val+1
ENDEMBED;

string cat(varstring what, string who) := EMBED(R)
paste(what,who)
ENDEMBED;

data testData(data val) := EMBED(R)
val[1] = val[2];
val;
ENDEMBED;

set of integer testSet(set of integer val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

set of unsigned2 testSet0(set of unsigned2 val) := EMBED(R)
sort(val);
ENDEMBED;

set of string testSet2(set of string val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

set of string testSet3(set of string8 val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;
```

```
set of varstring testSet4(set of varstring val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

set of varstring8 testSet5(set of varstring8 val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

set of boolean testSet6(set of boolean val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

set of real4 testSet7(set of real4 val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

set of real8 testSet8(set of real8 val) := EMBED(R)
t = val [1];
val[1] = val[2];
val[2] = t;
val;
ENDEMBED;

set of integer2 testSet9(set of integer2 val) := EMBED(R)
sort(val);
ENDEMBED;

add1(10);
cat('Hello', 'World');
testData(D'ab');
testSet([1,2,3]);
testSet0([30000,40000,50000]);
testSet2(['one','two','three']);
testSet3(['uno','dos','tre']);
testSet4(['un','deux','trois']);
testSet5(['ein','zwei','drei']);
testSet6([false,true,false,true]);
testSet7([1.1,2.2,3.3]);
testSet8([1.2,2.3,3.4]);
testSet9([-111,0,113]);

s1 :=DATASET(250000, TRANSFORM({ integer a }, SELF.a := add1(COUNTER)));
s2 :=DATASET(250000, TRANSFORM({ integer a }, SELF.a := add1(COUNTER/2)));
SUM(NOFOLD(s1 + s2), a);

s1b :=DATASET(250000, TRANSFORM({ integer a }, SELF.a := COUNTER+1));
s2b :=DATASET(250000, TRANSFORM({ integer a }, SELF.a := (COUNTER/2)+1));
SUM(NOFOLD(s1b + s2b), a);
```

If this successfully executes, you have correctly set up R to work with your HPCC platform.

Additional examples of HPCC code can be found at:

https://github.com/hpcc-systems/HPCC-Platform/tree/master/testing/ecl/embedjava.ecl