



ElectricAccelerator Help

Version 8.1

Electric Cloud, Inc.
35 South Market Street, Suite 100
San Jose, CA 95113
www.electric-cloud.com

ElectricAccelerator 8.1

Copyright © 2002–2015 Electric Cloud, Inc. All rights reserved.

Published 9/28/2015

Electric Cloud® believes the information in this publication is accurate as of its publication date. The information is subject to change without notice and does not represent a commitment from the vendor.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” ELECTRIC CLOUD, INCORPORATED MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any ELECTRIC CLOUD software described in this publication requires an applicable software license.

Copyright protection includes all forms and matters of copyrightable material and information now allowed by statutory or judicial law or hereinafter granted, including without limitation, material generated from software programs displayed on the screen such as icons and screen display appearance.

The software and/or databases described in this document are furnished under a license agreement or nondisclosure agreement. The software and/or databases may be used or copied only in accordance with terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement.

Trademarks

Electric Cloud, ElectricAccelerator, ElectricAccelerator Huddle, ElectricCommander, ElectricFlow, ElectricInsight, and Electric Make are registered trademarks or trademarks of Electric Cloud, Incorporated.

Electric Cloud products—ElectricAccelerator, ElectricAccelerator Huddle, ElectricCommander, ElectricFlow, ElectricInsight, and Electric Make—are commonly referred to by their “short names”—Accelerator, Huddle, Commander, Flow, Insight, and eMake—throughout various types of Electric Cloud product-specific documentation.

All other trademarks used herein are the property of their respective owners.

Contents

Welcome to ElectricAccelerator	1
About ElectricAccelerator	1
Electric Make (eMake)	1
Electric File System (EFS)	1
ElectricAccelerator Agent (Agent)	2
Cluster Manager	2
Electrify	2
Cluster Manager and Other Components	2
When to Use the ElectricAccelerator Cluster	2
Grid Integration	3
ElectricAccelerator Installed Tools/Files	5
Using Breakpoints	7
Using ecbreakpoint Applications	8
Using ecbreakpoint	8
Labeling Breakpoints	11
Using Agent-Side Breakpoints	13
Specifying a Breakpoint Through a File	15
Troubleshooting Breakpoints	17
Breakpoint Does Not Trigger	17
Breakpoint regex Is Wrong	17
Output Is Not as Expected	17
Use Your Editor to Run Breakpoints	17
Using Cluster Manager Administration Tools	18
Using clusterexec	18
Using clusterupload	21
Using clusterdownload	22
Sample Uses of Cluster Tools	24
Stopping, Starting, or Restarting the eRunner Daemon or Checking Its Status	26
Stopping, Starting, or Restarting the Daemon	26
Checking whether the Daemon is Running	26
Using ecconfig	27
Using ecagent Tools	28
Using ecinstconf	29
Using eclock	32
Using the runagent.local File	33

Preventing "runaway" Jobs	33
Configuring Stalled Job Detection Settings	33
Configuring Agent Log Rotation	34
System Maintenance Tasks	35
Cleaning Up Data	35
Deleting from the Builds List	35
Deleting Extraneous or Obsolete Builds	35
Managing the Message Log	35
Securing the Cluster Manager Database	35
Database Configuration	37
Configuring Accelerator to Use an External Database	37
Backing Up and Restoring a Database	37
Enabling LDAP	39
Import License	40
Importing a License	40
Using Comments	41
Adding a New Comment	41
Editing a Comment	41
Deleting a Comment	41
Stopping Builds	42
Stopping a Single Build from the ElectricAccelerator Web Interface	42
Stopping Multiple Builds from the Web Interface	42
Stopping a Build Using cmtool	42
Deleting Builds	43
Deleting a Single Build	43
Deleting a Group of Builds from the Build List	43
Creating Build Classes	44
Build Class Examples	44
Editing Build Classes	46
Deleting Build Classes	47
Deleting a Single Build Class	47
Deleting Multiple Build Classes	47
Agent Policies	48
Agent Allocation Policy	48
Preemption Policy	48
Agent Lock Interval (seconds)	49
Deep/Wide Agent Allocation Policy	49
Understanding a Cluster Sharing Policy	51
Modifying the Number of Agents per Host	52
Enabling Agents	53
Enabling a Single Agent	53
Enabling a Group of Agents	53
Disabling Agents	54
Disabling a Single Agent	54

Disabling a Group of Agents	54
Testing Agents	55
Checking the Status of a Single Agent	55
Checking the Status of a Group of Agents	55
Deleting Agents	56
Deleting a Single Agent	56
Deleting a Group of Agents from the Agent List	56
Removing an Agent from the Cluster Permanently	56
Creating Resources	57
Host Name and Glob-Style Pattern Example 1	57
Host Name and Glob-Style Pattern Example 2	57
Editing Resources	58
Deleting Resources	59
Deleting a Single Resource	59
Deleting Multiple Resources	59
Priority Pools	60
Using Priority Pools	60
Use Case 1—High Performance Builds	60
Use Case 2—Multiple Departments	61
Creating Filters	63
Editing Filters	64
Deleting Filters	65
Deleting a Single Filter	65
Deleting Multiple Filters	65
Generating Reports	66
Adding Custom Reports	67
Making Custom Reports Available to the Cluster Manager	67
Known Issues	67
The Home Page	68
The Top Bar	68
Server Information	68
Comments	68
Builds	69
Functionality	69
Column Descriptions	69
Build Details	71
Functionality	71
Details	71
Comments	71
Build Classes	72
About Build Classes	72
Functionality	72
Column Descriptions	72

Build Class Details	74
Functionality	74
Details	74
Comments	74
Build Classes- Create or Edit a Build Class	75
Field Descriptions	75
Agents	78
Functionality	78
Column Descriptions	78
Agent Details	80
Functionality	80
Status	80
Configuration	80
Agent Console	80
Comments	80
Agent Policies	81
Agent Allocation Policy	81
Preemption Policy	81
Agent Lock Interval (seconds)	82
Deep/Wide Agent Allocation Policy	82
Resources	83
Functionality	83
Column Descriptions	83
Resource Details	84
Functionality	84
Details	84
Comments	84
Resources - Create or Edit a Resource	85
Field Descriptions	85
Reports	86
Checking Your Permissions to Run Reports	86
Using Permanent Links to Reports	87
Downloading Report Data	87
Agent Usage Report	87
Build Duration Report	88
Build Duration vs. Conflict Report	93
Build Duration vs. Job Count Report	94
Build Summary Report	95
Build Wait Time Report	96
Cluster Usage Report	100
Cluster Utilization Report	101
Concurrent Builds Report	103
Effective Agent Allocation Summary Report	106
Effective Agent Allocation Per Day Report	108
JobCache License Allocation Report	109

JobCache Usage Summary Report	114
Resource Usage over Time Report	115
Sea Level Report	116
Server Load	117
About Server Load	117
Functionality	117
Column Descriptions	117
Realtime Server Metrics	119
Messages	120
Column Descriptions	120
Using the Custom SQL Filter	120
Message Policies	122
Administration	123
Filters	124
Filter Panel	124
Functionality	124
Column Descriptions	125
Filters - Create or Edit a Filter	126
Field Descriptions	126
Permissions	127
Functionality	127
Column Descriptions	127
Edit User Permissions	130
Enable User	131
Edit Group Permissions	132
Enable Group	133
Users	134
Functionality	134
Column Descriptions	134
User Details	134
Users - Create or Edit Local Users	135
Creating a User	135
Field Descriptions	135
Editing a User	135
User Password	136
Groups	137
Functionality	137
Column Descriptions	137
Group Details	137
Group Members	137
Groups - Create or Edit a Local Group	138
Group Members	138
Creating a Group	138

Editing a Group	138
Removing Users From a Group	138
Add New User to Group	139
Adding a User to a Group	139
User Settings	140
Editing User Settings	140
Field Descriptions	140
Server Settings	141
Field Descriptions	141
About Resource Managers	142
Database Configuration	143
Configuring Accelerator to Use an External Database	143
Backing Up and Restoring a Database	143
Licenses	145
Column Descriptions	145
Import License	146
Importing a License	146
Cloud	147
CloudHostManager Information	147
Current Jobs	147
Error Codes	148

Welcome to ElectricAccelerator

About ElectricAccelerator

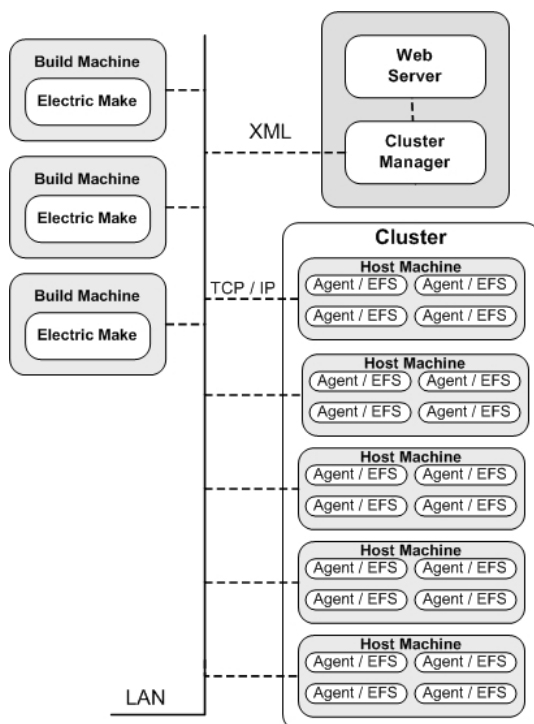
ElectricAccelerator® ("Accelerator") is a software build accelerator that dramatically reduces software build times by distributing the build over a large cluster of inexpensive servers. Using a patented dependency management system, ElectricAccelerator identifies and fixes problems in real time that would break traditional parallel builds. ElectricAccelerator plugs into existing software development environments seamlessly and includes web-based management and reporting tools.

ElectricAccelerator components:

- Electric Make® (eMake)
- Electric File System (EFS)
- ElectricAccelerator Agents (Electric Agent or Agent)
- Cluster Manager
- Electrify

Electric Make (eMake)

Electric Make, the main build application, is a new Make version invoked interactively or through build scripts. It reads makefiles in several different formats, including GNU Make and Microsoft NMAKE. Electric Make distributes commands to the cluster for remote execution and services file requests.



Electric File System (EFS)

Electric File System (EFS) is a special-purpose file system driver, monitoring every file access on the hosts and providing Electric Make with complete usage information. This driver collects dependency information, allowing

Electric Make to automatically detect and correct out-of-order build steps. Each EFS driver instance is paired with an ElectricAccelerator Agent. During the ElectricAccelerator installation process, the Agent and EFS are installed at the same time.

ElectricAccelerator Agent (Agent)

As the user-level component running on the hosts, the Agent and EFS are inseparable—the Agent is an intermediary between Electric Make and EFS. Depending on your system configuration requirements, you may have one EFS/Agent installed per virtual CPU.

Cluster Manager

The Cluster Manager machine is a server that contains two layers:

1. A service layer to allocate Agents to builds, collect build results, and monitor system activity, and
2. A web server layer for the web interface.

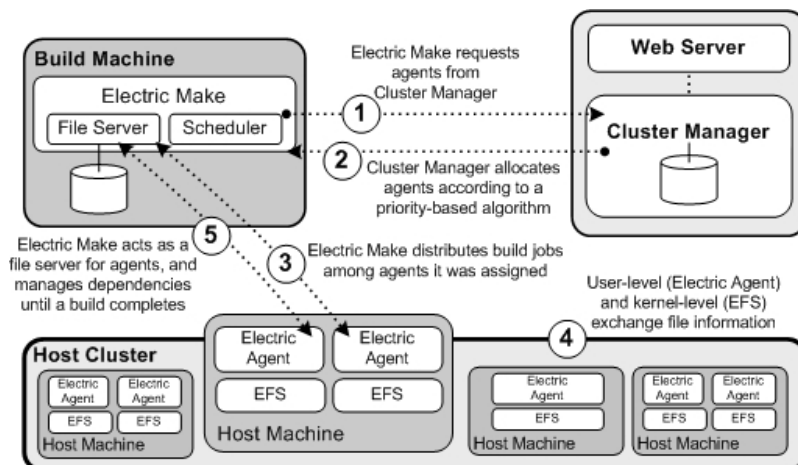
Cluster Manager also allows users to view upcoming and past builds, view the status of Agents in the cluster, and run reports.

Electrify

The Electrify component accelerates builds by parallelizing the build process and distributing build steps across clustered resources. You can use one of two methods to monitor your tools: a wrapper application or automatic process interception. The *ElectricAccelerator Electric Make Users Guide* contains information about using Electrify.

Cluster Manager and Other Components

Cluster Manager's primary responsibility is Agent allocation. However, through Cluster Manager, users also create and maintain the build infrastructure: build classes, user access, build history details, and manage Agent activity.



When to Use the ElectricAccelerator Cluster

ElectricAccelerator is designed to be a complete replacement for GNU Make or Microsoft NMAKE. It has the same command-line switches, produces identical output, and (with a few exceptions) reads your existing makefiles without change. Wherever you are currently invoking `gmake` or `nmake`—in a script, from a batch process, or interactively—you can run Electric Make to distribute work to the Cluster Manager and complete your build faster.

In some cases, however, distributed builds are not always faster. In particular, if a build has very little work to do and/or has much higher I/O activity compared to computation, it can be faster to allow the build to execute locally. The best examples of these build types are:

- "Clean" targets—often do little more than large recursive deletes.
- "One touch" build—a build consisting of nothing more than zero or one compilations followed immediately by a large link.

But then, depending on the build specifics, the cluster may outperform local builds for a large class of "clean" target or "one touch" builds—it is best to experiment with a few common configurations to determine which mode is more efficient. Typically, if a build has two or more jobs it can execute concurrently, running the build on the cluster is more efficient.

If you find that some common "clean" or "one touch" builds are faster when run locally, you can run Electric Make in local mode so it will behave exactly like your existing Make, running all jobs in series on the local system.

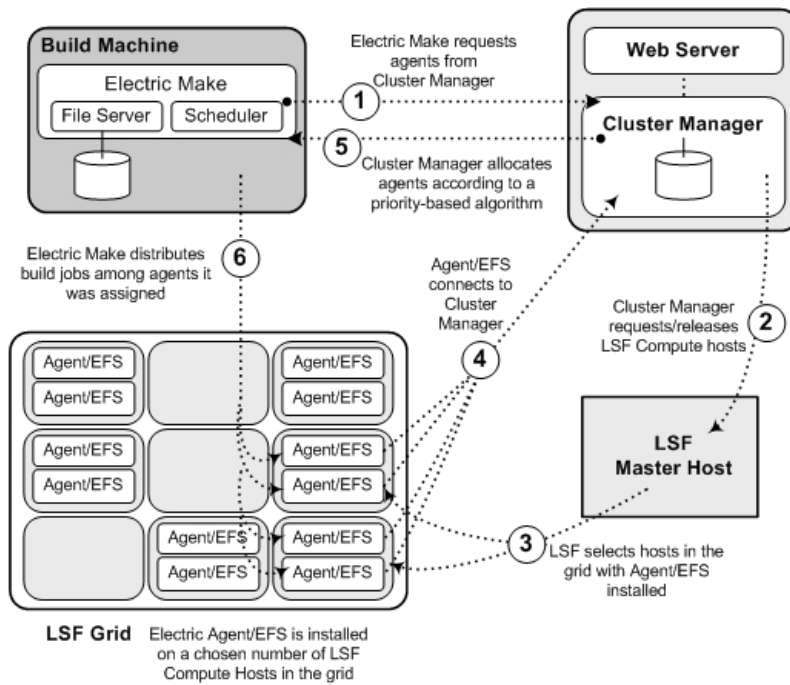
Grid Integration

If you plan to use a host grid with ElectricAccelerator, the following list shows the relationship between ElectricAccelerator components and the grid management software, such as LSF or Oracle Grid Engine.

- Electric Make (eMake) does not interact directly with grid management software—Electric Make can run inside or outside of a grid.
- Electric Agent/EFS software is designed to run on grid execution hosts. While the Agent/EFS does not "talk" to grid management software, the Agent/EFS can be activated by it, effectively creating a dynamically sized ElectricAccelerator cluster.
- Cluster Manager interfaces with the grid management software to add or subtract machines to/from the ElectricAccelerator cluster as build demands increase or decrease.

Note: Grid management client software must be installed on the designated Cluster Manager server *before* Cluster Manager is used in grid mode.

The following diagram illustrates how ElectricAccelerator interacts with grid management software, using LSF as an example.



ElectricAccelerator Installed Tools/Files

The following table is an overview of useful tools installed during the ElectricAccelerator product installation. Table columns are defined as follows:

- **File**—Lists the tool filenames in alphabetical order
- **Location**—Shows where the files are installed
For example: "cm" is the Cluster Manager machine.
- **Description**—Provides a short description or reference

File	Location	Description
clusterdownload	cm, agent, emake	Downloads files to the host. See Using Cluster Manager Administration Tools .
clusterexec	cm, agent, emake	Runs commands on Agent hosts. See Using Cluster Manager Administration Tools .
clusterupload	cm, agent, emake	Uploads files to Agent hosts. See Using Cluster Manager Administration Tools .
cmtool	cm, agent, emake	Command-line tool for Cluster Manager.
ecagent_restart /etc/init.d/ecagent restart	agent	Restarts the agent service or daemon. See Using ecagent Tools .
ecagent_start /etc/init.d/ecagent start	agent	Starts the agent service or daemon. See Using ecagent Tools .
ecagent_stop /etc/init.d/ecagent stop	agent	Stops the agent service or daemon. See Using ecagent Tools .
ecbreakpoint	agent	Debugging tool for problematic job steps and stalled processes during the build.
eccc	cm, agent, emake	A wrapper for SunForte Compiler Suite and SunOne Studio/Workshop if you are experiencing excessive serialization in your build.
ecconfig	agent	Reconfigures Agents. See Using ecconfig .
ecinstconf	cm, agent	Configures Cluster Manager and Agent settings. See Using ecinstconf .

File	Location	Description
eclock	cm, emake	Serializes multiple runnings of a program. For example, it may not be safe to run two copies of a particular application to run simultaneously on a single computer, even if it is run by different users. Use eclock (typically in makefiles) to cause various invocations of that program to wait for each other. See Using eclock .
emake	cm, agent, emake	ElectricAccelerator's Electric Make component. See the <i>ElectricAccelerator Electric Make Users Guide</i> , "Electric Make Command-line Options and Environment Variables."

Using Breakpoints

You can use breakpoints to help debug hard-to-isolate build issues. Two types of breakpoints exist:

- the [ecbreakpoint application](#) used within makefiles
- [agent-side breakpoints](#) written in Tcl that are triggered by the agent

The `ecbreakpoint` program is a client-server application written in C. It can be called from within makefile rule bodies, shell scripts, or from agent-side breakpoints. When `ecbreakpoint` runs, it sends an "I'm waiting" message to the Cluster Manager over an HTTP channel and creates a socket and sits in a loop waiting for shell or agent commands. In addition, you can interact with `ecbreakpoint` using `curl` or `wget`. It is important to note that only the job is stopped, not the entire build. Other jobs continue to be scheduled and run, including jobs on other agents on the same cluster host.

The agent-side breakpoint feature is more difficult to configure and debug, but it enables a higher level of precision when trying to find problems. The agent-side breakpoint enables agents to run arbitrary commands when specific conditions are encountered. Typically, the Tcl breakpoint code calls `ecbreakpoint` when the `stdout` from a job matches a particular string. However, any command can be run, including shell scripts and tools such as `lsof` and `strace/truss` on UNIX and `procmon` on Windows.

After `ecbreakpoint` stops a job, you can use the Cluster Manager Break Point Details page to send arbitrary shell and session commands to the agent. In addition, you can rerun or continue the job. You must have full breakpoint permissions to use all breakpoint operations. If you cannot see breakpoints, you must change your permissions. Go to Administration > Permissions > Edit Permissions (for your account) > Break Points: Full

By stopping jobs, you can query the agents or operating system in the context that a job ran. Scenarios where you might want to use the breakpoint feature:

- files are not available when they should be
- permission problems
- perform checksums on files
- examine an agent's state

For help troubleshooting breakpoints, see [Troubleshooting Breakpoints](#).

Using ecbreakpoint Applications

The `ecbreakpoint` executable (`ecbreakpoint/ecbreakpoint.exe`) resides in the agent machine's `ecloud` bin directory (for example, `/opt/ecloud/i686/bin/` on Linux, and `c:\ecloud\i686_win32\bin\` on Windows). Table 1 shows `ecbreakpoint` command line arguments.

Usage: <code>ecbreakpoint</code> [switches]	
<code>--cm=<value></code>	The name of the Cluster Manager to connect to.
<code>-d, --debug</code>	Enable debug mode.
<code>--dontConnectToCM</code>	Do not connect to the Cluster Manager.
<code>-h, --help</code>	This usage message.
<code>--label=<value></code>	The label for the break point. This is used in the web UI to identify the break point.
<code>--port=<value></code>	The HTTP port to use to contact this application.
<code>--printLabel</code>	Print the label when the break point is reached.
<code>--timeout=<value></code>	The number of seconds after which the break point continues. Default is 10 days.
<code>-v, --version</code>	Print version information.

Table 1: `ecbreakpoint` command-line options

Using ecbreakpoint

You can add `ecbreakpoint` to makefile rule bodies as in Figure 1. Emake pauses the job when it encounters the breakpoint. The Cluster Manager then displays the break point panel on the Build Details page (Figure 3). Click a breakpoint action to display the Break Point Detail page.

```
all:  a b
    @ echo Running $$

a:
    @ echo Before breakpoint
    ecbreakpoint

b:
    @ echo Some other job
```

Figure 1: Makefile using the `ecbreakpoint` program

Figure 2 illustrates a build (`chronic_641`) with jobs paused by `ecbreakpoint`.



	chronic_ecloudmain.12505-200701091453
	chronic_641
	chronic_642
	chronic_641
	chronic_642
	chronic_ecloudmain.100000001-200701091454

Figure 2: Paused job



Break Points					
Label	Agent Name	IP Address	Port	Created	Actions
	support-lin3-4	192.168.26.235	41184	2009-01-10T08:40:17.000	Retry Continue Agent Command Shell Command

Figure 3: Breakpoint UI on the Build Details page

If you have [sufficient permissions](#), you can interact with the `ecbreakpoint` program within the web interface. There are currently four actions available for `ecbreakpoint`:

- **Retry**—Redo this job step from the beginning
- **Continue**—Step past the breakpoint
- **Agent Command**—Issue agent commands, such as "session state", to the stopped agent
- **Shell Command**—Issue shell commands to the stopped agent. These commands run in the same context (filesystem, environment) as the job itself. For UNIX, `bash` shell is used to execute commands; for Windows, `cmd.exe` is used. Therefore, ";" is not allowed in the `ecbreakpoint` shell command for Windows.

Click **Retry** or **Continue** to delete the current `ecbreakpoint`.

Click **Agent Command** or **Shell Command** to go to the Break Point Details page (illustrated in Figure 4), which provides more information and interaction with the breakpoint. After you issue a command, the response is shown as "Waiting for response...". Refresh the page after a few seconds to view the result (the lower portion of Figure 4). The value of the request Type column can be S or A, which is shell or agent command respectively.

Figure 4 illustrates that `ps tree -alp` was passed to the agent. The `ps tree` command shows a hierarchical representation of the process tree and is often useful to see what commands agents are running. Note that `ps tree` is a child of `ecbreakpoint` and `ecbreakpoint` is a child of agent 0. The compilation and linking processes will probably be finished by the time you run `ps tree`. You may see other processes, for example, processes launched in the background such as license managers, that may provide clues about what is happening.

Break Point Details – 7

Details

Build ID: 185
Break Point Label:
Port: 41412
Agent Name: support-lin3-2
Agent IP Address: 192.168.26.235
Create Time: 2009-01-10 10:08:41

Command:

Agent Command
Shell Command

Type	Request	Response	Requested By	Created Time
S	pstree -alp	<pre> init,1 -acpid,2214 -agent,2324 /opt/eccloud/i686_Linux/bin/runagent -cm support-lin1:8030 -efsid 0 -numagents 4 -secureconsole n-d -execserver,1231 -sleep,1236 60 -agent,2341 /opt/eccloud/i686_Linux/bin/runagent -cm support-lin1:8030 -efsid 1 -numagents 4 -secureconsole n-d -execserver,1225 -ecbreakpoint,1229 -pstree,1249 -alp -agent,2358 /opt/eccloud/i686_Linux/bin/runagent -cm support-lin1:8030 -efsid 2 -numagents 4 -secureconsole n-d -execserver,1237 -sleep,1241 60 -agent,2375 /opt/eccloud/i686_Linux/bin/runagent -cm support-lin1:8030 -efsid 3 -numagents 4 -secureconsole n-d -execserver,31813 -sleep,1222 60 </pre>	admin	2009-01-10 10:09:03

Figure 4: pstree command run from the breakpoint menu

You can also get session state information to send to Electric Cloud technical support.

Type	Request	Response	Requested By	Created Time
A	session state	<pre> Executing job Seconds in current state: 151.73 Last request sent to emake: A2E_GET_ALL_VERSIONS Last request received from emake: E2A_RUN_COMMAND_EX Current build id: 244 52 emake requests, 52 EFS requests processed Last command executed: cat /this/is/a/really/long/string/that/doesn't/exist/on/the/computer </pre>	admin	2009-01-14 15:56:32

Figure 5: Session state information

You can access `ecbreakpoint` from the command-line or scripts using the `curl` command. Accelerator ships with `curl` in the `eccloud bin` directory. Figure 3 shows the IP address and port for `ecbreakpoint`. Use those values in `curl` and make sure to use the appropriate escape characters for spaces, slashes, and so on.

Example for running the `ps` command on the agent:

```
http://192.168.26.235:41184/exec.xml?command=%2Fbi2Fps%20-ef
```

Table 2 contains URLs you can use to access `ecbreakpoint`.

<code>retry.xml</code>	Rerun the command on a different agent.
<code>continue.xml</code>	Continue normal execution.

exec.xml	Execute the specified command. The command is one simple command passed with the command GET parameter.
agent.xml	Execute the specified agent command. The command is one simple command passed with the command GET parameter.
agentstat.xml	Show the current agent session state.

Table 2:ecbreakpoint access URLs

Labeling Breakpoints

The Makefile in Figure 1 contained one `ecbreakpoint` command. If you want to set multiple breakpoints in your build, you must use the `ecbreakpoint --label` command line option to distinguish them in the UI. Notice the `--label` option in the Makefile in Figure 6.

```
all:  a b c
      @ echo Running $@

a:
      @ ecbreakpoint --label $@

b:
      @ ecbreakpoint --label $@

c:
      @ ecbreakpoint --label $@
```

Figure 6: Makefile with a label for each target

When the Makefile in Figure 6 runs, the Cluster Manager displays labels for each breakpoint (Figure 7).

Break Points								
Label	Agent Name	IP Address	Port	Created	Actions			
a	support-lin3-4	192.168.26.235	44173	2009-01-11T08:17:57.000	Retry	Continue	Agent Command	Shell Command
b	support-lin3-2	192.168.26.235	44178	2009-01-11T08:17:58.000	Retry	Continue	Agent Command	Shell Command
c	support-lin3-3	192.168.26.235	44181	2009-01-11T08:17:58.000	Retry	Continue	Agent Command	Shell Command

Figure 7: Labeled breakpoints on the Build Details page

The label command may need more information than just the target name. For example, if a pattern rule was called multiple times from various submakes where the C files had the same names, this could result in a situation similar to the one in Figure 8 where the labels are the same.

Break Points								
Label	Agent Name	IP Address	Port	Created	Actions			
a.c	support-lin3-4	192.168.26.235	44326	2009-01-11T09:02:36.000	Retry	Continue	Agent Command	Shell Command
a.c	support-lin3-3	192.168.26.235	44331	2009-01-11T09:02:37.000	Retry	Continue	Agent Command	Shell Command

Figure 8: Breakpoint labels with the same name

You can make labels more distinct by specifying the label as `--label=${(})_`pwd`` which results in the output in Figure 10.

```
%c:
@ touch $@
@ ecbreakpoint --label=" ${(})_`pwd`"
```

Figure 9: ecbreakpoint label option augmented with the working directory

Break Points								
Label	Agent Name	IP Address	Port	Created	Actions			
a.c_/u01 /ak /work/breakpoints /pattern/proj_a	support-lin3-4	192.168.26.235	44348	2009-01-11T09:08:53.000	Retry	Continue	Agent Command	Shell Commma
a.c_/u01 /ak /work/breakpoints /pattern/proj_b	support-lin3-3	192.168.26.235	44353	2009-01-11T09:08:54.000	Retry	Continue	Agent Command	Shell Commma

Figure 10: Breakpoint labels with the working directory

Using Agent-Side Breakpoints

You can use agent-side breakpoints when:

- you do not know where in the makefile a problem occurs, but you want to stop the build based on some condition
- you know, after the fact, where a problem occurs in a makefile, but the location keeps changing

You can use agent-side breakpoints to configure agents to run arbitrary commands when a particular condition is found. The arbitrary command can be `ecbreakpoint`, but it does not have to be. You can use the `cmttool runAgentCmd` program to configure agent-side breakpoints (Figure 1).

```
cmttool runAgentCmd 'session breakpoint set {trigger_code} {command_to_execute}'
cmttool runAgentCmd 'session breakpoint get'
cmttool runAgentCmd 'session breakpoint clear'
```

Figure 1: Session commands to manipulate agent-side breakpoints

Agent-side breakpoints enable agents to run a block of Tcl code before and after running the job command. The breakpoint consists of a block of trigger code and a command to run if the trigger code returns a non-zero return code. If the trigger block results in a non-zero exit code, the agent runs the command in the second block. The trigger code has access to the read-only breakpoint associative array that contains the entries in Table 1. Generally, the Tcl code you write will use regular expressions to match the stdout element of the breakpoint array. By default, eMake runs with the "merge streams" option so there is usually no reason to look in stderr.

Array	Contents
<code>breakpoint (command)</code>	The command-line
<code>breakpoint (location)</code>	"pre" or "post", for before or after the command runs
<code>breakpoint (exitcode)</code>	Exit code of the command; -1 for "pre" checks
<code>breakpoint (stdout)</code>	Output of the command; empty for "pre" checks
<code>breakpoint (stderr)</code>	stderr of the command. Generally, mergestreams is enabled, so you can look at stdout.

Table 1: Breakpoint associative array

Figure 2 illustrates a breakpoint that triggers if the standard output of a command contains the text `parse error` or `undeclared identifier`.

```
# Set breakpoint using runAgentCmd
cmtool --cm=support-lin1 runAgentCmd 'session breakpoint set {
    # examine stdout from breakpoint array to see if we have the error messages
    # we are looking for
    if { [ regexp {(parse error)|(No such file)} $breakpoint (stdout)] } {
        # Send info about what we find to the agent logs
        puts "Triggering breakpoint:"
        puts " \t Command: $breakpoint(command) "
        puts " \t Stdout: $breakpoint(stdout) "
        return 1
    }
    return 0
} {ecbreakpoint --cm=support-lin1}' --agentName support-lin3-1
```

Figure 2: Agent breakpoint triggered by regex on stdout

When you set the breakpoint and run a makefile that emits the desired error strings (Figure 3), you see the breakpoints displayed on the Build Details page (Figure 4).

```
all: a b c
@ echo Running $$
a:
@ echo "good compilation"
b:
cat /this/is/a/really/long/string/that/doesnt/exist/on/the/computer
c:
@ echo "parse error"
```

Figure 3: Makefile to generate error

Using the Agent Name displayed in the Break Points panel (Figure 4), you can find the output from the agent breakpoint in the agent logs (/var/log/ecagent#.logs) (Figure 5).

Break Points								
Label	Agent Name	IP Address	Port	Created	Actions			
	support-lin3-4	192.168.26.235	44881	2009-01-11T10:45:56.000	Retry	Continue	Agent Command	Shell Command
	support-lin3-3	192.168.26.235	44885	2009-01-11T10:45:56.000	Retry	Continue	Agent Command	Shell Command

Figure 4: Cluster Manager UI for agent-side breakpoints

```
Triggering breakpoint:
    Command: /bin/sh -c echo "parse error"
    Stdout: parse error
```

Figure 5: /var/log/ecagent4.log

Figure 4 illustrates the empty Label column. You cannot use the `ecbreakpoint --label` argument in the Figure 2 breakpoint because the breakpoint is set before jobs are run. Remember that the agent-side breakpoint can run arbitrary commands. The breakpoint code in Figure 6 accesses the command-line string from the breakpoint array, constructs a shell script that calls `ecbreakpoint` with the `--label` you create, and then runs the shell script when the breakpoint triggers.

```

cmtool --cm=support-lin1 runAgentCmd      'session breakpoint set  {
    # Examine stdout for error messages that we want to stop the build when
    encountered
    if { [ regexp    {(No such file )|(undeclared identifier)} $breakpoint
(stdout)] } {
        # cleanup the command line so that we can use it for a label.
        # remove text that was prepended by the agent and change spaces to
underscores
        regsub {^/bin/sh -c } $::breakpoint (command) {} cmd
        regsub -all {\s+} $cmd    "_"    cmd
        # If we have a long cmd, just use the first 20 and last 20
characters
        # for the label
        if {[ string    length $cmd ] > 40 } {
            regexp {^(.{ 20 }).*({ 20 })$} $cmd    match cmd_a cmd_b
            set cmd "${cmd_b}=${cmd_e}"
        }
        # Create a file so that we can write out the new ecbreakpoint
command with the
        # appropriate label for this job. By using [efs use], we can create
a unique
        # temp file for each agent
        set f [ open    /tmp/breakpoint_[efs use].sh w]
        puts $f    "ecbreakpoint --cm=support-lin1 --label $cmd"
        close $f
        return 1
    }
    return 0
} [ list    /bin/sh /tmp/breakpoint_[efs use].sh] '

```

Figure 6: Agent breakpoint that initiates an arbitrary program

When you run the makefile in Figure 3 with the new breakpoint code, more descriptive labels are printed in the Break Points panel (Figure 7).

Break Points							
Label	Agent Name	IP Address	Port	Created	Actions		
echo_parse error	support-lin3-1	192.168.26.235	45224	2009-01-11T12:12:45.000	Retry	Continue	Agent Command Shell Comm
echo_undeclared Identifier	support-lin3-3	192.168.26.235	45228	2009-01-11T12:12:45.000	Retry	Continue	Agent Command Shell Comm

Figure 7: Breakpoint list on the Build Details page

Note: Because the job path is not available to the agents, the example uses the command name as an identifier even though it is a poor substitute for the job path and might not always be unique. The jobid is available, but it does not correlate to the annotation file at the point in time when the breakpoint triggers because job details are not written to the annotation file until termination time.

Specifying a Breakpoint Through a File

You can also specify the breakpoint by passing in a file. Begin the command argument with "file: *yourpath*" and the file will be read from the path specified for the runAgentCmd request.

Example breakpoint contained in a file named test.txt

```
session breakpoint set {  
  if { [regexp {(parse error)|(No such file)} $breakpoint(stdout)]  
  } {  
    puts "Triggering breakpoint:"  
    puts "\t Command: $breakpoint(command) "  
    return 1  
  }  
  return 0  
} {ecbreakpoint --cm=linbuild-cm}
```

Example of passing in the breakpoint contained in test.txt

```
cmtool runAgentCmd "file:C:\tmp\test.txt" --agentId 20
```

Troubleshooting Breakpoints

After creating an agent-side breakpoint, you may find that it does not work as you expect or you do not see any output. The following are some possible problems and solutions:

Breakpoint Does Not Trigger

If the breakpoint does not trigger when you think it should, look in the agent log files. If there are Tcl errors with your breakpoint, you may see something like the following:

```
[228/0] 0.081472/105997992.238145 SYSTEM_LOG: INFO Error evaluating breakpoint condition: can't read "agentId": no such variable
Tcl errors will prevent the breakpoint from running.
```

Breakpoint regex Is Wrong

Be sure that the text you are looking for actually appears in the stdout. If the text is in the output, open the tkcon console (in the ecloud bin directory) and test your regex code. Modify the regex until it works.

Output Is Not as Expected

Output from agent-side breakpoint 'puts' statements is written to agent logs and not the build's stdout. If the agent breakpoint stops the build with `ecbreakpoint`, you can use the Break Points panel to determine which agent log to examine. However, if your breakpoint calls a tool such as `strace`, you may have to examine all agent logs. On Linux, one useful hint to remember is that Gnu `tail(1)` can tail multiple files at the same time (`tail -f /var/log/ecagent?.log`).

Use Your Editor to Run Breakpoints

When creating agent breakpoints, it is useful to have an open editor window with the `cmtool runAgentCommand set breakpoint` command that you are modifying. When working with the breakpoint, you can execute it from within the editor by selecting the code in Vim's visual mode and then using the 'ex' command line to feed it into the shell for execution. You do not need to clear the breakpoint before setting a new one because setting a breakpoint overwrites the current one. You can also cut-and-paste the breakpoint code into a command window.

In the `.vimrc`, configure the `makeprg` variable to call `eMake` instead of GNU Make; type `:make` from the ex command line to invoke the Makefile.

```
set makeprg=/opt/ecloud/i686_Linux/bin/emake\ --emake-cm=support-lin1
```

In emacs, you can feed text to the shell using 'M-|'. Select the region of the buffer you want to send to the shell, and then type M-|.

Using Cluster Manager Administration Tools

To manage and administer an ElectricAccelerator host cluster, three tools are available to perform operations across all hosts simultaneously. The tools, `clusterexec`, `clusterupload`, and `clusterdownload` are platform-independent tools and part of the eRunner package. By default, the eRunner daemon (UNIX) and eRunner service (Windows) are installed on agent machines and eMake machines. The tools are installed on Cluster Manager machines and eMake machines. Only the server side (`erunnerd`) is installed on agents to allow them to serve requests from tools. If you decided *not* to install eRunner during Cluster Manager installation, you do not have access to these tools. These Cluster Manager administration tools allow you to:

- Start and stop agents
- Reboot hosts
- Run commands on hosts
- Upload files
- Download files

These tools are particularly useful for automating the cluster upgrade process to update the build environment (for example, build tools, compiler, header files, system libraries, and so on) and for collecting debug information. Other uses include restarting Agents, restarting a subset of Agents, or obtaining host status information.

Specific command-line tool functions are:

- `clusterexec`—Allows you to run arbitrary commands on a host machine or all hosts in a cluster
- `clusterupload`—Allows you to upload executables and files (for example, compiler, libraries, and so on) to all hosts in a cluster
- `clusterdownload`—Allows you to download files from host machines to a central location

`clusterupload`, `clusterexec`, and `clusterdownload` communicate with the eRunner service on Windows hosts or the eRunner daemon on UNIX machines. The eRunner service/daemon listens for connections on port 2411 by default.

Because these tools are platform-neutral, `clusterexec` running on Windows can communicate with eRunner on Linux hosts so those hosts can execute the commands. Conversely, use `clusterupload` on Linux to upload files to a Windows host. However, some rules should be followed when using these tools cross-platform *from* UNIX to Windows. For more detailed information, see the next section, "Using `clusterexec`."

IMPORTANT: `clusterupload` and `clusterdownload` do not support ACLs. During upload/download, they are ignored. An effect of ignoring ACLs is that Cygwin symlinks are not handled properly.

Using `clusterexec`

This command executes shell commands given by `cmd1`, `cmd2`, and so on, on one or more hosts—typically, all hosts in a cluster. The syntax for running `clusterexec` is:

```
% clusterexec [options] "cmd1 arg2; cmd2; ..."
```

For example, if you are running Linux and want to find out how long the operating system on each host in the cluster has been running since the last reboot, you could run the `uptime` commands on each host in the cluster, `lin-cluster`. Using `clusterexec`, the syntax would be:

```
% clusterexec --cm=lin-cluster uptime
```

Each command may be the following type:

- A fully-qualified path to an executable on the hosts
- The executable name on the hosts. The eRunner service that processes the `clusterexec` command does a `PATH` search based on the following:
 - For Linux and Solaris, the `PATH` that `erunner` searches is defined in `/etc/init.d/erunner` and includes the well-known binary and system binary installations on the host
 - For Windows, the `path` that `erunner` searches is the `PATH` system environment variable on the host
- A built-in command defined in the eRunner service-see the "eRunner Built-in Commands" table [below] for a description of these commands

Some platform-specific rules to observe:

- `clusterexec` commands run as the `root` user on Solaris and Linux, or as the `LocalSystem` user on Windows
- When using `clusterexec` on Solaris/Linux to run a command on a Windows host, the shell may alter the command, for example, the UNIX shell treats "\" characters as escapes. This issue may apply also if the user is running in a UNIX-like shell on Windows (for example, `cygwin bash`). To correct this situation, use one of the following methods:
 - Use single quotes instead of double quotes, for example:
`clusterexec --hosts=win1 'c:\\winnt\\system32\\xcopy c:\\a.txt`
 - Use double-escape backslashes, for example:
`clusterexec --hosts=win1 "c:\\\\winnt\\\\system32\\\\xcopy`
 - Use forward slashes "/" instead of backslashes where possible, for example:
`clusterexec --hosts=win1 "c:/winnt/system32/xcopy c:/a.txt`

The `--hosts` argument can take patterns in the form `[X-Y]` to indicate a range of host Name/IP. This argument can also take patterns in the form `[X,Y,Z]` to indicate a list of hosts with X,Y, and Z substituted at the pattern location to operate on `host1`, `host2`, `host3`, `host5`, `host7`, `host-a`, `host-b`, and `host-c`, you can use a `--hosts` argument such as `"host[1-3] host[5,7] host-[a-c]"`. Patterns must be specified in `[]` or `{ }` brackets.

If a command fails, `clusterexec` does not run subsequent commands by default. Use the `-k` option to override this behavior and "keep going" after failure.

`clusterexec` command-line options are described in the following table:

`clusterexec` **Command-line Options**

Switch Options	Description
<code>-h, --help</code>	Prints a usage message summarizing information in this table.
<code>--hosts=<value></code>	Set of hosts on which to run commands. <i>value</i> should be in the form <code>host1</code> <code>host2</code>
<code>--cm=<host[:port]></code>	Cluster Manager that is contacted to get the relevant hosts. This is used only when <code>--hosts</code> is <i>not</i> specified. If this option is specified, commands run on all hosts in the cluster, subject to the <code>--platform</code> and <code>--good-hosts</code> options below. Defaults to the value of the <code>EMAKE_CM</code> environment variable (if present).

Switch Options	Description
<code>--platform=<value></code>	Platform of desired hosts. This is used only when <code>--cm</code> is specified and <code>--hosts</code> is <i>not</i> specified. <i>value</i> is either <code>windows</code> , <code>solaris</code> , <code>solarisx86</code> , or <code>linux</code> . Commands are executed on hosts of the named platform type only. Defaults to the platform on which this client program is running.
<code>--good-hosts-only</code>	Only use hosts with at least one pingable enabled agent in Cluster Manager. This option applies only when <code>--cm</code> is specified. Default behavior uses all hosts regardless of their state.
<code>--file=<value></code>	Name of a file containing commands to run. If this option is specified, an inline script (as shown above), must <i>not</i> be specified.
<code>-k, --keep-going</code>	Continue running commands after failure. If more than one command is specified, continue to run subsequent commands, even if an earlier command fails.
<code>--mergestreams=<0/1></code>	Default = <code>true</code> (merge <code>stdout</code> and <code>stderr</code> commands and write to <code>clusterexec stdout stream</code>). If set to <code>false</code> , <code>stdout</code> and <code>stderr</code> commands are sent to <code>clusterexec stdout</code> and <code>stderr</code> , respectively. Note: You cannot disable <code>mergestreams</code> if you enable annotation. Enabling annotation automatically enables <code>mergestreams</code> , even if it was explicitly disabled.
<code>-v, --version</code>	Displays <code>clusterexec</code> version.
<code>-s, --use-shell</code>	Send the entire command string to the shell (<code>sh</code> on UNIX, <code>cmd</code> on Windows).
<code>--timeout=<value></code>	Abort execution after <i>value</i> seconds.

Commands may refer to executables on hosts or commands built into the eRunner service/daemon on the hosts. Valid built-in commands are listed in the following table:

eRunner Built-in Commands

Commands	Description
<code>stopAgent</code>	Stops the Electric Agent service on the hosts.
<code>startAgent</code>	Starts the Electric Agent service on the hosts.
<code>restartAgent</code>	Restarts the Electric Agent service on the hosts.
<code>reboot [<delay>]</code>	Reboot the hosts. If delay is specified, reboot each host <i>delay</i> ms (milliseconds) apart.
<code>stopEranner</code>	Shutdown eRunner service on the hosts.

Commands	Description
<code>restartEranner</code>	Restart eRunner service on the hosts.
<code>printVersion</code>	Print the eRunner service version on the hosts.
<code>logLevel <level></code>	Query or set the eRunner service log level on the hosts. Valid values for <i>level</i> are <code>DEBUG</code> and <code>INFO</code> .

Using clusterupload

The `clusterupload` command is a convenient tool for transferring a file or files to all hosts in a cluster. This command can upload one or more files to one or more hosts in the cluster. The syntax for running `clusterupload` is:

```
clusterupload [options] <source> <target>
```

where *target* could be a file or a directory. The interpretation depends on whether or not *target* already exists on the host (as a file or as a directory). To interpret the logic, *source* is the relative or absolute path of the file/directory on the local machine and *target* is the absolute path to the file/directory on the hosts. Then,

- if *source* is a file and *target* is a directory, *source* is copied into *target*.
- if *source* is a file and *target* is a file, *source* is copied over *target*.
- if *source* is a directory, and *target* is a file, an error is reported.
- if *source* is a directory ending in "/" or "\", the *source* contents are copied into the *target* if *target* is a directory. If *target* is not a directory, an error is reported.
- if *source* is a directory not ending in "/" or "\", the last component of the *source* path becomes a subdirectory under *target*, and the *source* contents are copied into that subdirectory.

Target paths need not exist in advance—relevant directories are created as needed. If you specify a relative path for *target*, `clusterupload` gives you an error—relative paths are not allowed.

For taking multiple *sources*, the syntax is:

```
clusterupload [options] <source1> <source2> <target>
```

where *target* is assumed to be a directory. (The *target* is always assumed to be a directory when multiple sources are specified.)

Switch option values are described in the following table:

`clusterupload` **Command-line Options**

Switch Options	Description
<code>-h, --help</code>	Prints a usage message summarizing information in this table.
<code>--verbose</code>	Verbose mode.
<code>--hosts=<value></code>	Set of hosts within the cluster to update. <i>value</i> should be in the form "host1 host2[:port] ..."

Switch Options	Description
<code>--cm=<host[:port]></code>	The Cluster Manager contacted to get relevant hosts. This is used only when <code>--hosts</code> is <i>not</i> specified. If this option is specified, commands are run on all hosts in the cluster, subject to the <code>--platform</code> and <code>--good-hosts</code> options below. Defaults to the value of the <code>EMAKE_CM</code> environment variable (if present).
<code>--platform=<value></code>	Platform of desired hosts. This is used only when <code>--cm</code> is specified. <i>value</i> is either <code>windows</code> , <code>solaris</code> , or <code>linux</code> . Commands are executed on hosts of the specific platform type only. Defaults to the platform on which this client program is running.
<code>--good-hosts-only</code>	Only upload to hosts pingable and enabled in Cluster Manager. This option applies only when <code>--cm</code> is specified. Default behavior is to upload to all hosts regardless of their state.
<code>--filelist=<value></code>	Filename containing a list of source and target files to upload. If this option is specified, <i>source</i> and <i>target</i> must not be specified. Each line in the file must be in the form: <i>sourcePath => targetPath</i> . If <code>--filelist</code> is specified with a dash value, file list data is read from <code>stdin</code> .
<code>--stop-agents</code>	Stops Agents when committing uploaded data to its final location.
<code>-v, --version</code>	Displays <code>clusterupload</code> version.
<code>--timeout=<value></code>	Aborts execution after <i>value</i> seconds.

Using clusterdownload

The `clusterdownload` command is a convenient tool for transferring a file or files from hosts in a cluster to a central location. This command downloads one or more files to one or more hosts in the cluster. The syntax for running `clusterdownload` is:

```
clusterdownload [options] < sourcedir > < targetdir >
```

The target directory does not need to exist in advance; it is created as needed. Relative paths are supported for *targetDir* but not for *sourceDir*.

For downloading multiple sources, you can use a pattern in *sourceDir* or use the `--include` option:

```
clusterdownload [options] "/opt/eccloud/i686_Linux/logs/*.log" <target>
```

Note: If wildcard characters are used, quotes are required so the command is not expanded first by the shell.

-- or use --

```
clusterdownload [options] --include=*.log /opt/eccloud/i686_Linux/logs <target>
```

The second example above copies all `.log` files from the specified directory.

Switch option values are described in the following table:

`clusterdownload` **Command-line Options**

Switch Options	Description
-h, --help	Prints a usage message summarizing information in this table.
-v, --version	Print version information.
--verbose	Verbose mode: shows progress information as well as connection requests.
--hosts=<value>	Set of hosts within the cluster from which files need to come. <i>value</i> should be in the form: "host1 host2[:port] ..." Patterns can be used also: "host[1-3] host[5,7] host-[a-c]" Also you can use curly brackets instead of square brackets. The --hosts option overrides the --cm option.
--cm=<host[:port]>	Cluster Manager that is contacted to get the relevant hosts. This is used only when --hosts is <i>not</i> specified. If this option is specified, the commands run on all hosts on the cluster, subject to the --platform and --good-hosts options below. Defaults to the value of the <code>EMAKE_CM</code> environment variable (if present).
--platform=<value>	Platform of desired hosts. This is used only when --cm is specified. <i>value</i> is either <code>windows</code> , <code>solaris</code> , or <code>linux</code> . Commands are executed only on hosts of the specific platform type. Defaults to the platform on which this client program is running.
--timeout=<value>	Abort the connection after <i>value</i> seconds. Any partially downloaded files are discarded.
-r, --recursive	Recurse into subdirectories.
-u, --update	Update only. Do not overwrite newer files. File modification times are used.
-t, --times	Preserve file timestamps.
-n, --dry-run	Show what would have been transferred.
--existing	Only update existing files.
--progress	Show progress during the file transfer. When this option is used, <code>clusterdownload</code> lists each file when it begins its download. <i>Progress</i> also prints dots (...) as files are being written to identify an active process.
--exclude=<pattern>	Exclude files matching <i>pattern</i> . Patterns can include the usual ? and * meta-characters. Patterns apply to filenames, not to paths. When a file matches both the include and exclude patterns, it is excluded.

Switch Options	Description
<code>--include=<pattern></code>	Include files matching <i>pattern</i> . Patterns can include the usual ? and * meta-characters. Patterns apply to filenames, not to paths. When a file matches both the include and exclude patterns, it is excluded.
<code>--logfile=<file></code>	Set the logfile name. This can include an absolute or relative path. If no path is specified, the default location on Solaris and Linux is <code>/var/log</code> . On Windows the default location is where ElectricAccelerator is installed. (Default = <code>c:\ECloud\i686_win32</code>) If the target file is not writable for some reason, the log is written to <code>STDOUT</code> . If you specify <code>--logfile</code> , you must also specify <code>--loglevel</code> because there is no default log level.
<code>--loglevel=<info debug></code>	There is no default log level. If you specify <code>--loglevel</code> but not <code>--logfile</code> , the default log filename is <code>clusterdownload.log</code> in the location described above. To set log levels on Cluster Manager or on agent hosts see <code>clusterexec</code> .

Sample Uses of Cluster Tools

In the following example, `clusterupload` is used to upload a new version of `bash`, and `clusterexec` is used to run a command on ElectricAccelerator cluster hosts.

1. Upload `bash` to `/usr/bin` on a Linux/Solaris cluster:

```
% clusterupload --cm=dilbert-cm ~/download/bash /usr/bin/bash
```
2. Find out how long hosts have been up on a Linux/Solaris cluster:

```
% clusterexec --cm=dilbert-cm uptime
```

The result would be similar to:

```
dilbert1.electric-cloud.com output:
16:02:46 up 5:47, 0 users, load average: 0.00, 0.04, 0.07
```

```
dilbert2.electric-cloud.com output:
16:02:45 up 5:47, 0 users, load average: 0.08, 0.04, 0.06
```

```
dilbert3.electric-cloud.com output:
16:02:46 up 6:16, 0 users, load average: 0.00, 0.02, 0.05
```

```
dilbert4.electric-cloud.com output:
16:02:45 up 6:16, 0 users, load average: 0.00, 0.01, 0.02
```

```
dilbert6.electric-cloud.com output:
16:02:45 up 6:17, 0 users, load average: 0.00, 0.04, 0.05
```

```
dilbert5.electric-cloud.com output:
16:02:45 up 6:17, 0 users, load average: 0.00, 0.00, 0.01
```

In the next examples, `clusterexec` is used to gather various statistics on an ElectricAccelerator host cluster.

1. Find out how long hosts have been up on a Windows cluster:

```
C:\> clusterexec --cm=win-cm "net statistics workstation"
```

The output would be similar to:

WIN2 output:

```
Workstation Statistics for \\

Statistics since 10/25/2005 3:54 PM
. . .
The command completed successfully.
```

WIN1 output:

```
Workstation Statistics for \\

Statistics since 10/25/2005 4:02 PM
. . .
The command completed successfully.
```

2. Get the last 5 lines of agent log files on host1:

```
% clusterexec --hosts=host1 -s "tail -5 /var/log/ecagent*.log"
```

The result would be:

host1 output:

```
==> /var/log/ecagent1.log <==
*** -numagents 2
*** -version
*** -webport 8001
***-----

==> /var/log/ecagent2.log <==
*** -numagents 2
*** -version
*** -webport 2421
***-----
```

In the example above:

To obtain the last 5 lines of log files for all Agents, an `sh` invocation is necessary. Use the `***` wildcard to achieve this result. To use the wildcard, a shell must invoke the expansion; `clusterexec` does not automatically invoke a shell in which to run commands.

`clusterexec` does not handle pipelines. For pipelines, the `-s` option should be used, for example:

```
clusterexec --hosts=host1 -s "ps -ef | grep agent"
```

By default, `clusterexec` stops running commands after the first failure, for example:

```
clusterexec --hosts="lin1" "badCmd.sh; /bin/echo hi"
```

will not run the `echo` command on `lin1` if `badCmd.sh` returns a non-zero exit code.

A simple example for a Windows cluster download:

```
clusterdownload --cm=mycm c:/ECloud/ecagent*log c:/tmp
```

This example downloads all agent log files in `c:\ECloud` from all agent hosts on the cluster to directory `/tmp`. Suppose your cluster has `host1` and `host2` and each host has 2 agents:

On the host where you ran `clusterdownload`, the result is:

```
/tmp/host1/ecagent1.log
/tmp/host1/ecagent2.log
/tmp/host2/ecagent1.log
/tmp/host2/ecagent2.log
```

To get agent logs from Solaris or Linux hosts use:

```
clusterdownload --cm=mycm "/var/log/ecagent*.log" /tmp
```

If you want to get all the log files under `/opt/ecloud`, you could use:

```
clusterdownload --cm=mycm -r "/opt/ecloud/*.log" /tmp
-- or --
clusterdownload --cm=mycm --include=*.log -r /opt/ecloud /tmp
```

Note: If wildcard characters are used, quotes are required so the command is not expanded first by the shell.

Stopping, Starting, or Restarting the eRunner Daemon or Checking Its Status

Stopping, Starting, or Restarting the Daemon

On Linux and Solaris platforms, you can stop, start, or restart the eRunner daemon by using the following commands:

```
/etc/init.d/erunner stop
/etc/init.d/erunner start
/etc/init.d/erunner restart
```

Checking whether the Daemon is Running

On Linux and Solaris platforms, you can check whether the eRunner daemon is running by using the following command:

```
/etc/init.d/erunner status
```

If the daemon is running, the following message appears:

```
erunner running with PID <PROCESS_ID>
```

If the daemon is not running, the following message appears:

```
erunner is not running
```

Using ecconfig

The `ecconfig` program lets you configure agents. Open a command-line window to use `ecconfig`.

Option	Description
<code>-cm <host:port></code>	Host and port of the Cluster Manager to which the agent host should point. If the default port (8030) is used, omit the port number.
<code>-numagents <number></code>	Number of agents to start. If you modify <code>numagents</code> , you must restart the agents for your change to take effect. On Windows platforms, you can restart the agents by entering the <code>ecagent_restart</code> command at a command prompt. On Linux and Solaris platforms, you can do so by entering <code>/etc/init.d/ecagent restart</code> at a shell prompt. After the restart, the server database is updated, and the agents become active.
<code>-agentusers "<user1>/<password1> <user2>/<password2>...<userN>/<passwordN>"</code>	(Windows platforms) Specifies multiple agent user names and passwords. The quotes are required. Note: You must specify the same number of users as the actual number of agents.
<code>-agentuser "<agentID> <user>/<password>"</code>	(Windows platforms) Specifies the user name and password for a specific agent. <code><agentID></code> is the numerical ID of the agent to configure (between 1 and the actual number of agents). The quotes are required.
<code>-lsfhost <value></code>	Specifies whether this machine is part of an LSF grid. Enter <code>y</code> (yes) or <code>n</code> (no).
<code>-maxopenfds <value></code>	Set the maximum number of open file descriptors for agents. To clear the number and revert to the OS default, enter <code>ecconfig -maxopenfds ""</code> . The value is recorded via the <code>MAX_OPEN_FDS</code> field in the <code>ecagent.conf</code> file. The recommended value is 8192.
<code>-secureconsole <value></code>	Specifies whether to use the secure agent console port. Enter <code>y</code> (yes) or <code>n</code> (no). The default is <code>n</code> (no).
<code>-tempdir <path></code>	Location where agents store temporary files. To revert to the system default, use double quotes as the argument (<code>-tempdir ""</code>). The system default is <code>/tmp</code> on Linux and Solaris and <code>C:\WINDOWS\Temp</code> on Windows.
<code>-help, -?</code>	Displays the <code>ecconfig</code> command usage message.

Note: `ecconfig` does not let you set agent user names to nonunique names. If you must do so, reinstall the agents or contact Electric Cloud technical support for the necessary registry fix.

Using ecagent Tools

You can issue scripts from the agent host to assist with managing agent services or daemons. The following scripts are available:

- `ecagent_start` or `/etc/init.d/ecagent start`—Run this command to start the agent service or daemon. For example: To start a stopped agent, issue `ecagent_start`, and the agent will be detected automatically and reappear in the Cluster Manager agent list in the web interface.
- `ecagent_stop` or `/etc/init.d/ecagent stop`—Run this command to stop the agent service or daemon. For example: To delete an agent permanently, issue `ecagent_stop` and then delete the agent from the Cluster Manager agent list in the web interface.
- `ecagent_restart` or `/etc/init.d/ecagent restart`—Run this command to stop and then start the agent service or daemon.

Using ecinstconf

Open a command-line window to use `ecinstconf`. The following table is the output from running `ecinstconf -help`.

Usage: ecinstconf option [new value]	
-agentcmhost	get/set the agent cluster manager host
-agentcmport	get/set the agent cluster manager port
-agents	get/set the agent number
-agentreboot	get/set the agent reboot selection
-agentdrive	get/set the agent drive letter
-agentinfo1	get/set the agent info
-agentinfo2	get/set the agent info
-agentlogremove	get/set the agent log removal
-cmport	get/set the cluster manager listening port
-cmserverport	get/set the cluster manager accelerator server listening port
-cmphpserverport	get/set the accelerator server port that the CM web UI should use (startup.php)
-cmacceleratoradmin	get/set the accelerator server admin username
-cmserveradmin	get/set the cluster manager admin mail address
-cmservername	get/set the cluster manager name
-cmdbport	get/set the cluster manager database port
-cmmaxclients	get/set the cluster manager MaxClients setting, which controls the number of simultaneous requests the CM will handle
-cmmaxrequestsperschild	get/set the cluster manager MaxRequestsPerChild setting
-cmrotatelogerror	get/set the cluster manager error log rotation
-cmrotatelogcustom	get/set the cluster manager custom log rotation
-cmmigratedb	get/set the cluster manager database migration

-cmstatssoft	get/set the cluster manager stats soft limit
-cmstatshard	get/set the cluster manager stats hard limit
-cmstrictusers	get/set the cluster manager strict user check
-cmallowcookies	get/set the cluster manager cookie permission
-cmcookieexpiration	get/set the cluster manager cookie expiration time
-cmcookiedomain	get/set the cluster manager cookie domain
-cmmakeheartbeat	get/set the cluster manager emake timeout
-cmconntimeout	get/set the cluster manager connection timeout
-cmwebtimeout	get/set the cluster manager web timeout
-cmwebmaxattempts	get/set the cluster manager web max attempts
-cmmaxagents	get/set the cluster manager max agents
-cmmaxbuilds	get/set the cluster manager max builds
-cmversionmismatch	get/set the cluster manager version mismatch file
-cmcharset	get/set the cluster manager character set
-cmlogfilepath	get/set the cluster manager build log file path
-cmlogfileprocessor	get/set the cluster manager log file processor
-cmlogremove	get/set the cluster manager log removal
-upgradehost	get/set the agent upgrade host
-upgradeport	get/set the agent upgrade port
-dbuser	get/set the database root user
-dbpassword	get/set the database root password
-dbbasedir	get/set the database base directory
-dbdatadir	get/set the database directory
-dbtmpdir	get/set the database tmp director

-dbsocket	get/set the database socket path
-dbserve	get/set the database server path
-dbcharset	get/set the database charset

Using eclock

Eclock is a tool that allows you to serialize multiple runnings of a program. For example, it may not be safe to run two copies of a particular application simultaneously on a single computer, even if it is run by different users—an example is vb6. Use eclock (typically in makefiles) to cause various invocations of that program to wait for each other. Eclock is located in the 32-bit `bin` directory.

An eclock example:

```
eclock.exe vb6 arg1 arg2 arg3
```

Using the previous example, multiple invocations of vb6 are now serialized. Note that direct vb6 invocations are not serialized.

Unless you use the `--mutex` option, you get a reasonable default for your system. The `%1` in that default (or your override) is replaced by the base filename of your program (without the trailing `.exe` on Windows). (Also, `%%` is replaced by `%`.) After such expansion, all runs of eclock that refer to the same mutex take turns, running one at a time (in an unspecified order). If two eclock invocations have different mutexes, after expansion, then the eclock invocations run concurrently, without interfering with each other. Arguments are passed as-is to the target program, and the output and exit code are the same as the target program.

Note: Mutex names are case sensitive.

For Solaris and Linux only. The lock file (the `--mutex` option specifies a filename pattern for the lock file) must go on a local disk **outside** of `$EMAKE_ROOT`. Otherwise, the result could be concurrency instead of exclusion. The pathname is interpreted as relative to `/tmp`. Eclock may append a suffix of its own to the filename component to produce the final name of the lock file.

For Windows only. `%1` expands to the lowercase equivalent of the base program filename. If you override the default mutex name, do not use any backslashes. The only exceptions are that you may start the mutex name with either `"Local\"` to force the mutex to be session-local (which is not general practice) or `"Global\"` (which makes the default explicit).

The following help text is the eclock Windows help.

```
eclock
Usage:
eclock.exe [--mutex <name>] <process> <process args>

-m or --mutex or /mutex - specify
    mutex name, overrides ECLOCK_MUTEX
-h or --help or /help or /? - print this help and exit

Environmental variables:
ECLOCK_MUTEX - specify mutex name, default 'eclock_%1'. Use %1 to substitute
    process name
ECLOCK_BUSY_WAIT - CPU used every half minute while waiting (milliseconds,
    default 100)
ECLOCK_DEBUG - turn on debugging
ECLOCK_DEBUG_LOG - filename of debug log (stderr if omitted)
```

In case multiple mutexes are required, this program may append suffixes to the mutex name. And unless you give a pattern starting with `'Local\'`, mutexes are global. Case matters in mutex names.

Using the runagent.local File

The runagent.local file allows you to hard-code configuration options into an agent, without modifying Electric Cloud files. Using the runagent.local file may become necessary to obtain specific agent behavior or to resolve certain issues. This topic discusses a few of the most common reasons for using the runagent.local file.

You must create the file in the `<ECloud install>\<arch>\bin` directory of the agent hosts. Then you must restart the agents.

Preventing "runaway" Jobs

To protect against "runaway" jobs running on the cluster, the agent checks the amount of output text produced by a job. If the amount of output exceeds a threshold, then the agent sends back a portion of the output only and dumps the remainder. This prevents Accelerator from consuming too many resources in the agent, on the network, and in eMake to store and transmit what is likely extraneous output.

This feature relates to only the amount of standard output text produced by a job and has nothing to do with the size of any files produced by the job. Also, this feature does not in any way affect the running of the job, only the amount of output text that is included in the log file. These "errors" are not errors at all; they are only warnings to notify you that there may be a problem with the job. If you know there is nothing wrong with the job, then you can safely ignore the warnings.

Add the following to runagent.local to change when and how this feature takes effect.

```
set TruncationThreshold <threshold>
set TruncationLength <length>
```

Threshold is the number of bytes at which this feature takes effect (default is 5 MB), and length is the number of bytes to return after the feature takes effect (default is 5000 bytes).

You can verify the result by doing the following after restarting the agents:

```
cmtool --cm=<cm> runAgentCmd "session exec truncationparam"
```

Configuring Stalled Job Detection Settings

This is a Windows-only setting.

Adjust the timeout used by the agent to detect stalled commands on the agent. Add a call to the following command in the runagent.local file:

```
exec cmtool --cm=<cm> login <user> <password>
exec cmtool --cm=<cm> runAgentCmd {agentexec timeout {{.* 120000 {cpu disk}}}}
```

The format of the commandTimeout value (in this instance, the argument to "agentexec timeout") is a list of the following three-element-lists:

- A regular expression used to match the process name.
- The amount of time with no activity after which the process is considered stalled, in milliseconds.
- A list of the process attributes which will be monitored for activity. To monitor for activity, this must contain at least one of "cpu", which means the agent will watch for CPU activity from the process; and "disk", which means it will watch for I/O activity. It may contain both.
- You can explicitly disable stalled-job detection for a process by adding an entry to the commandTimeout value with the timeout duration set to zero, or by specifying neither "cpu" nor "disk" in the attribute list.

You can verify the settings by doing the following after restarting the agents:

```
cmtool --cm=<cm> runAgentCmd "agentexec timeout"
```

Configuring Agent Log Rotation

ElectricAccelerator automatically rotates session logs using default settings. Information about session logs:

- stdout/stderr logs go to `consoleN.log` (where N is the agent number).
- Normal session trace messages go to `ecagentN.log`.
- Only session trace logs will be rotated.
- The default size limit for log files before being rotated is 1 GB.
- The default number of rotated files to keep is 20. When that limit is reached, the oldest log files will begin to be deleted to make space for newer logs.
- The most recent log messages are in `ecagentN.log`. When the log file reaches its size limit, that file is renamed to `ecagentN.log.1`. If `ecagentN.log.1` already exists, then it is renamed to `ecagentN.log.2` before `ecagentN.log` is renamed to `ecagentN.log.1`. If `ecagentN.log.2` exists, it is first renamed to `ecagentN.log.3` and so on. This rotation continues until the retain count is met, at which point file deletion begins.

To configure log rotation, use the agent's `[session logger]` command.

Use this format: `session logger <option>`

For example: `session logger retaincount 10`

Available log rotation options

Option	Description
<code>maxfilesize <bytes></code>	Get/set the maximum file size for log files before they are rotated. The default is 1 GB.
<code>filename <path></code>	Get/set the file name to log to. The default for UNIX is <code>/var/log/ecagentN.log</code> . The default for Windows is <code><install dir>\ecagentN.log</code> (for example, <code>C:\ECloud\ecagentN.log</code>).
<code>retaincount <count></code>	Get/set the number of rotated log files to retain before starting to delete them. The default is 20. To turn off log rotation, set to 0.
<code>clear</code>	Clear the contents of the <code>ecagentN.log</code> file. This is useful when log rotation is turned off.

System Maintenance Tasks

General maintenance of build-generated data and logs ensures a cleaner operating environment and quicker response times in the build environment. Some basic user maintenance may prove useful for running builds also. Basic build environment maintenance is included in this topic.

Cleaning Up Data

To increase system response time and help users stay organized, it is a good idea to periodically delete obsolete build data and Agent logs for the Cluster Manager.

Deleting from the Builds List

The builds list includes Cluster Manager build history. It is important to remove build logs periodically so they do not fill up the Cluster Manager's available disk space. Uploaded annotation is also considered part of build logs, so remember to clean up build logs regularly if annotation is frequently uploaded to the Cluster Manager.

Deleting Extraneous or Obsolete Builds

1. Click the **Builds** tab to open the build history list.
2. Filter the build list by your desired criteria; for example, Good Builds or Builds by Date, and so on.
3. Click **Delete Filtered Builds**.

Confirm your decision at the prompt to delete the results. After you confirm the delete, it cannot be undone.

You can automate build deletion tasks by using the `cmtool deleteBuilds` command.

For example, if you are using MySQL, this example deletes all builds more than 20 days old:

```
cmtool deleteBuilds --filter "start_time <date_sub(curdate(), interval 20 day) "
```

Note: If you use a different database, use syntax that is appropriate for your respective database.

Managing the Message Log

The Message Log contains an error event list for all agents in the cluster. Message Log format is relatively free-form and descriptive. Message entries contain:

- severity of the error being reported
- log entry date and time
- build number
- message text
- name of the Agent that generated the error

Using the Message Log, you can track Agent issues by severity. If an Agent experienced a high number of error events, it may indicate a problem with the system setup or hardware.

The Cluster Manager automatically deletes Message Log entries older than a configurable number of days. See [Administration > Server Settings](#), the "Message Log Keep (days)" setting.

Securing the Cluster Manager Database

Default settings are located in the `accelerator.properties` file.

```
ACCELERATOR_JDBC_USER_NAME  
ACCELERATOR_JDBC_PASSWORD
```

The default user name and password are both set to `ecloud`.

Database Configuration

Configuring Accelerator to Use an External Database

Before converting to an external database using MariaDB, Microsoft SQL Server, MySQL, or Oracle, ensure the database is up and running correctly. (Electric Cloud bundles MariaDB and installs it as the default local database on the Cluster Manager.)

- For Oracle, ensure that when you create a user, Roles has **Resource** added.
- For Microsoft SQL Server, ensure that when you create a user, Server Roles has **sysadmin** (or another server role that allows the necessary actions) enabled.

1. Fill in the fields as follows:

- Database Name—Database instance name.
- Database Type—MariaDB, Microsoft SQL Server, MySQL, or Oracle.
- Host Name—Machine name where the database is installed.
- Port—Database port number. The default port is filled in automatically after Database Type is selected. Type a different port number if applicable.
- User Name—Unique name of the user that is used to access the database.
- Password and Confirm—Secret value that identifies an account for a particular user.

The Cluster Manager tests the database connection before saving `database.properties`. If the connection is unsuccessful, an error appears, and `database.properties` is not saved.

2. Re-import your license.
3. Restart the Cluster Manager service.

Note: After changing the database type, your original database service might continue to start each time the Cluster Manager service starts. Manually disable the database service.

Backing Up and Restoring a Database

You use the `exportData` and `importData` `cmtool` commands to back up and restore a database.

Note: Manual migration using the `exportData` and `importData` commands is recommended only to replicate data between Cluster Manager instances running the same version of the Cluster Manager. For example, you could use it to make a backup dump and then restore it to a new instance of the Cluster Manager of the same version.

Backing Up Your Database

Run the following command to export the Cluster Manager data to a file:

```
cmtool exportData <file_name>
```

where `<file_name>` is the file name or path to export to. If you use a file name, the destination is the current working directory of the Java process (`/opt/ecloud/i686_Linux` or `C:\ECloud\i686_win32` by default. If you use a path, the Cluster Manager Java user (`eacmuser`) must have execute and write access to that path.

Note: This is a full database dump, which might take substantial time for a large database.

Restoring Your Database

Run the following command to import the Cluster Manager data from a file:

```
cmtool importData <file_name>
```

where *<file_name>* is the name of the file to import. The file's path is relative to the current working directory of the Java process (/opt/eccloud/i686_Linux or C:\ECloud\i686_win32 by default).

Note: This command imports a full database dump, which might take substantial time for a large database.

Make sure that you

- manually delete any old or unused agents from the agents list.
- update the license file after import, if it has previously expired.

Enabling LDAP

To enable LDAP, follow this procedure:

1. Go to the following directory on the Cluster Manager server: `<ECloud install>/<arch>/conf`
2. Make a copy of the `ldap_template.xml` file if you intend to use LDAP, or the `ad_template.xml` file if you prefer to use Active Directory. Save the copy in the `conf` directory as any name you choose or use the name: `securityContext.xml`.
3. Still working in `conf`, open the `accelerator.properties` file.
4. Locate the following commented-out text string:
`#ACCELERATOR_SECURITY_CONTEXT_FILES=conf/securityContext.xml`
5. If you did not name your template copy `securityContext.xml`, replace `securityContext.xml` with the filename you chose.
6. Uncomment the `#ACCELERATOR_SECURITY_...` text string by removing the lead “#” sign.
7. Comment out `ACCELERATOR_SECURITY_CONTEXT_FILES=` (which immediately follows the line you uncommented).
8. Update your own copy of the LDAP template with your LDAP configuration details.
9. Restart the Cluster Manager.
10. After the Cluster Manager is running, log in to the Cluster Manager UI as “admin”.
11. Go to Administration > Permissions.
12. Click **Enable User**.
13. Search for “*”.

If you set it up correctly, all LDAP users are visible.

Import License

Use this page to provide a license to the ElectricAccelerator server.

You should already have received a license from Electric Cloud. If you did not receive a license, contact Electric Cloud Customer Support or your sales representative.

Importing a License

Do one of the following:

- Browse for and upload the license file.
- Follow these steps:
 1. Use a text editor to open the license file.
 2. Copy the text into the Data field.
 3. Click **OK**.
 4. Go back to the License page to see your license information.

Using Comments

Using the Comments section is a convenient way to promote build team communication, track build or hardware issues, assign tasks, or any other information you need to maintain or share in permanent or temporary notes. A Comments section is available on the following product web pages:

- Home—While comments on this page are intended for server-specific information, you can use this comment section for other general messages or notes, for example, system-wide notices, schedule announcements, task assignments, and so on. Home page comments are available immediately each time you log in or "one-click away" if you are on another Accelerator web page.
- Build Details
- Build Class Details
- Agent Details
- Resource Details

Adding a New Comment

A user may view or add comments only if permissions were granted by an administrator.

1. Click the **New Comment** link.
This displays the New Comment page.
2. Type-in the information you want to share or track.
3. Click **OK**.

Return to the (Home, Build Details, Build Class Details, Agent Details, or Resource Details) page to see your new comment displayed in the Comment box.

Using HTML for Comments

Instead of typing a comment directly into the comment box, you may "cut and paste" HTML text into the comment box or add HTML tags to existing text (for example, to create a link). Using HTML text in the comment box, however, is problematic if the HTML is incorrect or unconventional—it may "lock" the comment box, preventing the addition of more comments. If this occurs, use `cmtool` (the command-line tool) to clear the offending comment. The solution is:

1. In a command-line console, type `cmtool getServerComments`
2. Find the `commentId` for the target comment, for example, `<commentId>12</commentId>`
3. Type `cmtool deleteServerComment <commentId>` for example: `cmtool deleteServerComment 12`

Now if you re-type: `cmtool getServerComments`, you will not see the offending comment. Any other comments already in the comment box are not affected and you will be able to add more comments as needed.

Editing a Comment

At any time, you may click **Edit Comment** to update a comment.

Deleting a Comment

At any time, you may click **Delete Comment** to delete a comment. When you click **Delete Comment**, a you have a chance to confirm or cancel your delete request.

Stopping Builds

You can stop an in-progress build by one of three methods:

- Press **Ctrl-C** from the terminal where you invoked Electric Make
- Use ElectricAccelerator's web interface
- Use `cmtool`—ElectricAccelerator's command-line application tool

The Cluster Manager automatically terminates builds that seem to be hung. If the Cluster Manager does not receive a request from eMake for 60 seconds, it considers the build hung and the build is terminated.

Stopping a Single Build from the ElectricAccelerator Web Interface

1. Click the Builds tab.
2. Click **Stop Build** in the Action column on the row exhibiting your build ID and Name.

Stopping Multiple Builds from the Web Interface

1. Click the Builds tab.
2. Go to the Filter drop-down menu and select the desired filter.
3. Click **Run Filter**.
4. Click **Stop Filtered Builds**.
5. From the Filter menu, select none, then click **Run Filter**.

Stopping a Build Using `cmtool`

Note: This is an "advanced user" option, unless you are already familiar with using command-line tools.

1. Request a list of running builds. The syntax is:

```
% cmtool --cm <clustermanager:port> getBuilds --filter <field name>=<value>
```

For example, if linuxbuilder is the Cluster Manager host name, type:

```
% cmtool --cm linuxbuilder getBuilds --filter "result =1"
```

A list of running builds will display accompanied by a number of attributes for each build—for example: Build ID, machine name, build class, owner, build start time, and so on. The Build ID is used to identify a build for termination. Also, you can obtain additional information and/or include comments about the build. For example, to sort builds by start time and request only the first ten builds display, enter:

```
% cmtool --cm linuxbuilder getBuilds --order Id --filter="ID<11"
```

To get failed builds:

```
% cmtool --cm linuxbuilder getBuilds --order "start_time desc" --filter "result >0"
```

2. After you determine which build you need to terminate, use this syntax for ending a build:

```
% cmtool --cm <clustermanager> stopBuild <buildId>
```

For example, to end build 4458, type:

```
% cmtool --cm linuxbuilder stopBuild 4458
```

Deleting Builds

Deleting a Single Build

On the Builds page, click the **Delete Build** link in the Action column for the build you want to delete.

On the Build Details page, click **Delete Build** to delete that displayed build.

Deleting a Group of Builds from the Build List

1. Go to the Filter drop-down menu and select the desired filter.
2. Click **Run Filter**.
3. Click **Delete Filtered Builds**.
4. From the Filter menu, select none, then click **Run Filter**.

Creating Build Classes

1. On the Build Classes page, click **New Build Class**.

This takes you to an empty build class details page.

2. Name the build class.

The build class name identifies the class to Electric Make for invoking a build.

You must assign the build's class name through an Electric Make command-line option when the build is invoked: `--emake-class=<exact build class name>`. If no build class is assigned, the Cluster Manager assigns the build to the default class. If the class name typed on the eMake command line does not match a class name already in Cluster Manager, eMake exits.

3. Enter the build class tag definition.

To avoid errors, follow standard naming conventions for tag definitions by using numbers, letters, and underscores only **without** leading or trailing white space. Use a percent sign on either side of any variables used. (For example, `%DATE%`.) Use underscores (`_`) instead of spaces.

4. Edit the remaining fields or keep their default values. See [Build Classes- Create or Edit a Build Class](#) for field definitions.

5. Click **OK** when all fields are filled-in.

When you return to the Build Classes page, you will see the new build class in the table.

Build Class Examples

Suppose your company has two major product lines: SuperSoftware and MegaSoftware. SuperSoftware runs on Windows and Solaris platforms. MegaSoftware runs on Windows only. You could begin by setting up three classes that include the product name, the platform, and the current version number for each product:

- You could name the first class `Supersoftware_Win_v.2.1`. The tag definition for this class would be:

```
%BUILD_CLASS%_%LC%_%a%_%b%_%d%_%H%_%M%_%S%
```

The result would be a series of builds each named, or *tagged*, with the product name, the platform, the version number, a serial number (unique to the class), and the date for each build, for example:

```
Supersoftware_Win_v.2.1_12345_WED_AUG_22_14_37_12
```

- You could name the second class `Supersoftware_Sol_v.1.7`. The tag definition can be the same as in our first example because it would be distinguished by the second build class name. Build tags in the second class would look like:

```
Supersoftware_Sol_v.1.7_12356_WED_AUG_23_11_14_39
```

- You could name the third class `MegaSoftware_Win_v.1.3`. For this product, the tag definition would be similar to the previous examples, but also could include the name of the user who started the build because the MegaSoftware team is spread over several different locations. For this class, the tag definition might look like: `%BUILD_CLASS%_%LC%_%USER_NAME%_%DATE%`

As in the first two examples, the result would be a sequentially numbered series of builds with the product name, platform, version number, name of the user who ran the build, and the date of each build assigned through the build class:

```
MegaSoftware_Win_v.1.3_12356_JSMITH_20050411100838
```


Additional classes could be created when the development of SuperSoftware or MegaSoftware entered a new phase, such as a new platform release or a new version release. In this way, the builds for each stage of development can be segmented into logical sets facilitating a more manageable and organized workflow.

Editing Build Classes

1. On the Build Classes page, click the **Edit Build Class** link in the Actions column.

or

On the Build Class Details page, click **Edit Build Class**.

This opens an Edit Build Class page with the fields populated with the information you used to create the build class.

2. Re-type or re-select the information you want to change.

See [Build Classes- Create or Edit a Build Class](#) for field definitions.

3. Click **OK** after you finish making changes.

Deleting Build Classes

Deleting a Single Build Class

On the Build Classes page, click **Delete Build Class** in the Actions column to delete the build class in that row.

On the Build Class Details page, click **Delete Build Class** to delete that displayed build class.

Deleting Multiple Build Classes

1. On the Build Classes page, select the checkboxes in the first column for the build classes you want to delete.
2. Click **Delete Build Classes**.

Agent Policies

Use this page to set up and manage an ElectricAccelerator cluster by choosing an agent allocation policy, the cluster's preemption policy, the agent lock interval, and deep/wide allocation policy.

See [Understanding a Cluster Sharing Policy](#) for information about cluster sharing policies.

Agent Allocation Policy

Before selecting the agent allocation policy, consider these choices:

- **Exclusive**—(Default) All agents on a specific machine are assigned to the same build.
- **Shared**—Agents on the same machine can be assigned to different builds. This policy requires that eMake client and agent machines have synchronized clocks. You must choose this policy if using Priority Pools.

Preemption Policy

The preemption policy determines how the allocation algorithm responds to requests to preempt agents. To avoid wasted work, a currently-running build may “lock” agents that have been working on the same command for a specified amount of time. The allocation algorithm does not reassign locked agents. In some circumstances the allocation algorithm can reassign “unlocked” agents to balance the load.

Reassigning agents can end in two results, and these two possible results are what determine if agents can be reassigned.

Preemption Policy Field Descriptions

Setting	Description
Always	If reassigning an agent would drop a build below its minimum number of agents, agents cannot be taken . If reassigning an agent would leave a build with at least its minimum number of agents, unlocked agents can always be taken .
Priority	If reassigning an agent drops a build below its minimum number of agents, agents can be taken from a lower priority build in order to bring a higher priority build up to its minimum number of agents. If reassigning an agent leaves a build with at least its minimum number of agents, unlocked agents can be taken from a lower priority build or an equivalent priority build.
Never	Agents can never be taken .

Note: The build class priority and minimum and maximum agent settings are set on the New Build Class page or the Edit Build Class page.

The following scenarios illustrate how the preemption policy setting affects agent reassignment:

Preemption Policy Scenario 1:

- Build A currently has 1 agent and its minimum number of agents is 2.
- Build A needs 1 agent from Build B to meet Build A's minimum number of agents.

- Build B currently has 2 agents and its minimum number of agents is 2.
- If 1 agent is reassigned to Build A, then Build B falls below its minimum number of agents.

Use the following matrix to determine if the agent is reassigned to Build A from Build B.

Preemption Policy Setting	Reassign the agent to A from B?
Always	No
Priority	Yes, if Build A is higher priority No, if Build B is higher priority No, if priority is equal
Never	No

Note: Whether agents are locked or unlocked is disregarded in this scenario because reassigning an agent from Build B results in that build falling below its minimum number of agents.

Preemption Policy Scenario 2:

- Build A currently has at least its minimum number of agents; or Build A needs 1 agent from Build B to meet Build A's minimum number of agents.
- Build B currently has 3 agents and its minimum number of agents is 2.
- If 1 agent is reassigned to Build A, then Build B still has its minimum number of agents.

For this scenario, determining whether to reassign agents depends on Preemption Policy settings and whether agents are locked or unlocked.

The default Agent Lock Interval is 60 (seconds). This means agents are locked after 60 seconds; until 60 seconds elapse, the agents are unlocked. Setting the Agent Lock Interval to 0 means the agents remain unlocked indefinitely.

Preemption Policy Setting	Reassign the agent to A from B?	
	Unlocked or Lock Interval = 0	Lock Interval > 0 and time elapsed exceeds interval
Always	Yes	No
Priority	Yes, if Build A is higher priority No, if Build B is higher priority Yes, if priority is equal	No
Never	No	No

Agent Lock Interval (seconds)

The value indicates when an agent is locked. Locked agents cannot be taken by another build.

Deep/Wide Agent Allocation Policy

Indicates whether the agent allocation policy is set to deep or wide. Deep means the agent allocation algorithm favors assigning more agents on the same host to a build. Wide means the algorithm favors assigning more agents from different hosts.

By default, this setting is deep. If you change this setting to wide, be sure the agent allocation policy is set to "shared."

Click **OK** to save all selections.

Understanding a Cluster Sharing Policy

All builds continually adjust the number of agents they can use. Cluster sharing requires constant cooperation between Electric Make and the Cluster Manager. Each Electric Make machine sends a message to the Cluster Manager whenever it wants more agents.

Cluster sharing allows more than one build to run on a cluster by dynamically reallocating agents based on the cluster sharing policy set up in Cluster Manager. The policy is based on an ElectricAccelerator *fair-sharing* algorithm.

When you create a cluster sharing policy, in addition to the total number of Agents in the cluster, you should consider the following information before setting limits on the minimum number of agents for low, normal, and high priority builds:

- Same priority builds with the same boost share the cluster equally.
- Higher priority class builds can take Agents away from lower priority class builds.
- A build cannot lose an agent if losing an agent will cause the build to go below its minimum (unless preemption is set to "priority"). Builds that cannot get a minimum number of Agents must wait (where minimum is the lower of these values: the build class's MinAgents and the build's current MaxAgents).

The cluster sharing algorithm assigns agents to builds by reading the values for:

- Maximum number of Agents the build is capable of using (as specified by Electric Make the last time it made an Agent allocation request).
- BuildID —Unique identifier for each build that sorts builds by age within each priority.
- Priority —High, normal, or low.
- Boost within each priority.

Any leftover or unused Agents are distributed evenly based on priority and *neededAgents* determination of running builds.

Modifying the Number of Agents per Host

On Linux and Solaris platforms, follow these steps:

1. Locate the `ecagent.conf` file.
 - Linux: `/etc/sysconfig/ecagent.conf`
 - Solaris: `/etc/ecagent.conf`
2. Change the value of the `AGENT_NUMBER` to the number of Agents you wish to run.

For example, `AGENT_NUMBER=2`

On Windows platforms, use the `ecconfig` utility (installed in `C:\ECloud\i686_win32\bin`) to change the number of Agents.

For example, `ecconfig -numagents 2`

Enabling Agents

A disabled Agent, indicated by an empty checkbox, indicates the Agent was taken offline from the cluster.

Enabling a Single Agent

On the Agents page, click the checkbox in the Enabled column of the Agent list to enable the Agent in that row.

On the Agent Details page, click **Enable Agent** to enable that displayed Agent.

Enabling a Group of Agents

1. Go to the Filter menu and select **Disabled Agents** or another filter.
2. Click **Run Filter**.
3. Click the **Enabled Filtered Agents** link to enable the selected Agents.
4. From the Filter menu, select **none** to return to the Agent list.

Disabling Agents

A check mark in the Enabled column indicates an Agent *is* enabled, meaning it is ready to work for a build, using the cluster.

Disabling a Single Agent

On the Agents page, click the checkbox in the Enabled column of the Agent list to disable the Agent in that row.

On the Agent Details page, click **Disable Agent** to disable that displayed Agent.

Disabling a Group of Agents

1. Go to the Filter menu and select **Enabled Agents** or another filter.
2. Click **Run Filter**.
3. Click the **Disable Filtered Agents** link.
4. From the Filter menu, select **none** to return to the Agent list.

Testing Agents

On the [Agents](#) page, for each agent in the Agent list, a green check mark in the Status column indicates an agent is alive; an error icon (a white dash against a red background) indicates the agent is not responding or has timed out.

At any time, you can test one or all agents to update the status list.

Checking the Status of a Single Agent

Click the **[Test]** link in the Status column.

Click **Test Agent** on the Agent Details page to test that displayed Agent.

Checking the Status of a Group of Agents

1. From the Filter drop-down menu, select your desired agent criteria.
2. Click the **Test Filtered Agents** link.

After a few seconds, a green check mark in the Status column indicates an agent is alive. The error icon indicates the agent is not responding or has timed-out.

3. If necessary, click the **View Messages** link in the Action column to help you determine what the error icon means.
4. Click the Agent tab to return to the Agent list.

Deleting Agents

Deleting agents can mean simply deleting an agent from the Agent list on the [Agents](#) page, or it could mean removing an agent from the cluster permanently.

Unless agent hardware was physically disconnected, or the agent service was stopped, agents reappear in the agent list (because connected agents are automatically detected).

Deleting a Single Agent

On the Agents page, click the **Delete Agent** link in the Action column for the agent you want to delete. This action removes the agent from the database.

On the Agent Details page, click **Delete Agent** to delete that displayed Agent.

Deleting a Group of Agents from the Agent List

1. Go to the Filter drop-down menu and select the desired filter.
2. Click **Run Filter**.
3. Click **Delete Filtered Agents**.
4. From the Filter menu, select none, then click **Run Filter**.

This removes the agents from the database.

Removing an Agent from the Cluster Permanently

You must do one of the following:

- Uninstall the agent software
- Point the agent to a different Cluster Manager, or
- Disable the agent service or daemon [by issuing `ecagent_stop` from the agent host] before deleting it from the Cluster Manager agent list.

Creating Resources

1. On the Resources page, click **New Resource**.

This takes you to an empty resource details page.

2. Provide a unique name for your resource definition. The name can contain numbers and letters.

This is also the name that you specify in the `--emake-resource=resourceName` command or in the build class's **Resource Request** field [on the web interface].

If you use Priority Pools, you must use this form: `__pool_xxx`, for example, `__pool_a`

3. Add a description if you wish.

4. Click **Add Host Mask Pattern**.

Host name masks follow the same standard globbing rules as Make. ("Globbing" allows you to use a pattern to match one or more files.) Examples are below.

5. Click **OK** when all fields are filled-in.

When you return to the Build Classes page, you will see the new build class in the table.

Host Name and Glob-Style Pattern Example 1

Resource Definition:

Name—"Xcard"
Hosts—"earth;wind;fire"

eMake execution:

`--emake-resource=Xcard`

Note: If you use the `--emake-resource` option, it overrides the resource request defined in the `--emake-class` option if both options exist.

With this specification, the build will run with agents on hosts *earth*, *wind*, or *fire* only. If these hosts are not available, the build waits until they are. If other builds are running on these hosts, ElectricAccelerator may move those builds to other hosts to free up agents for this build.

Host Name and Glob-Style Pattern Example 2

Resource Definition:

Name—"Xsoft"
Hosts—"earth;asteroid*"

Build Class Definition:

Name—"Xbuilds"
Resource Request—"Xsoft"

eMake execution:

`--emake-class=Xbuilds`

Note: If you use the `--emake-resource` option, it overrides the resource request defined in the `--emake-class` option if both options exist.

With this specification, the build runs on host "earth" only or hosts whose names start with "asteroid."

Editing Resources

1. On the Resources page, click the **Edit Resource** link in the Actions column.

or

On the Resource Details page, click **Edit Resource**.

This opens a Resource Details page with the fields populated with the information you used to create the resource.

2. Re-type or re-select the information you want to change.

See [Resources - Create or Edit a Resource](#) for field definitions.

3. Click **OK** after you finish making changes.

Deleting Resources

Deleting a Single Resource

On the Resources page, click the **Delete Resource** link in the Action column for the resource you want to delete.

On the Resource Details page, click **Delete Resource** to delete that displayed resource.

Deleting Multiple Resources

1. On the Resources page, select the checkboxes in the first column for the resources you want to delete.
2. Click **Delete Resources**.

Priority Pools

Priority Pools allow you to group resources into pools that can be prioritized differently among groups. Each pool's resources can potentially be utilized by any build, but builds originating from a pool's "owner" always have first priority to use that pool's resources. Using Priority Pools allows you to manage resource allocation for builds more efficiently.

Using Priority Pools

Follow this procedure:

1. Enable Priority Pools through either the Cluster Manager or `cmtool`.
Using the Cluster Manager, select Priority Pools on the Server Settings page.
Using `cmtool`, run this command:

```
cmtool modifyServer --resourceManagerType prioritypool
```
2. Define your pool using the Agents > Resources page. A pool resource uses this form: `__pool_xxx`, for example, `__pool_a`
Note: When you define a pool on the Resources page, you must include `__pool_`. When you include a pool in the `--emake-resource` option, you can omit `__pool_` from the option.
3. Go to Agent > Agent Policies and set Agent allocation policy to shared.
4. Define a build class that uses the resource. Go to Builds > Build Classes > New Build Class or Edit Build Class > Resource Request field.
5. Launch eMake and include the following option:

```
--emake-resource="<pool resource>:<static resource>"
```


Information about values for `--emake-resource=`
 - When adding a pool resource to the option, you can omit `__pool_` from `__pool_xxx:` and use `xxx:` only.
 - A pool name (before the `:`) is not required. Not defining the pool name means the build will not use a pool resource.
 - A resource name (after the `:`) is not required. Not defining the resource name means the build will attempt to use any unused resources that it is allowed to use.

Note: When attempting to use a pool resource as a normal resource, there is a period (default is 60 seconds) during which, before your build starts, a build from the pool resource's owner can take back the resource.

Use Case 1—High Performance Builds

You have two pools of resources, one for high performance builds and one for low performance builds. You want to ensure that high performance builds can always use the more powerful 8-core machines and that low performance builds use the 4-core machines. You also want to allow high performance builds to use the 4-core machines when low performance builds are not running. And you want to allow low performance builds to use the 8-core machines when high performance builds are not running. You also have two special software packages, so you define a static resource for each.

Pool makeup:

- Pool a—High performance build resources, five 8-core machines, defined on the Resources page as `__pool_a`
The machines are named `h_1`, `h_2`, `h_3`, `s_1`, and `s_2`.

- Pool b—Low performance build resources, five 4-core machines, defined on the Resources page as `__pool_b`. The machines are named `h_4`, `m_1`, `m_2`, `s_3`, and `s_4`.
- Two static resources are also defined on the Resources page:
 - `s`—This resource includes these machines (which have a specific software package): `s_1`, `s_2`, `s_3`, and `s_4` (from using host mask `s*`)
 - `m`—This resource includes these machines (which have a specific software package): `m_1` and `m_2` (from using host mask `m_1`, `m_2`)

Launching eMake with <code>--emake-resource=</code>	means the build uses
<code>"a:"</code>	Pool "a" and any unused resources
<code>"a:s"</code>	Static resource "s" (four machines). If a low performance build is running, only machines "s_1" and "s_2" (two machines) are used.
<code>"b:m"</code>	Static resource "m" only
<code>"a:m"</code>	Static resource "m" only. If a low performance build is running, no machines are available and the build must wait.
<code>":s"</code> or <code>"s"</code>	Static resource "s" if low and high performance builds are not running. If a low performance build is running, then only machines "s_1" and "s_2" are used. If a high performance build is running, then only machines "s_3" and "s_4" are used. If low and high performance builds are running, no machines are available and the build must wait.
<code>": "</code> or <code>""</code>	Any unused resources (this is considered a common build)

Use Case 2—Multiple Departments

There are two departments and each department has its own pool of machines. They want to contribute their machines toward a common pool so each department can use the other department's machines while still ensuring that their own machines are available for their department's builds. One department owns 20 machines and the other department owns 10 machines. Each department also owns a small number of 64-bit machines. IT contributes an additional four machines that any department can use.

Pool makeup:

- Pool Depta—Department A's resources, 20 machines, defined on the Resources page as `__pool_Depta`. The machines are named `a_01` through `a_16` and `a_17_64` through `a_20_64` (these last four machines are 64-bit).

- Pool `Deptb`—Department B's resources, 10 machines, defined on the Resources page as `__pool_Deptb`. The machines are named `b_01` through `b_08` and `b_09_64` and `b_10_64` (these last two hosts are 64-bit).
- Static resource `64bit`—This resource includes these machines: `a_17_64` through `a_20_64` and `b_09_64` and `b_10_64` (from using host mask `*_64`).
- General build machines—IT department-supplied general build machines, four machines

Launching eMake with <code>--emake-resource=</code>	means the build uses
<code>"Deptb:"</code>	Pool "Deptb" and any unused resources, using general build machines first and then Department A's machines if that department is not running a build.
<code>"Depta:"</code>	Pool "Depta" and any unused resources, using general build machines first and then Department B's machines if that department is not running a build.
<code>"Depta:64bit"</code>	Static resource "64bit" (six machines). If Department B is running a build, only "a_17_64" through "a_20_64" (four machines) are used.
<code>": " or ""</code>	Any unused resource, using general build machines first and then Department A's and Department B's machines if those departments are not running builds (this is considered a common build).

Creating Filters

1. On the Filters page, click **New Filter**.

or

From any Filter panel, click the **New Filter** button.

2. Name the filter.
3. Select the table where this filter will run.
4. Select the Global checkbox if you want the filter to be available to other users.
5. Select the SQL order by clause for the associated table.
6. Supply the SQL query you want to use.

The query value is a SQL WHERE clause.

7. Click **OK** when finished.

If the syntax is incorrect, a Cluster Manager page may return SQL errors. In general, the Cluster Manager resets the current filter choice if this happens, so a refresh should resolve this issue (this means the filter can be fixed). However, if this does not work, the Home page contains a **Reset Session** link to reset all current session variables.

Editing Filters

1. On the Filters page, click the **Edit Filter** link in the Actions column.

This opens an Edit Filter page with the fields populated with the information you used to create the filter.

2. Re-type or re-select the information you want to change.

See [Filters - Create or Edit a Filter](#) for field definitions.

3. Click **OK** after you finish making changes.

Deleting Filters

Deleting a Single Filter

On the Filter page, click **Delete Filter** in the Actions column to delete the filter in that row.

Deleting Multiple Filters

1. On the Filters page, select the checkboxes in the first column for the filters you want to delete.
2. Click **Delete Filters**.

Generating Reports

1. Select the report from the drop-down menu you would like to generate.
2. Complete the From/To fields for the report dates.

Click the calendar icon adjacent to each field to select the dates.

3. Fill-in any additional fields to customize the report.
4. Customize the Ranges field if desired.

If available, the Ranges field allows you to customize which ranges the report displays. Use this format for the Ranges field: *rangename,[beginning of range,end of range),rangename,[beginning of range,end of range)*, and so on. A [or] bracket denotes that the number is inclusive. A (or) denotes that the number is exclusive.

See individual [report descriptions](#) for additional information.

5. Click **Run Report** to generate the report.

Adding Custom Reports

You can add custom reports to the Build Details page. Typically, the reports access the uploaded annotation file to present analysis of various aspects of the build. You must write a script/application that takes three arguments and produces an html/text output file.

The arguments are (in this order):

1. Script name (the base name of the script)
2. Cluster Manager build ID
3. Directory where the annotation file is located. The filename with annotation is always `build.xml`. This is where the script writes its output files.

Generally, the script reads the `build.xml` file and creates a number of files. The first file it must create is the temporary html file. That file is named after the provided script name, followed by `.html_tmp`. This is where the html output is created. After the file is done, it is moved under the script name followed by `.html`.

If the file produces downloadable output (such as plain text), it produces a file named after the script name followed by `.out_tmp` (the temporary file where the file is created). After the file is created, it is moved under the script name followed by `.out`.

It is important that the script is well-behaved if failure occurs (for example broken XML). In this case, it must remove the temporary files so the Cluster Manager does not continue to show the *Generating...* state. Electric Cloud recommends generating an html output file (following the naming scheme above) that states the error, so users can learn of a failure.

Recommendation: Start generated output with a table element, which then may contain whatever is desired. To pick up common stylesheet settings, make the table and its cells of the "mainWindow" class.

Making Custom Reports Available to the Cluster Manager

To make the report available:

1. Install the script/application in the `<ECloud install>/<arch>/bin` directory.
2. Run the application `ecaddcmreport` with the following arguments (in this order):
 - a. `shortname` (the name used internally to identify the report)
 - b. `label` (the text to display in the UI to represent the report)
 - c. `commandline` (the full path to the executable)
 - d. `wait flag` (1 means the UI stalls the request to wait for the result file and 0 means the request returns immediately)

Known Issues

Due to an issue using wrapped Tcl applications from Windows Apache, `tcsh` is invoked with the script as the argument.

The Home Page

The **Home** page and all other pages in the Cluster Manager web UI provide tab-based access to administration functions as well as information about builds, agents, reports, and messages.

The Top Bar

The top bar provides the following links:

- **Logout**—Logs out the current user
- **About**—Displays which ElectricAccelerator version is running on your server
- **Help**—Opens page-specific (context-sensitive) HTML help throughout the web UI. The help is loaded from the Electric Cloud documentation website.

The web UI also provides overview and concept help topics that are not context-sensitive (meaning that they are not linked to a specific page in the web UI). To see these help topics, click **Help** from any page in the web UI. The left-pane (table of contents) contains the list of help topics.

Server Information

The Home page provides the following information:

- Number of currently running builds
- Number of active, inactive, or disabled agents
- Resource manager type
- Database type

Comments

New Comment lets you add a new server comment. For details, see [Using Comments](#).

Builds

The Builds page allows you to see:

- running and completed builds
- additional build information
- whether completed builds succeeded or failed










Functionality

- Clicking the name of any particular build takes you to that build's [Build Details](#) page, which provides much more detailed information about the build, including access to the Build Log.

Note: If you wish to configure the Builds page so users can see their own builds only, see the [Permissions](#) topic.

- After [filtering builds](#), you can then [stop](#) or [delete](#) your filtered builds.
- The Actions column allows you to view build-related messages and [delete](#) or [stop](#) the build in that row.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.
 - Set the number of records you want to see per page.

Column Descriptions

Column	Description
Result	<p>Indicates the build's result or status.</p> <ul style="list-style-type: none">  indicates a successful, completed build  indicates a build error/failure  indicates a 'stopped' build  indicates a running build  indicates a local running build, running for an unusual amount of time  indicates a build waiting on a breakpoint
Agents	<p>Displays agent availability (using a percentage bar) during a running build. On mouse over, Requested indicates the number of separate build steps that eMake identified and requested agents for.</p>
Priority	<p>Indicates the build's priority for acquiring agents when the build runs.</p> <ul style="list-style-type: none">  High priority (3 green bars)  Normal priority (2 yellow and 1 white bar)  Low priority (1 blue and 2 white bars)

Column	Description
ID	Indicates the build ID number assigned by the Cluster Manager.
Name	This is the build name as defined by the build class tag definition.
Class	This build belongs to the build class whose name is displayed. A default build class name is displayed if one was not assigned.
User@Machine	Displays the user who started the build and the machine where the build was invoked (mouse over the field to display the IP address).
Start Time	Displays the date and time the build was started.
Duration	Displays the amount of time the build took to run-elapsed time between the start and end time.
Actions	<p>Provides two links:</p> <ul style="list-style-type: none">• View Messages—Takes you to the Messages page for the specific build ID where you can set up a filter or delete filtered messages. and either• Delete Build—Deletes the corresponding build. This links is available if the build is not running. <i>or</i>• Stop Build—Stops the corresponding build. This link is available if build is running.

Build Details

This page displays additional build properties, including the full command-line, environment, and Electric Make version for the build shown at the top of the page.

Functionality

- **View Message**—Takes you to the [Messages](#) page.
- **Download Build Log**—Allows you to download the build's annotation file.

For more information about the *annotation* file, see the "Annotation" chapter in the *Electric Make Users Guide*.

- **Delete Build**—[Deletes](#) the build on this page.

Details

On the left-side of the page, a pull-down allows you to select Details, Configuration, Log Output, or Metric information for your build.

- **Details**—This selection displays summary information about the build and the machine that invoked the build.

Note: If a break point is active and was triggered, you will see this information here also. For more information on using ecbreakpoint, go to [Using Breakpoints](#).

- **Workload**—Sum of agent usage over all agents (in seconds). This value also appears in the annotation file.
- **Command Line**—The command line used to invoke eMake
- **CWD**—The directory from which eMake ran
- **EMake Version**—Electric Make version information
- **Build Log Directory**—The location of the build log
- **History File**—The full path to the history file that was input into the build
- **Configuration**—This selection includes several build properties and displays build environment variables.
- **AnnoDetail**—Lists the annotation options used for the build. These options may have been specified on the eMake command-line or they might be inherited from the build class.
- **Log Output**—This selection provides the last 100 lines of log output (extracted from the annotation file).
- **Metrics**—This selection includes more than 50 metrics characterizing different aspects of build performance.

Note: To see metric definitions, refer to the Metric Values in Annotation Files section in the "Annotation" chapter of the *Electric Make Users Guide*.

Comments

New Comment allows you to add a new build comment as described in [Using Comments](#).

Build Classes

After you create build classes, a list of those classes and their parameters appears on this page.

About Build Classes

A *build class* is a flexible, user-defined classification for a designated group of builds. Using build classes is optional, but if you do not assign a build class, Cluster Manager assigns the build to a default class. Electric Cloud recommends using build classes to provide organization to the build management process.

Depending on your company requirements, you might use build classes to organize build groups by version/release, product type, development stage, or platform. You can decide how to use build classes to organize your builds into sets.

When a build is invoked in Electric Make, the Electric Make option `--emake-class=<class>` specifies the class assignment for the current build. To use the Electric Make class option, the class must match an existing class in Cluster Manager.

Functionality

- Clicking a build class name takes you to the build class' [Build Class Details](#) page.
- You can [create](#) a new build class or [delete](#) existing build classes.
- Selecting **Notify** enables email notification for builds (belonging to that class) that complete.
- The Actions column allows you to [edit](#) or [delete](#) the build class in that row.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.
 - Set the number of records you want to see per page.

Column Descriptions

Column	Description
Name	Name of the build class.
Tag Definition	<p>Tag definition for this build, entered on the Build Class Details page. Together, the build name and variables are referred to as the <i>tag definition</i>. Variable names are case-sensitive.</p> <p>For example, the tag definition <code>%BUILD_CLASS%_%LC%_%DATE%</code> for a build class named <code>QA_BUILD</code> creates the following build tag:</p> <pre>QA_BUILD_1234_20060123185958</pre>
Resource	Resource that the build class uses.
Platform	Operating system that the build class uses/supports.

Column	Description
Priority	Class' build priority level. Classes have default priorities and boost values. Boost values range from -10 to +10 (default 0) where a higher boost value means that builds in that class can use available agents ahead of builds with the same priority and lower boost. Read about this setting's role in agent policies .
Limits	Minimum (default 2) and maximum (default 64) number of agents the build class will use for the build. Read about this setting's role in agent policies .
Notify	Enables email notification (based on the build class) for builds that complete. In the list of classes, select the checkbox in the Notify column for each class for which you want to receive email notifications. When a build from the selected class(es) completes, an email message is sent to each user that requested notification—each user (individually) must select the classes for which they want to receive notifications. Changing Notify settings requires Modify permission defined in the User field on the Permissions page. The administrator can edit a user's permission on the Administration > Permissions page.
Actions	Edit Build Class —Takes you to the corresponding Edit Build Class page. Your existing build class specifications populate the appropriate fields. Delete Build Class —Deletes the corresponding build class.

Build Class Details

This page displays additional information for the build class shown at the top of the page.

Functionality

- **Delete Build Class**—[Deletes](#) the build class displayed on this page.
- **Edit Build Class**—Takes you to the [Edit Build Class](#) page.

Details

This section displays all settings for the build class named at the top of the page. See [Build Classes- Create or Edit a Build Class](#) for definitions.

Comments

New Comment allows you to add a new build class comment as described in [Using Comments](#).

Build Classes- Create or Edit a Build Class

See [Creating Build Classes](#) for the procedure for creating a build class. See [Editing Build Classes](#) for the procedure for editing a build class.

Field Descriptions

Field	Description
Name	Name of the build class.

Field	Description
Tag Definition	<p>Tag that generally consists of a generic build name appended with build-specific data that you construct from the following variables:</p> <p>GC—Globally unique number (Global Counter)</p> <p>LC—Number unique to the build class (Local Counter; the build serial number within the class)</p> <p>BUILD_CLASS—User-defined build class name</p> <p>BUILD_CLASS_ID—System-generated number that the Cluster Manager uses to identify each class</p> <p>USER_NAME—Name of the user who invoked eMake</p> <p>MACHINE_NAME—Name of the machine where eMake was invoked</p> <p>USER_BUILD_LABEL—Label specified at the eMake command line. For example, <code>--emake-build-label=my_build</code></p> <p>BUILD_OS_ID—Operating system ID under which the build was invoked (0 = undefined, 1 = Windows, 2 = Solaris, and 3 = Linux)</p> <p>DATE—Build start date and time using variables Y, y, m, d, H, M, and S (for example, 2005-01-18 10:14:32 is 20050118101432)</p> <p>Y—Year at build start time (YYYY)</p> <p>y—Year at build start time (YY)</p> <p>m—Sequential month number at build start time (1-12)</p> <p>d—Sequential day of month at build start time (1-31)</p> <p>H—Hour of the day at build start time (0-23)</p> <p>M—Minutes at build start time (0-59)</p> <p>S—Seconds at build start time (0-60)</p> <p>a—Abbreviated day of week at build start time (WED)</p> <p>A—Full name day of week at build start time (Wednesday)</p> <p>b—Abbreviated month name at build start time (AUG)</p> <p>B—Full month name at build start time (August)</p> <p>c—Build start date and time using the variables A, B, d, H, M, S, and Y (for example, 2005-01-18 10:14:32 means 18/01/05 10:14:32)</p> <p>For more information about user-defined variables, see the “Electric Make Command-Line Options, Environment Variables, and Configuration File” section in Chapter 3, <i>Electric Make Basics</i>, of the <i>ElectricAccelerator Electric Make User Guide</i>. For information about constructing tag definitions, see the “Tag Definitions” section in Chapter 4, <i>Additional Electric Make Settings and Features</i>, of the <i>ElectricAccelerator Electric Make User Guide</i>.</p>
Minimum Agents	Minimum number of agents that must be available before a build is assigned agents. The default is 2 agents. For more information, see agent policies .
Maximum Agents	Maximum number of agents that can be assigned to a build. The default is 64 agents. For more information, see agent policies .

Field	Description
Priority	Priority for builds in this build class. Select Low, Normal, or High, and if using a boost value, use the adjacent pull-down to select a boost value. For more information, see agent policies .
Platform	Operating system to use for this build class.
Resource Request	Field used to request particular agents for this build class. (If you have not created resources, or if you need to create additional resources, click the Agents tab, then click the Resources subtab, and then from the Resources page, click the New Resource link to define resources.)
Allow jobcache	<p>Checkbox that determines whether to allow the JobCache feature to be used for this build class. For the default build class and for build classes that exist when you upgrade to Accelerator 8.0, this checkbox is checked (meaning that JobCache is allowed) by default. For a build class that you create, this checkbox is unchecked by default.</p> <p>JobCache is an add-on option. It is available only on Linux and Solaris platforms and requires separately-purchased licensing. For more information about JobCache, see the <i>ElectricAccelerator Electric Make User Guide</i>.</p>
Annotation Upload	Checkbox that determines whether to upload the annotation file to the Cluster Manager.
Annotation Levels	<p>Checkboxes that indicate which levels of information to include in the annotation file. Available annotation levels are as follows:</p> <ul style="list-style-type: none"> • Basic Annotation (if the JobCache add-on option is enabled, basic annotation includes JobCache annotation) • Env (extended environment information) • History (serialization details) • File (files read or written) • Lookup (all file names accessed) • Waiting (complete dependency graph) • Registry (registry updates for Windows)

eMake automatically creates makefile macros (`ECLLOUD_BUILD_CLASS` and `ECLLOUD_BUILD_TAG`) from Cluster Manager build class data. You can use these macros to add generated values to your makefiles. For more information, see “Using Electric Make Variables” in the *ElectricAccelerator Electric Make User Guide*.

Agents

The Agents page allows you to view all configured agents.




Note: If you do not see an agent you expect to see in the list, check the hardware for proper connection and setup. If the agent still does not appear in the list, check whether or not the agent is assigned to the correct Cluster Manager host name.

Functionality

- Clicking an agent Name in the first column takes you to the [Agent Details](#) page, where more information is available for that specific agent.
- Clicking a host in the second column filters the agent list to agents on that host only.
- Clicking a build in the Build column takes you to that build's [Build Details](#) page.
- Clicking an agent in the Enabled column enables/disables that agent.
- Clicking **[Test]** tests the agent in that row. See [Testing Agents](#).
- After [filtering agents](#), you can then [enable](#), [disable](#), [test](#), or [delete](#) your filtered agents.
- The Actions column allows you to view agent-related messages or [delete](#) the agent in that row.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.
 - Set the number of records you want to see per page.

Column Descriptions

Column	Description
Name	This is the name of the Agent, which is a combination of the host name and a sequential, Cluster Manager assigned number.
Host	This is the DNS host name.
Platform	This is the platform the agent is running.
Agent Version	This is the release number of Agent software installed on the host. Electric Cloud recommends the Agent release number match the Cluster Manager release number.
EFS Version	This is the release number of the EFS version installed on the host.
Build	This is the name of the build the agent is currently servicing.

Column	Description
Enabled	If the agent is enabled and its status is "green," the Cluster Manager can use the agent.
Status	 indicates the agent is running and connected to the Cluster Manager  indicates the agent has a problem connecting to the Cluster Manager  indicates the agent failed and was not used for the build. The build will attempt to re-use the agent for later builds. If a build job causes an agent failure, that agent is placed in "penalty" status for 30 seconds. This prevents the agent from being reassigned to another build job during that time. The intent is to prevent an agent that has a fatal hardware failure from being repeatedly reassigned to build jobs and then causing multiple (successive) failures, which would eventually cause the entire build to fail.
Actions	View Messages —Takes you to the Messages page so you can view all system log entries for the corresponding agent. Delete Agent —Deletes the corresponding agent.

Agent Details

This page displays information for the Agent shown at the top of the page.

Functionality

- **Enable Agent**—[Enables](#) the agent displayed on this page.
- **Disable Agent**—[Disables](#) the agent displayed on this page.
- **Test Agent**—[Tests](#) the Agent displayed on this page.
- **Delete Agent**—[Deletes](#) the Agent displayed on this page.

Status

This section provides information about status checks from the Cluster Manager. If the Cluster Manager is unable to contact the agent, this section provides troubleshooting information.

Configuration

This section provides detailed information about the agent's configuration, including the port number and version information.

Agent Console

This section provides additional information about the agent (from direct communication with agent on the console port), including session state, system information, and file system disk space usage.

Comments

New Comment allows you to add a new agent comment as described in [Using Comments](#).

Agent Policies

Use this page to set up and manage an ElectricAccelerator cluster by choosing an agent allocation policy, the cluster's preemption policy, the agent lock interval, and deep/wide allocation policy.

See [Understanding a Cluster Sharing Policy](#) for information about cluster sharing policies.

Agent Allocation Policy

Before selecting the agent allocation policy, consider these choices:

- **Exclusive**—(Default) All agents on a specific machine are assigned to the same build.
- **Shared**—Agents on the same machine can be assigned to different builds. This policy requires that eMake client and agent machines have synchronized clocks. You must choose this policy if using Priority Pools.

Preemption Policy

The preemption policy determines how the allocation algorithm responds to requests to preempt agents. To avoid wasted work, a currently-running build may “lock” agents that have been working on the same command for a specified amount of time. The allocation algorithm does not reassign locked agents. In some circumstances the allocation algorithm can reassign “unlocked” agents to balance the load.

Reassigning agents can end in two results, and these two possible results are what determine if agents can be reassigned.

Preemption Policy Field Descriptions

Setting	Description
Always	If reassigning an agent would drop a build below its minimum number of agents, agents cannot be taken . If reassigning an agent would leave a build with at least its minimum number of agents, unlocked agents can always be taken .
Priority	If reassigning an agent drops a build below its minimum number of agents, agents can be taken from a lower priority build in order to bring a higher priority build up to its minimum number of agents. If reassigning an agent leaves a build with at least its minimum number of agents, unlocked agents can be taken from a lower priority build or an equivalent priority build.
Never	Agents can never be taken .

Note: The build class priority and minimum and maximum agent settings are set on the New Build Class page or the Edit Build Class page.

The following scenarios illustrate how the preemption policy setting affects agent reassignment:

Preemption Policy Scenario 1:

- Build A currently has 1 agent and its minimum number of agents is 2.
- Build A needs 1 agent from Build B to meet Build A's minimum number of agents.
- Build B currently has 2 agents and its minimum number of agents is 2.
- If 1 agent is reassigned to Build A, then Build B falls below its minimum number of agents.

Use the following matrix to determine if the agent is reassigned to Build A from Build B.

Preemption Policy Setting	Reassign the agent to A from B?
Always	No
Priority	Yes, if Build A is higher priority No, if Build B is higher priority No, if priority is equal
Never	No

Note: Whether agents are locked or unlocked is disregarded in this scenario because reassigning an agent from Build B results in that build falling below its minimum number of agents.

Preemption Policy Scenario 2:

- Build A currently has at least its minimum number of agents; or Build A needs 1 agent from Build B to meet Build A's minimum number of agents.
- Build B currently has 3 agents and its minimum number of agents is 2.
- If 1 agent is reassigned to Build A, then Build B still has its minimum number of agents.

For this scenario, determining whether to reassign agents depends on Preemption Policy settings and whether agents are locked or unlocked.

The default Agent Lock Interval is 60 (seconds). This means agents are locked after 60 seconds; until 60 seconds elapse, the agents are unlocked. Setting the Agent Lock Interval to 0 means the agents remain unlocked indefinitely.

Preemption Policy Setting	Reassign the agent to A from B?	
	Unlocked or Lock Interval = 0	Lock Interval > 0 and time elapsed exceeds interval
Always	Yes	No
Priority	Yes, if Build A is higher priority No, if Build B is higher priority Yes, if priority is equal	No
Never	No	No

Agent Lock Interval (seconds)

The value indicates when an agent is locked. Locked agents cannot be taken by another build.

Deep/Wide Agent Allocation Policy

Indicates whether the agent allocation policy is set to deep or wide. Deep means the agent allocation algorithm favors assigning more agents on the same host to a build. Wide means the algorithm favors assigning more agents from different hosts.

By default, this setting is deep. If you change this setting to wide, be sure the agent allocation policy is set to "shared."

Click **OK** to save all selections.

Resources

This page displays all defined resources.

Functionality

- Clicking a resource name takes you to the resource's [Resource Details](#) page.
- You can [create](#) a new resource or [delete](#) existing resources.
- The Actions column allows you to [edit](#) or [delete](#) the resource in that row.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.
 - Set the number of records you want to see per page.

Column Descriptions

Column	Description
Name	This is the name of the resource. Click a resource name to go to the Resource Details page for that resource.
Hosts	This is a semi-colon delimited list of host name glob-style patterns.
Actions	Edit Resource —Takes you to the corresponding Edit Resource page. Delete Resource —Deletes the corresponding resource.

Resource Details

This page displays additional information for the resource shown at the top of the page.

Functionality

- **Delete Resource**—[Deletes](#) the resource on this page.
- **Edit Resource**—Takes you to the [Edit Resource](#) page.

Details

Resource details for the particular resource you selected are displayed in this section.

Comments

New Comment allows you to add a new resource comment as described in [Using Comments](#).

Resources - Create or Edit a Resource

See [Creating Resources](#) for information about creating a resource, and [Editing Resources](#) for information about editing a resource.

Field Descriptions

A resource definition consists of two parts; the resource name and a list of host names and/or masks.

Field	Description
Name	<p>This is a unique name for your resource definition. The name can contain numbers and letters.</p> <p>This is also the name that you specify in the <code>--emake-resource=resourceName</code> command or in the build class's Resource Request field [on the web interface].</p> <p>If you use Priority Pools, you must use this form: <code>__pool_xxx</code>, for example, <code>__pool_a</code></p>
Hosts	<p>Host names that satisfy the resource name.</p> <p>Host name masks follow the same standard globbing rules as Make. ("Globbing" allows you to use a pattern to match one or more files.)</p> <p>Pattern arguments may contain any of the following special characters:</p> <ul style="list-style-type: none">*—Matches any sequence of zero or more characters.?—Matches any single character.[...]—Matches a set or range of characters. For example, <code>myhost[1-9]</code> matches the hosts 'myhost1' through 'myhost9'\—Escapes the following characters.

Reports

Cluster Manager report results are displayed on this page. These reports provide many types of analytics about your build cluster and use several types of data visualization, such as scatter plots and pie charts. The following reports are available:

- [Agent Usage Report](#)—Stacked area chart that shows the agent license count and agents available versus agent demand in a specified time period
- [Build Duration Report](#)—Scatter plot that shows the durations of builds over a specified time period
- [Build Duration vs. Conflict Report](#)—Scatter plot of build duration versus the number of conflicts per build
- [Build Duration vs. Job Count Report](#)—Scatter plot of build duration versus the number of jobs per build
- [Build Summary Report](#)—Table that shows aggregate statistics (average wait time and total workload) that are grouped by either build class or build name
- [Build Wait Time Report](#)—Scatter plot that shows the wait times for builds over a specified time period
- [Cluster Usage Report](#)—Chart that shows the agent shortage, available agents, and builds running concurrently for a specified time period
- [Cluster Utilization Report](#)—Two stacked, stepped line charts that represent average agent demand and average builds running over a specified time period
- [Concurrent Builds Report](#)—Skyline chart that shows the number of running builds at any time and how many were of each build class or build name pattern as specified
- [Effective Agent Allocation Summary Report](#)—Pie chart that shows the effective agent allocation percentage that is reported by eMake for a specified time period
- [Effective Agent Allocation Per Day Report](#)—Stacked bar chart that shows the effective agent allocation percentage that is reported by eMake per day
- [JobCache License Allocation Report](#)—Skyline chart that shows concurrent builds at any time (stacked by JobCache license request results) to show how many were able to use JobCache
- [JobCache Usage Summary Report](#)—Several pie charts that show job cache requests and their results
- [Resource Usage over Time Report](#)—Chart that shows the number of simultaneous builds that ran on each resource over a specified time period
- [Sea Level Report](#)—Plot of agent requests against a backdrop of projected cluster sizes at or above 100%

Certain reports use statistics about resources. The Cluster Manager keeps resource statistics for 30 days by default. To ensure that results appear in these reports, you must specify only date ranges for the time period for which resource statistics exist.

Note that reports assume that the Cluster Manager and your browser are in the same time zone. The dates assume that time zone.

Checking Your Permissions to Run Reports

You must have “read” permission to run reports. You can verify that you have this permission in the **Reports** column in the **Permissions** subtab under the **Administration** tab. (The Cluster Manager administrator always has permissions to run any report.) Certain reports also require “read” permission for resources. If you do not have the required “read” permissions to run a report, ask your Cluster Manager administrator to add them.

Using Permanent Links to Reports

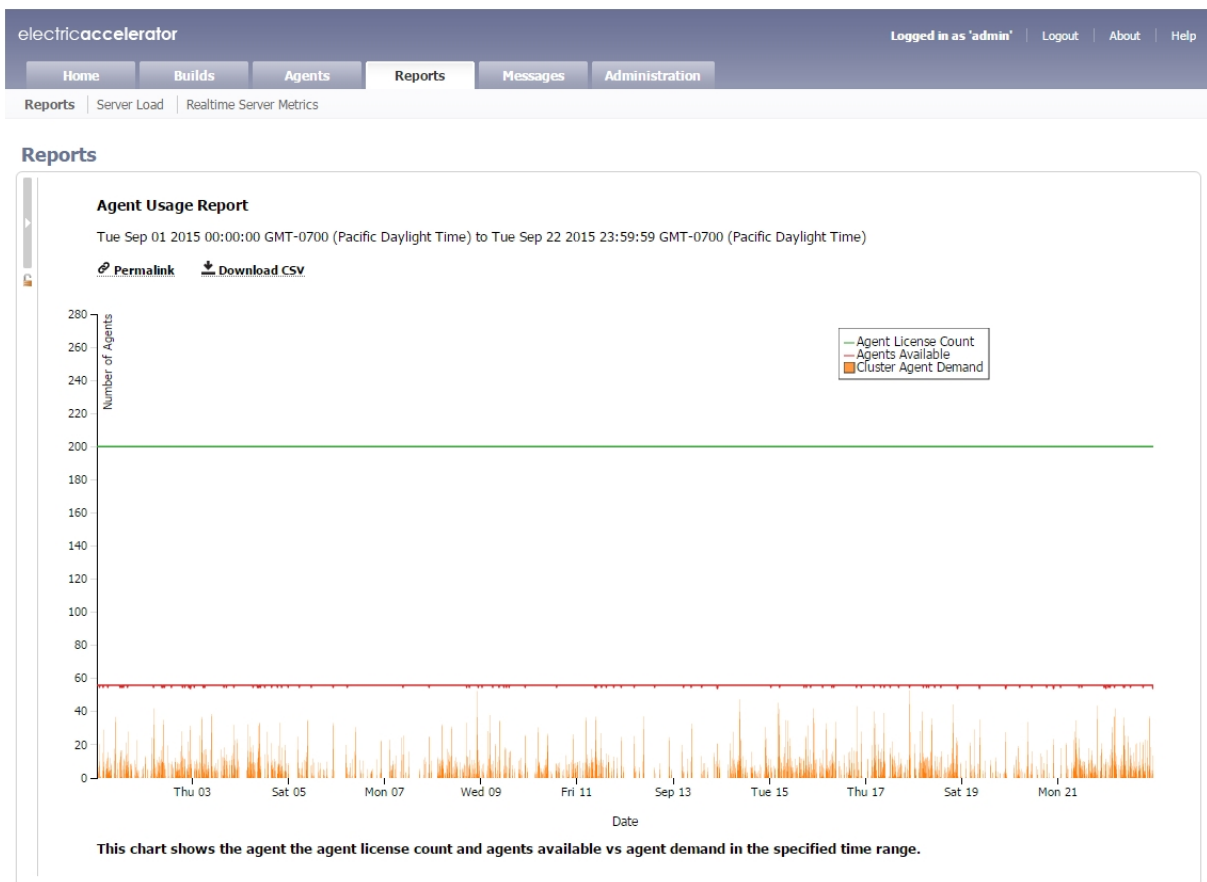
Most reports include a **Permalink** button, which provides permanent, direct access to a report. Click the **Permalink** button to browse to the version of the report with a permanent URL, which you can bookmark in your browser or send to others for convenient viewing.

Downloading Report Data

Most reports include a **Download CSV** button, which downloads report data into a .csv (comma-separated values) file. You can open this file in a spreadsheet application such as Microsoft Excel.

Agent Usage Report

This report is a stacked area chart that shows the agent license count and agents available versus agent demand in a specified time period:



You can input one or more resource names. If no resource name is specified, the cumulative usage for the entire cluster is shown.

The number of agents available should remain fairly constant at the number of agents in the cluster. Discrepancies could result from one of the following: network issues might be causing agents to become unavailable, there might be maintenance performed on agents, or there might be an issue with the agents themselves. A sustained level of agent shortage indicates that the cluster is probably too small.

Build Duration Report

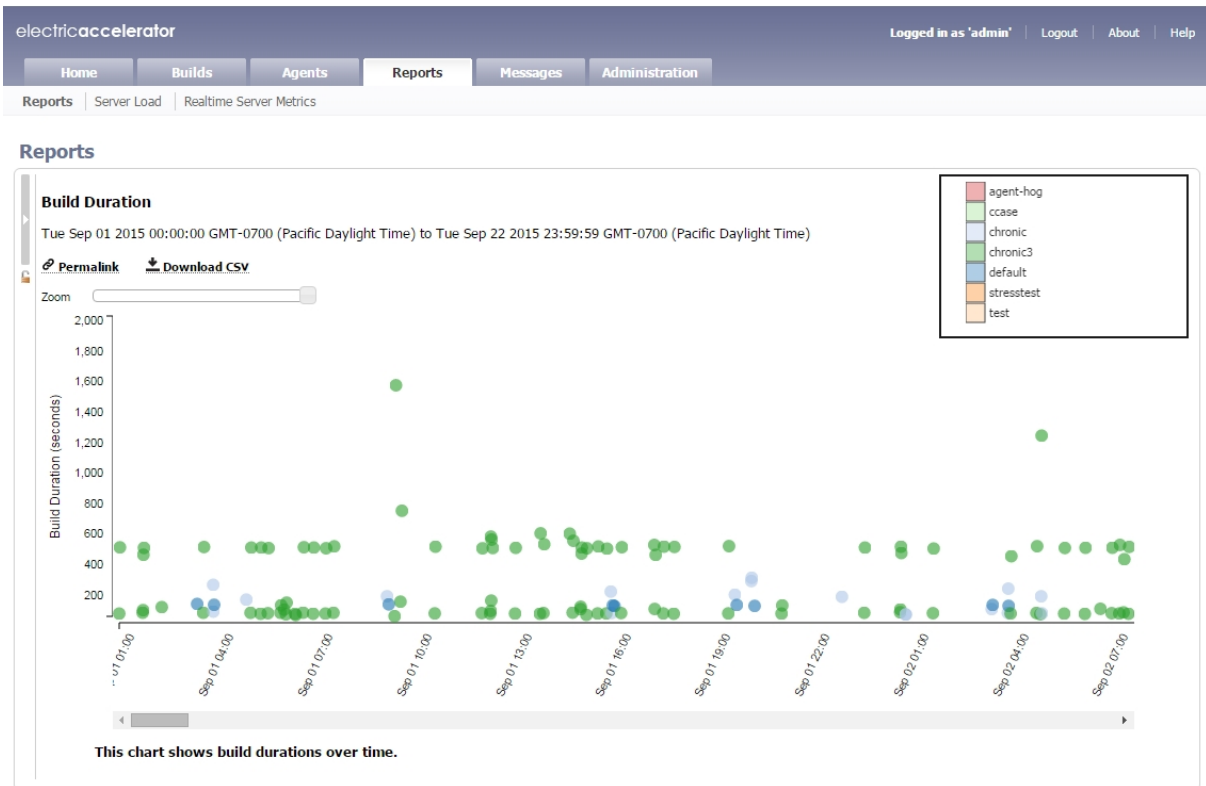
This report is a scatter plot that shows the durations of builds over a specified time period:



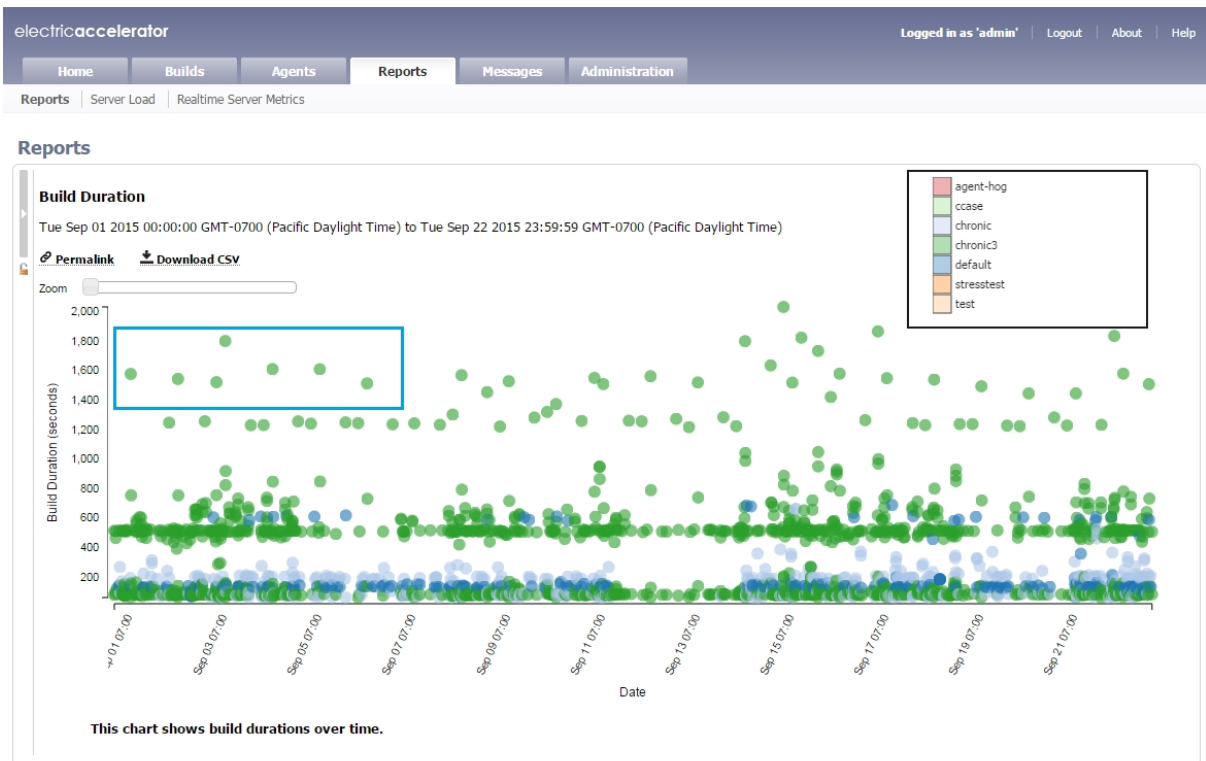
You can input a build class or a build name along with a date range. You can also specify whether you want builds (points) to be colored by build class or build name.

This report is most useful to identify outliers in build or build class wait times, especially when build classes can be used to group similar builds together.

You can scroll or zoom to see a larger range or zoom in on a cluster of builds that are close together:



As with other scatter plots, you can click and drag to select multiple builds to view a table that shows details for those builds:



Following is a table of build details for the set of builds selected above:

electricaccelerator Logged in as 'admin' | Logout | About | Help

Home Builds Agents Reports Messages Administration

Reports Server Load Realtime Server Metrics

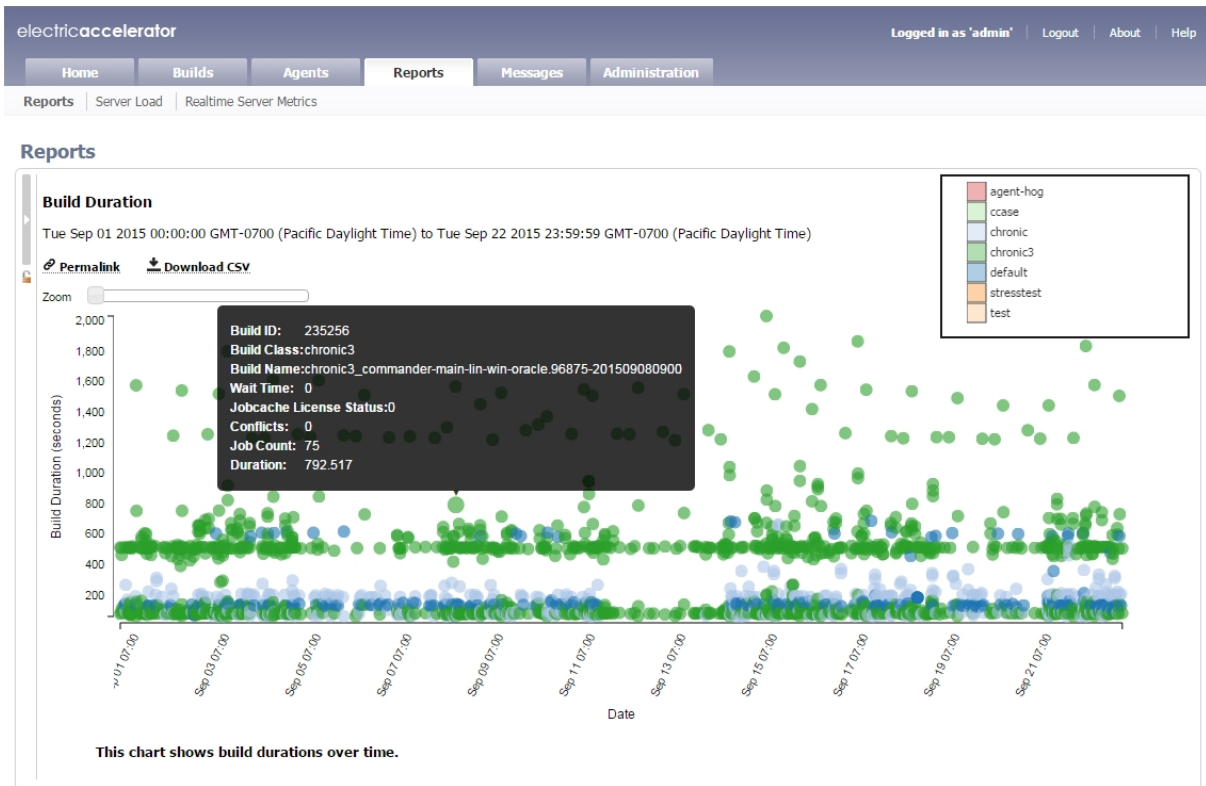
Reports

Selected builds (Back to chart)

[Download CSV](#)

Build ID	Build Class	Duration
234463	chronic3	1577.457
234607	chronic3	1543.306
234760	chronic3	1521.621
234801	chronic3	1800.849
234961	chronic3	1610.21
235074	chronic3	1609.403
235117	chronic3	1513.375

You can hover over a build for details:



You can click a build to navigate to the **Build Details** page for that build:

electricaccelerator

[Home](#)
[Builds](#)
[Agents](#)
[Reports](#)
[Messages](#)
[Administration](#)

[Builds](#)
[Build Classes](#)

Name: chronic3_commander-main-lin-win-oracle.96875-201509080900

Status: OK

Start Time: 2015-09-08 16:14:56 UTC

Duration: 13 m 12 s

Conflicts: 0

[View Messages](#)
[Download Build Log](#)
[Delete Build](#)

Details

Build ID: 235256

Build Class: chronic3

Job Count: 75

Priority:

Min Agents: 2

Max Agents: 30

Effective Agent Alloc: 99.00 %

Wait Time: 0 s

IP set: 0

Workload: 1079.95

Host Name: ecbuild-lin27

IP Address: 10.1.216.206

User Name: build

OS User Name: build

User Label: commander-main-lin-win-oracle.96875-201509080900

Platform: Linux

History Exists: Yes

Command Line: /usr/local/tools/i686_Linux/eccloud/current/bin/emake --emake-cm=linbuild-cm --emake-anno=anno-i686_Linux.xml --emake-class=chronic3 --emake-build-label=commander-main-lin-win-oracle.96875-201509080900 --emake-historyfile=emake-linux.data -k -C nimbus BUILDNUMBER=96875 buildNonServer

Resource Request: commander

CWD: /net/chronic3build/commander-main-lin-win-oracle.96875-201509080900

EMake Version: 6.2.1.47990 32-bit (build_6.2_47990_OPT_2013.01.31_04:41:52)

Build Log Directory: /opt/eccloud/i686_Linux/build_logs/2/3/5/2/5/235256

History File: emake-linux.data

Comments

Comment

Modified By

Last Modified

Actions

Currently, there are no records to display in this list.

New Comment

Build Duration vs. Conflict Report

This report is a scatter plot of build duration versus the number of conflicts per build:

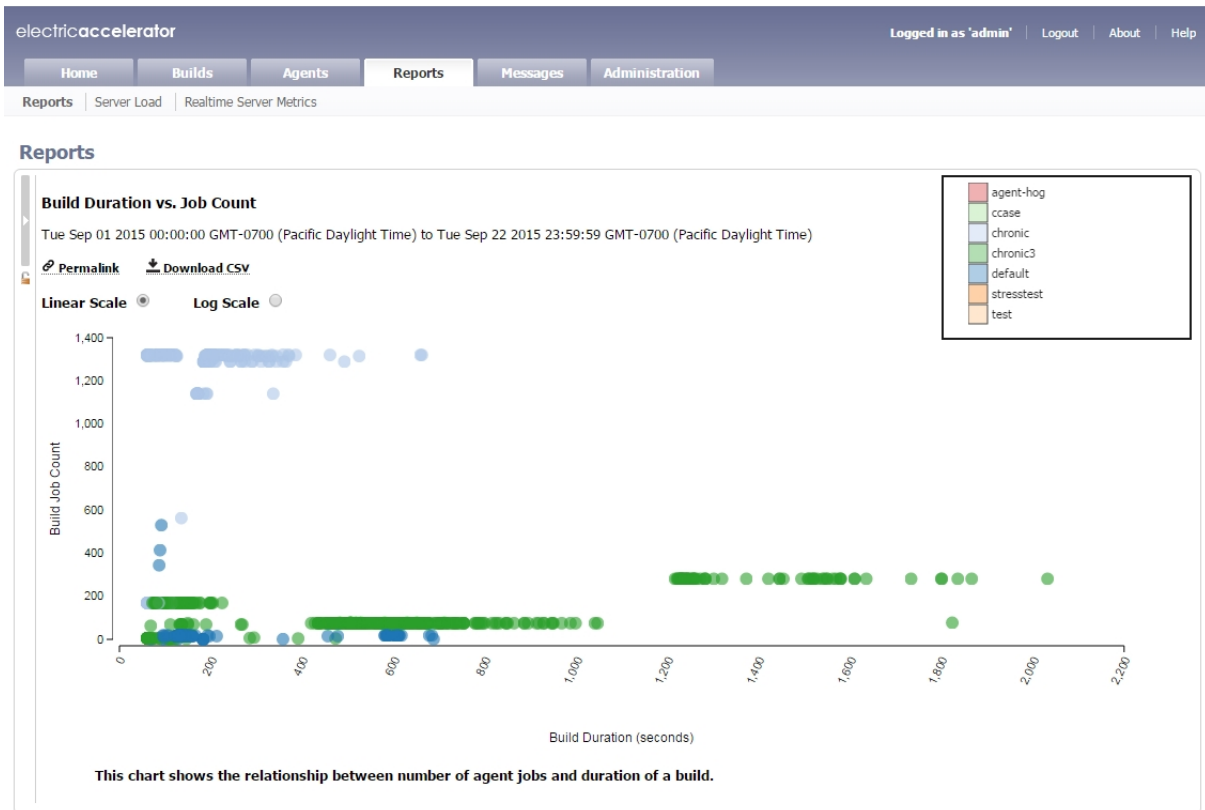


You can input a build class or one or more build names along with a date range. Also, you can specify a threshold for conflicts in a build. Builds with conflicts lower than the threshold are filtered out. You can also select whether to color the report by build class or build name.

As with other scatter plots, you can hover over a build for details about that build and can click a build to navigate to the **Build Details** page for that build (see the screenshots in the “[Build Duration Report](#)” section above for an example). Also, as with other scatter plots, you can click and drag to select multiple builds to view a table that shows details for those builds (see the screenshots in the “[Build Duration Report](#)” section above for examples). You can switch from linear scale (the default) to logarithmic scale.

Build Duration vs. Job Count Report

This report is a scatter plot of build duration versus the number of jobs per build:



You can input a build class or one or more build names along with a date range. You can also select whether to color the report by build class or build name.

As with other scatter plots, you can hover over a build for details about that build and can click a build to navigate to the **Build Details** page for that build (see the screenshots in the “[Build Duration Report](#)” section above for an example). Also, as with other scatter plots, you can click and drag to select multiple builds to view a table that shows details for those builds (see the screenshots in the “[Build Duration Report](#)” section above for examples). You can switch from linear scale (the default) to logarithmic scale.

Build Summary Report

This report is a table that shows aggregate statistics (average wait time and total workload) that are grouped by either build class or build name:

The screenshot shows the ElectricAccelerator web interface. The top navigation bar includes 'Home', 'Builds', 'Agents', 'Reports' (selected), 'Messages', and 'Administration'. Below this, a sub-navigation bar shows 'Reports', 'Server Load', and 'Realtime Server Metrics'. The main content area is titled 'Reports' and contains a 'Build Summary' section. This section displays a date range: 'Tue Sep 01 2015 00:00:00 GMT-0700 (Pacific Daylight Time) to Tue Sep 22 2015 23:59:59 GMT-0700 (Pacific Daylight Time)'. It includes links for 'Permalink' and 'Download CSV'. A table shows aggregate build stats grouped by build class. Below the table, a note states: 'This chart shows aggregate build stats grouped by either build class or build user label.'

Build Class	Average Wait Time	Total Workload
chronic	0	607455.779
chronic3	0	1594256.6459
default	0	151928.3907

You can input a date range. You can also select whether to group by build class (the default) or by build name patterns, where you can specify a list of build names through wild card patterns.

You can click a build class to see builds for that class grouped by user:

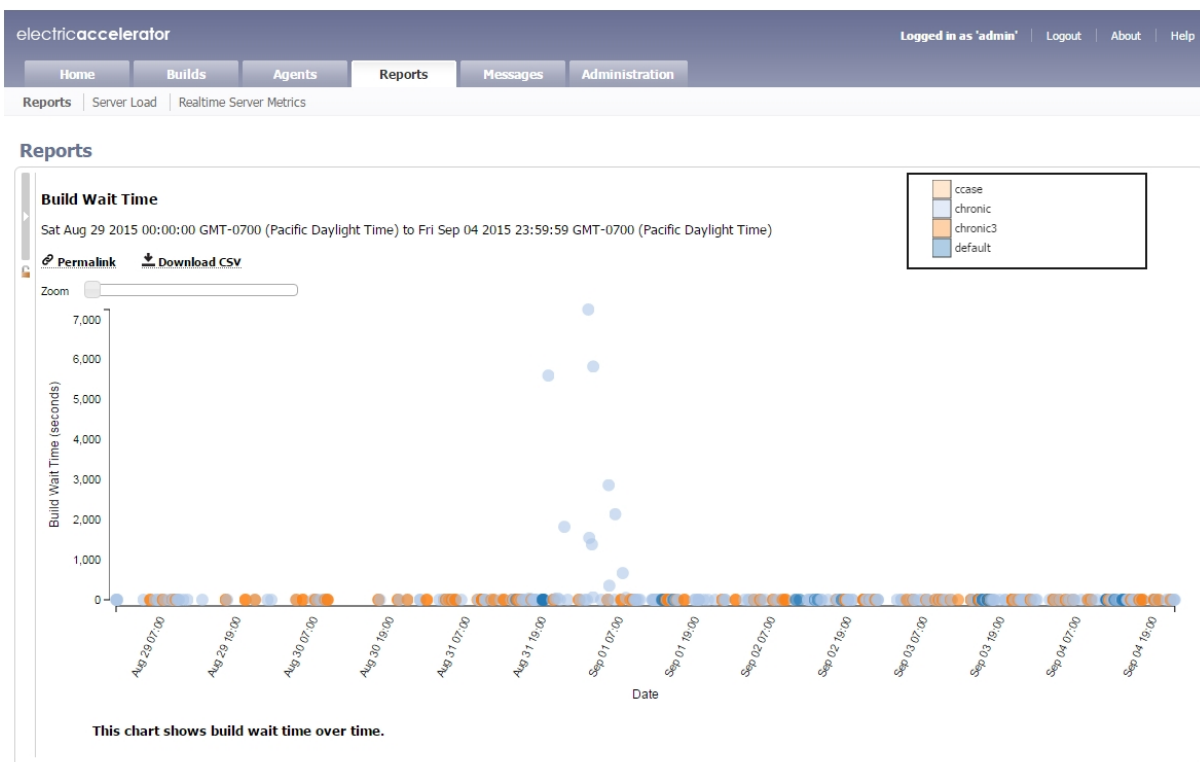
The screenshot shows the ElectricAccelerator web interface with the 'Reports' section selected. The main content area is titled 'Reports' and contains a 'Build Summary for default (Back)' section. This section displays a table showing aggregate build stats grouped by user (Username). The table has three columns: 'Username', 'Average Wait Time', and 'Total Workload'. The data rows are 'a', 'build', and 'ericm'.

Username	Average Wait Time	Total Workload
a	0	1498.0062
build	0	147571.9125
ericm	0	2858.472

This report is useful for chargebacks to business units that are using the cluster.

Build Wait Time Report

This report is a scatter plot that shows the wait times for builds over a specified time period:

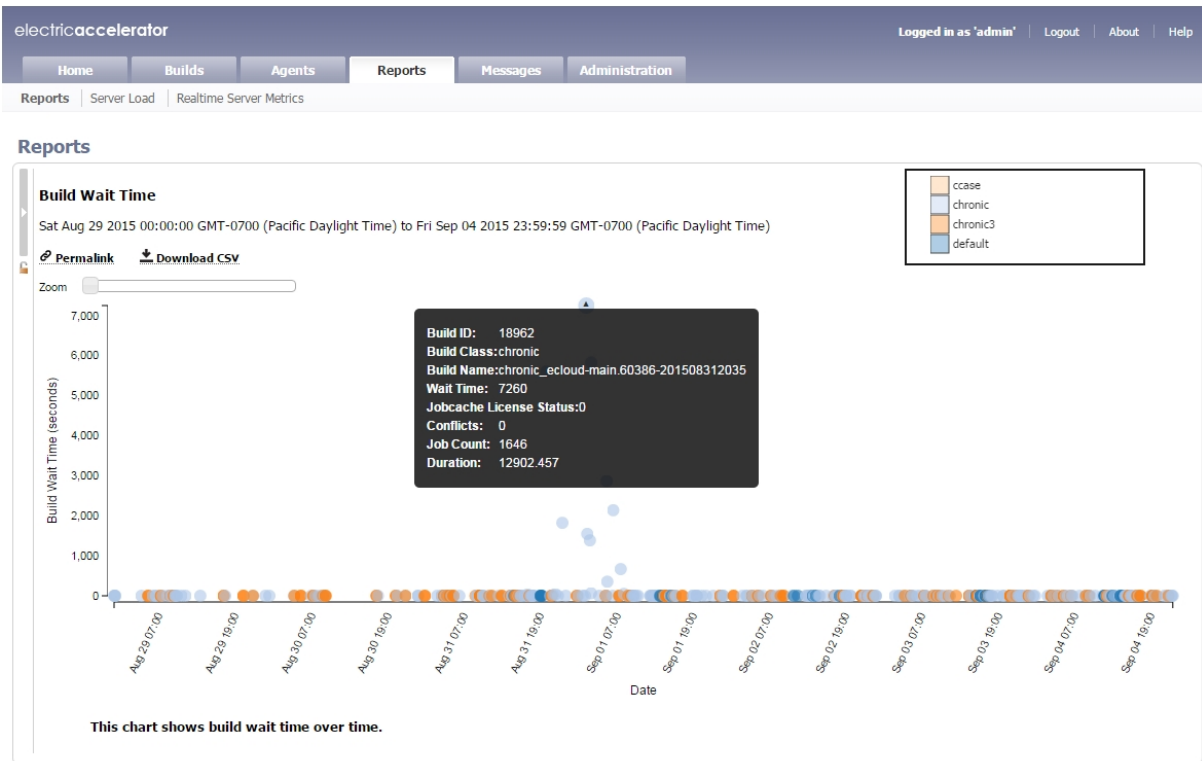


Wait time is the time when eMake has no agents available to run the build. A high number of builds with a non-zero wait time indicates that the cluster might be too small.

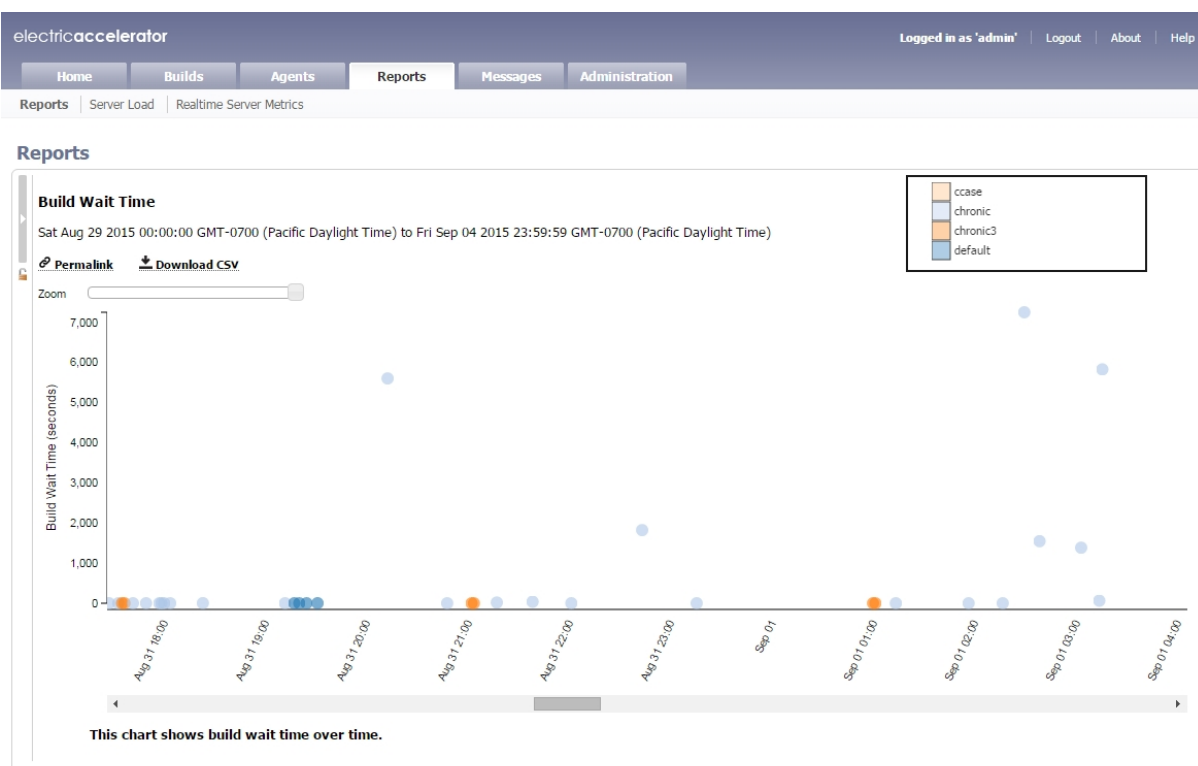
You can specify a build class or one or more build names along with a date range as input.

Also, as with other scatter plots, you can click and drag to select multiple builds to view a table that shows details for those builds (see the screenshots in the ["Build Duration Report"](#) section above for examples).

You can hover over a build for details about that build:



You can use the **Zoom** slider to zoom in on a cluster of points that are close together:



You can click a build to navigate to the **Build Details** page for that build:


electricaccelerator

[Home](#)
[Builds](#)
[Agents](#)
[Reports](#)
[Messages](#)
[Administration](#)

[Builds](#)
[Build Classes](#)

Build Details – chronic_ecloud-main.60386-201508312035

[View Messages](#)
[Download Build Log](#)
[Delete Build](#)



Name: chronic_ecloud-main.60386-201508312035

Status: OK

Start Time: 2015-09-01 09:30:23 UTC

Duration: 3 h 35 m 2 s

Conflicts: 0

Details

Build ID: 18962


Job Count: 1646

Min Agents: 2

Effective Agent Alloc: 19.00 %

IP set: 0

Build Class: chronic

Priority: 

Max Agents: 64

Wait Time: 2 h 1 m

Workload: 5541.01


Host Name: win4-ea-cm

User Name: build

User Label: ecloud-main.60386-201508312035

IP Address: 10.1.216.167

OS User Name: build

Platform:  Windows

History Exists: Yes

Command Line: c:\tools\ECloud\i686_win32\bin\emake.exe --make-localagents=n --make-annofile=anno-windows.log --make-annodetail=basic.md5,waiting --make-debug=g --make-logfile=debug-windows.dlog --make-cm=winbuild-cm --make-class=chronic --make-build-label=ecloud-main.60386-201508312035 --make-history=create --make-historyfile=ecloud-windows.data -k -C ecloud_all_and_buildtest

Resource Request: accelerator

CWD: N:/ecloud-main.60386-201508312035

EMake Version: 2015.06.05.59102 32-bit (build_main_59102_OPT_2015.06.05_19:20:59)

Build Log Directory: C:\ECloud\i686_win32\build_logs\1\8\9\6\18962

History File: ecloud-windows.data

Comments

Comment

Modified By

Last Modified

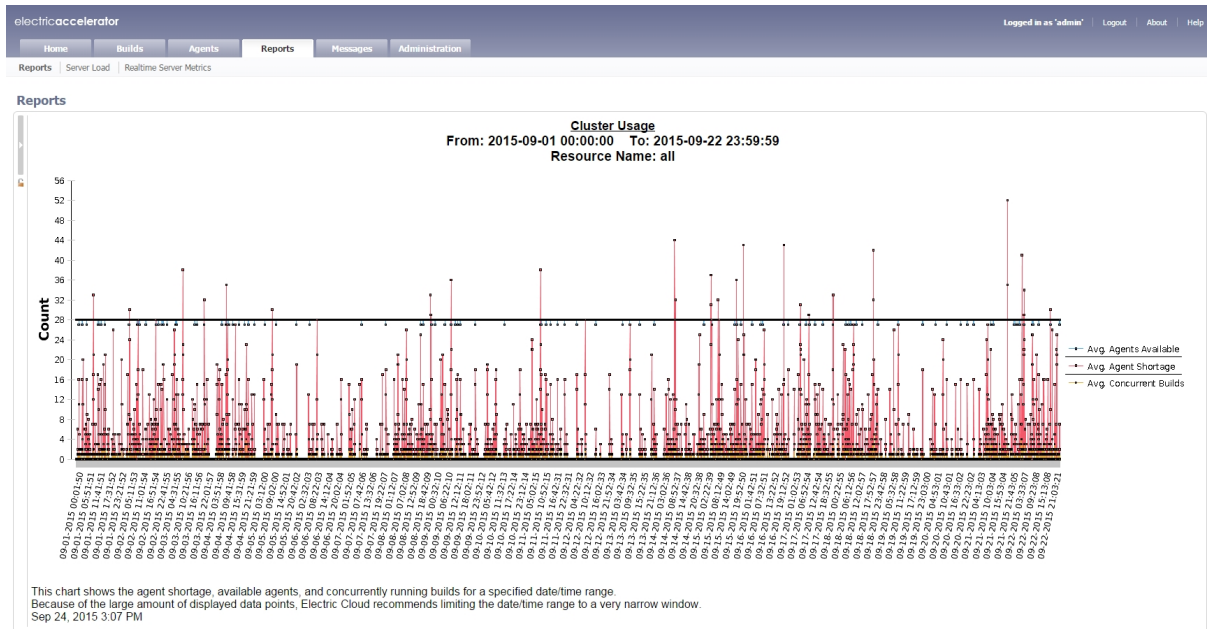
Actions

Currently, there are no records to display in this list.

New Comment

Cluster Usage Report

This report is a chart that shows the agent shortage, available agents, and builds running concurrently for a specified time period:



The report lists all available data points; these might be too numerous to show in the chart at one time, so you should specify a short time period. For a specific date range, data appears in a line chart to show the change of certain key properties in the server:

- **Avg. Agent Shortage**—If this value exceeds 0, at least one build could have used more agents than the cluster had. This generally means that the cluster needs more agents.
- **Avg. Agents Available**—This is a view of how many agents were available at a time.
- **Avg. Concurrent Builds**—This view shows the number of builds running at a particular time.

Note: Results in the Cluster Usage report in the Cluster Manager might not appear in Internet Explorer 10. To work around this issue, choose **Tools > Compatibility View** from the browser menu and then re-run the report.

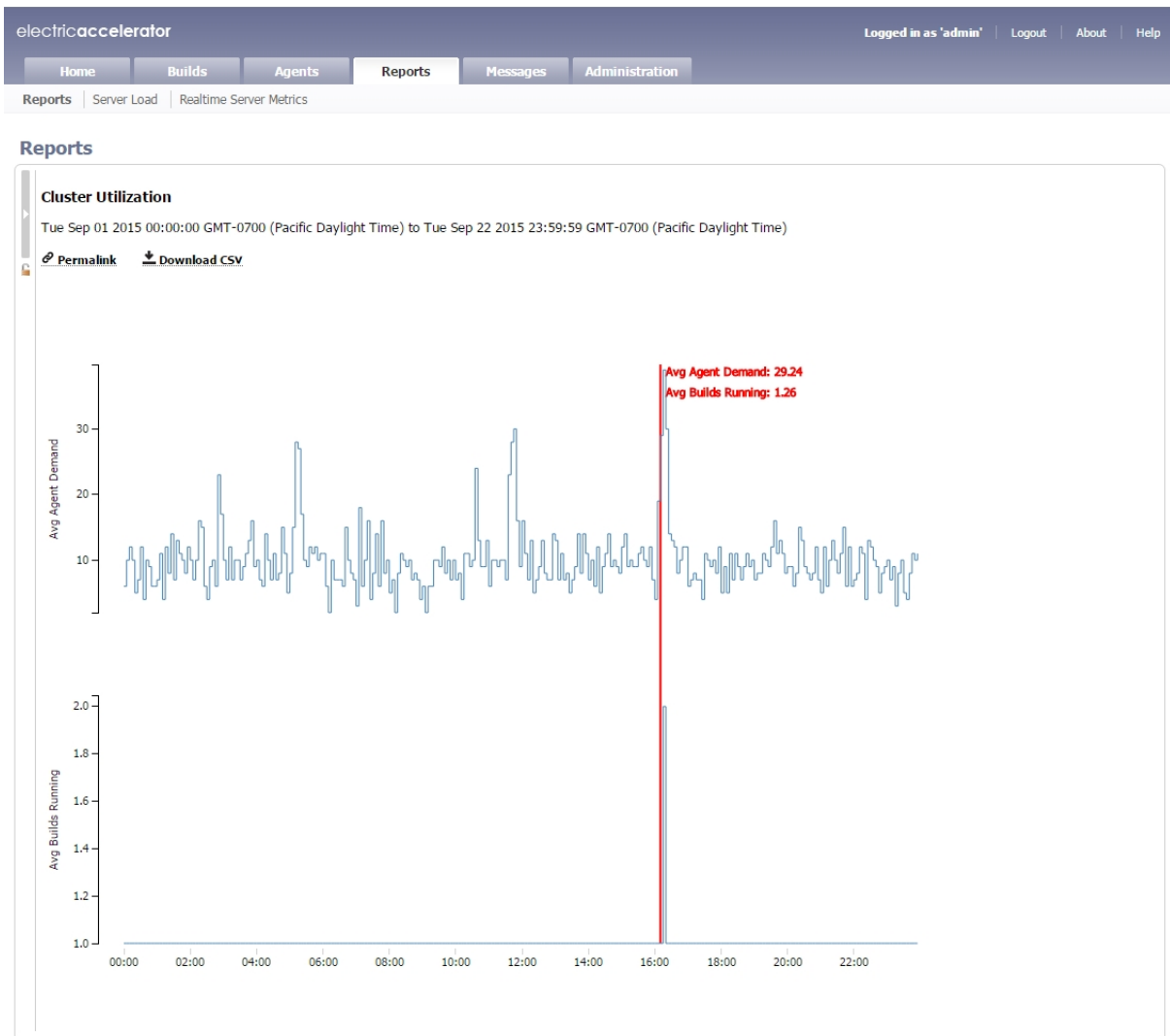
Cluster Utilization Report

This report is two stacked, stepped line charts that represent the average number of builds running and the average combined agent demand from all running builds over a specified time period:



You can input a date range along with a resource name.

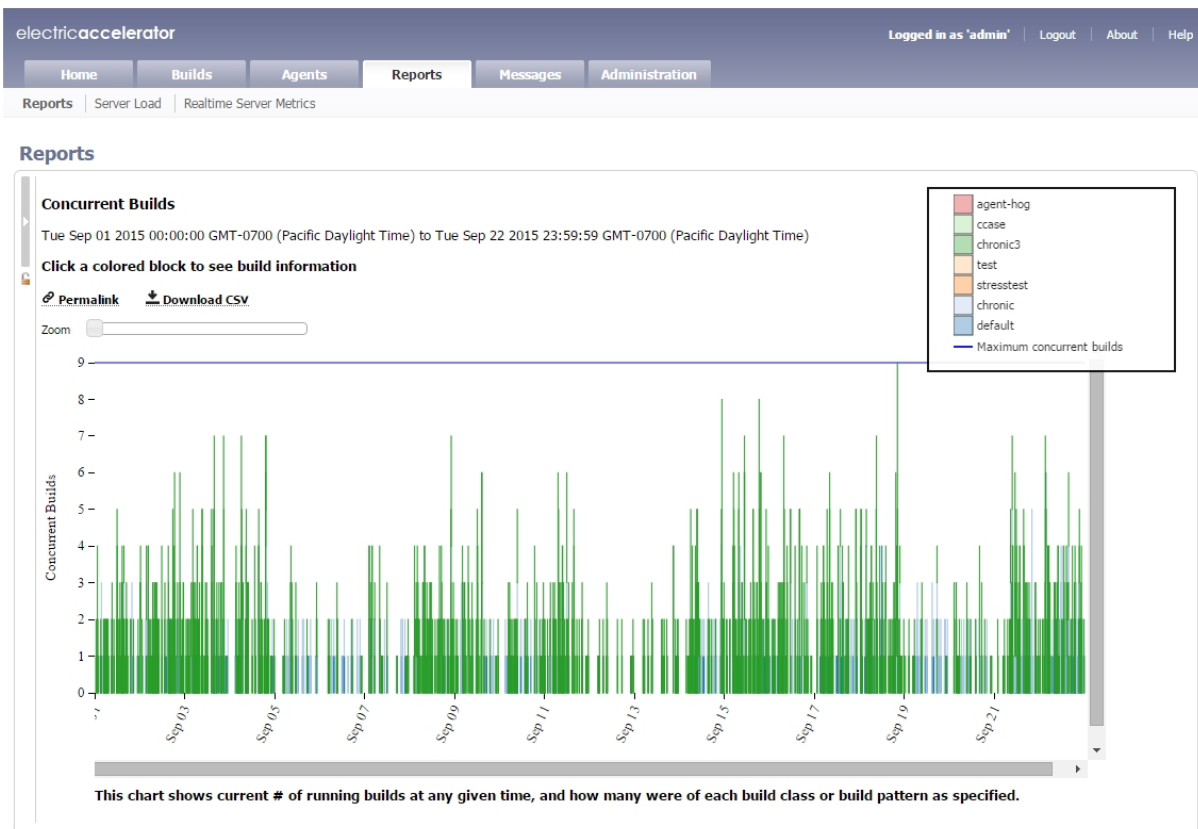
You can hover over the chart to see the average agent demand and average builds running at any point of time:



To view this report, you must have “read” permission for resources and reports. For details, see the “[Checking Your Permissions to Run Reports](#)” section above. If you do not have “read” permission for both, reports will still be available, but the following popup appears: Error: AccessDenied: User <username> does not have the required ResourcesRead privileges to perform this operation.

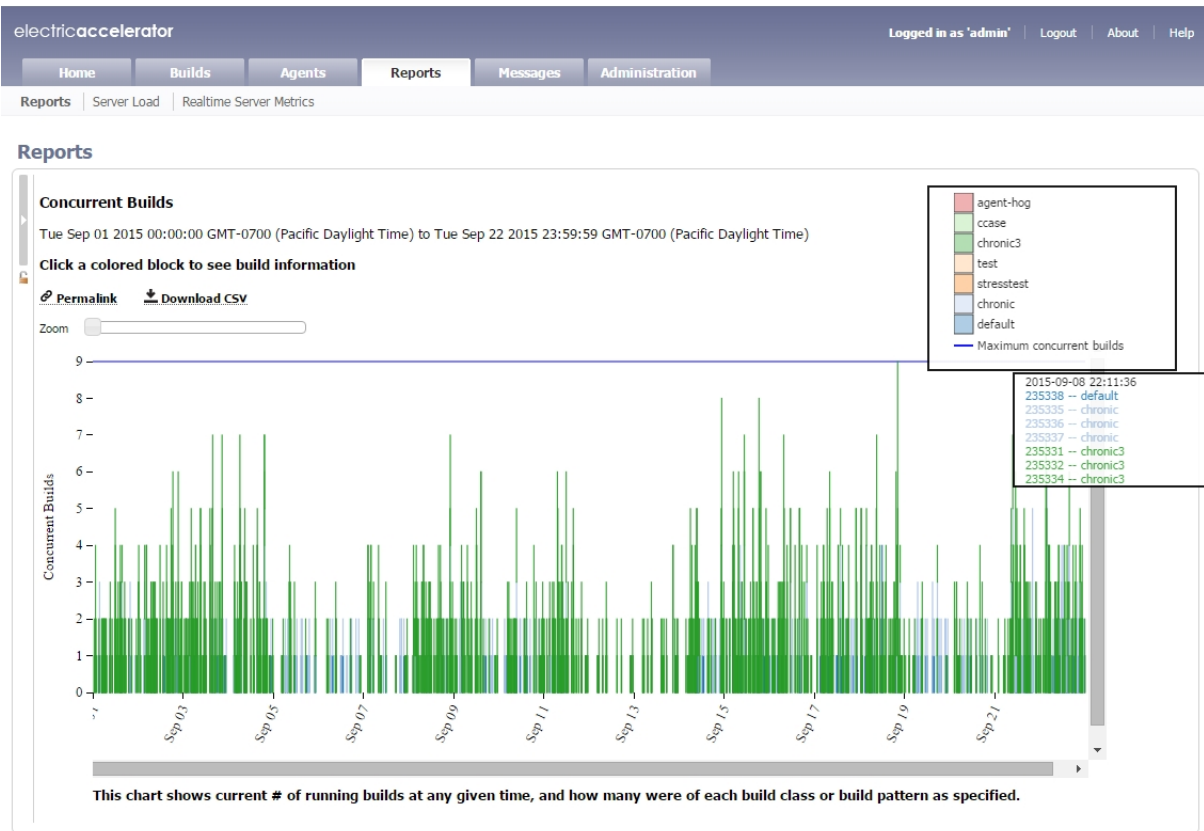
Concurrent Builds Report

This report is a skyline chart, stacked by build class, that shows the current number of running builds at any time and how many were of each specified build class or build name pattern:

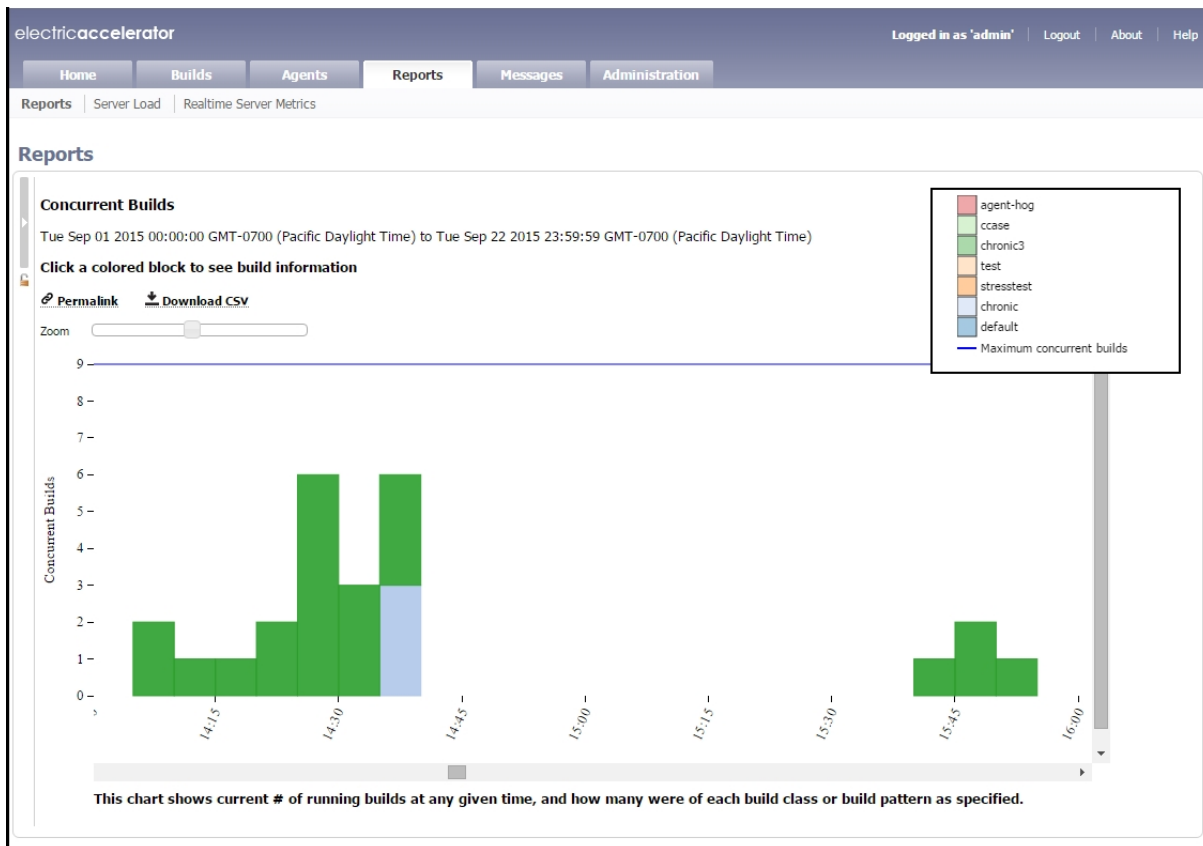


You can input a date range, a sample interval size (the default is five minutes), and whether to group by build class (the default) or by build name. You can omit builds under a specified number of seconds. You can specify a series of build name patterns.

You can hover over a stack to see a list of concurrent builds for that stack:



You can use the **Zoom** slider to zoom in to a particular stack or set of stacks:



You can click to see a table of builds for each build event (when a build starts or when a build ends):

electricaccelerator Logged in as 'admin' [Logout](#) [About](#) [Help](#)

[Home](#) [Builds](#) [Agents](#) [Reports](#) [Messages](#) [Administration](#)

[Reports](#) [Server Load](#) [Realtime Server Metrics](#)

Reports

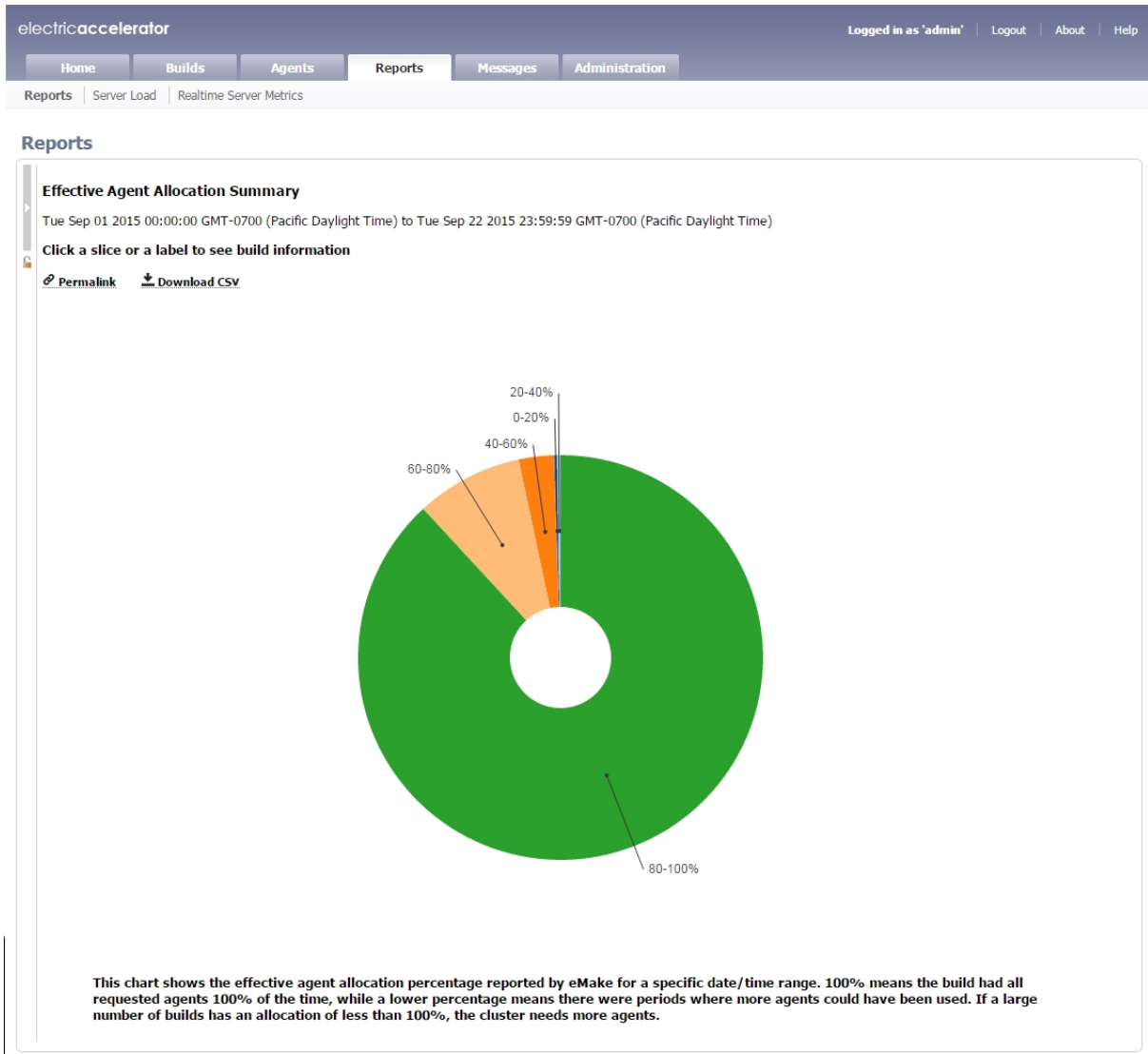
Builds running at 2015-09-09 14:28:11 ([Back to chart](#))

[Download CSV](#)

Build ID	Build Class	Duration
235412	chronic3	114.949
235413	chronic3	87.867
235414	chronic3	617.953
235415	chronic3	626.578
235416	chronic3	120.775
235417	chronic3	519.888

Effective Agent Allocation Summary Report

This report is a pie chart that shows the effective agent allocation percentage reported by eMake for a specified time period:



100% means the build had all requested agents 100% of the time, while a lower percentage means there were periods where more agents could have been used. The majority of your builds should be in the 90–100% range. If a large number of builds has an allocation of less than 100%, the cluster needs more agents.

You can input either a build class or a build name along with a date range for this report. By default, the Ranges field contains: 0–20%, [0,20), 20–40%, [20,40), 40–60%, [40,60), 60–80%, [60,80), 80–100%, [80,100]. As a result, the default report displays the following allocation ranges respectively: 0–20%, 20–40%, 40–60%, 60–80%, and 80–100%.

You can click a slice or a label to see build information:

electricaccelerator

Logged in as 'admin' | Logout | About | Help

HomeBuildsAgentsReportsMessagesAdministration

ReportsServer LoadRealtime Server Metrics

Reports

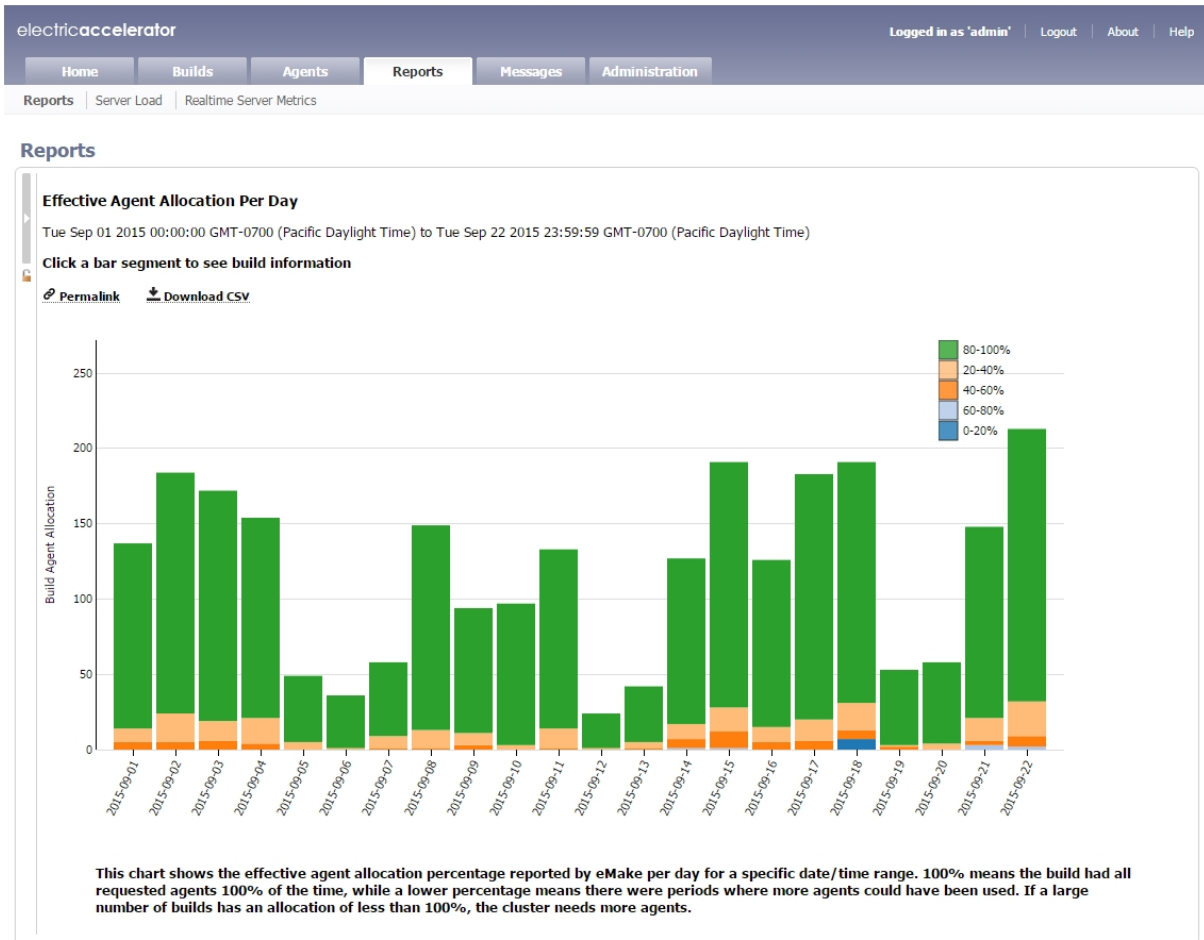
Builds with 60-80% agent allocation (Back to chart)

Download CSV

Build ID	Build Class	Duration
234412	chronic3	83693
234413	chronic3	100957
234422	chronic3	83671
234425	chronic	267126
234475	chronic3	95726
234512	chronic	81876
234519	chronic3	108531
234535	chronic	313309
234543	chronic3	83283
234546	chronic3	86956
234550	chronic	71848
234551	chronic	72730
234558	chronic	241882
234560	chronic3	78870
234563	chronic3	82112
234568	chronic	79862
234577	chronic3	80949
234582	chronic3	85856
234593	chronic3	102923
234608	chronic3	140045
234611	chronic3	82678
234617	chronic3	78875
234635	chronic3	78745
234636	chronic3	106174
234675	chronic3	114393
234684	chronic3	90220
234686	chronic3	111631
234726	chronic3	84254

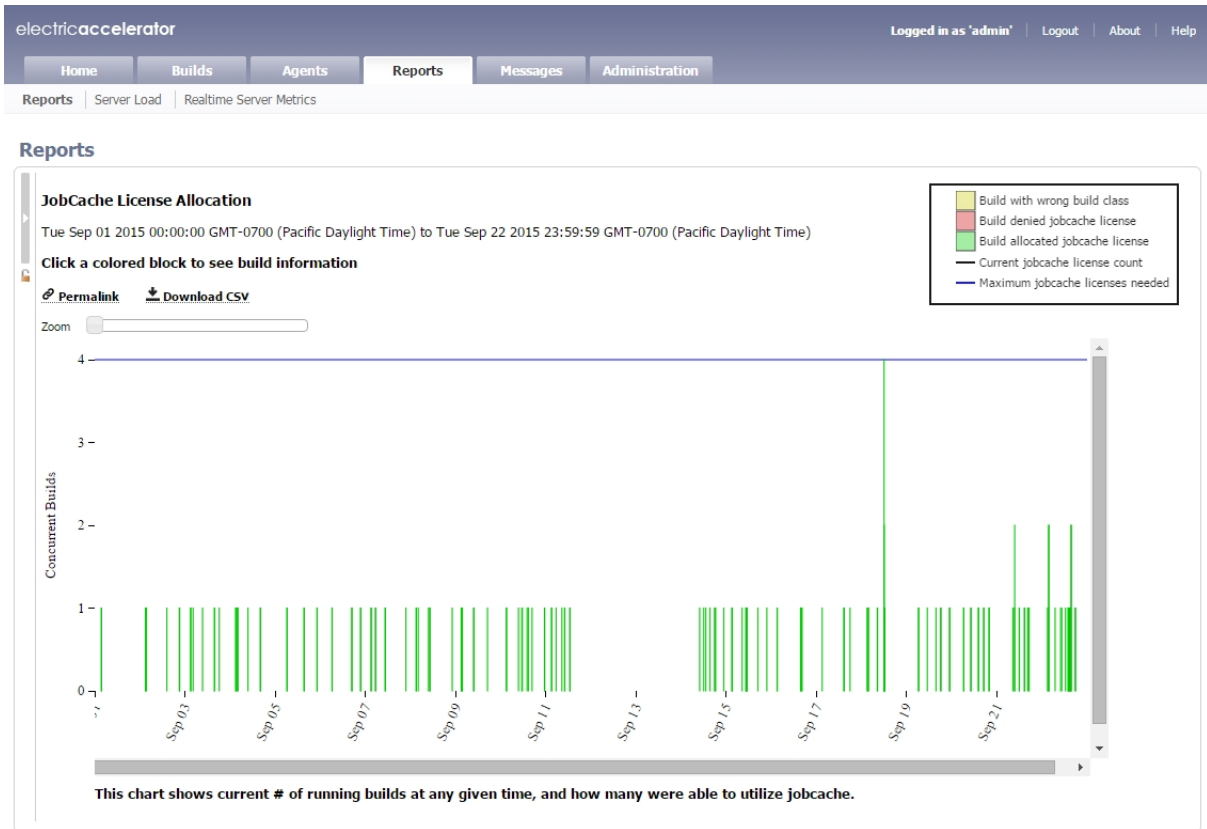
Effective Agent Allocation Per Day Report

This report is the same as the Effective Agent Allocation Summary report, except it is a stacked bar chart that shows agent allocation per day:



JobCache License Allocation Report

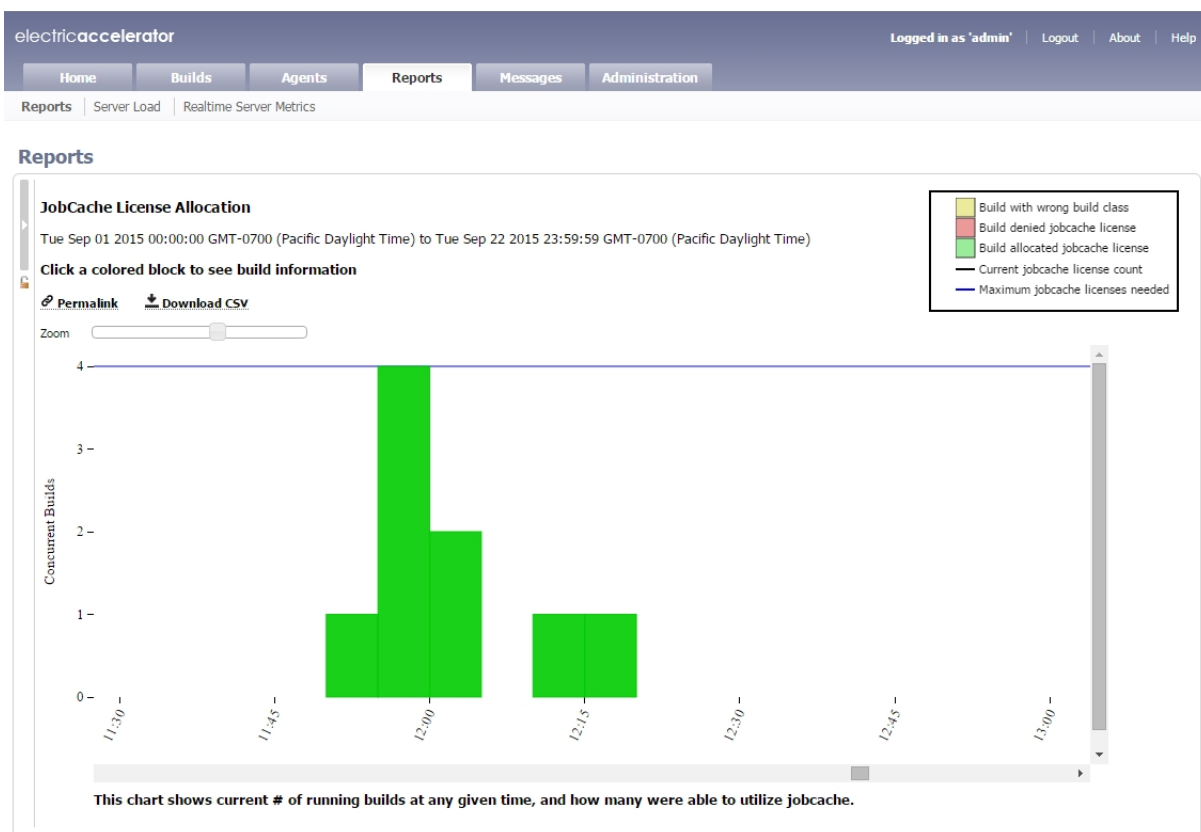
This report is a skyline chart that shows concurrent builds at any time, stacked by JobCache license request results, to show how many were able to use JobCache:



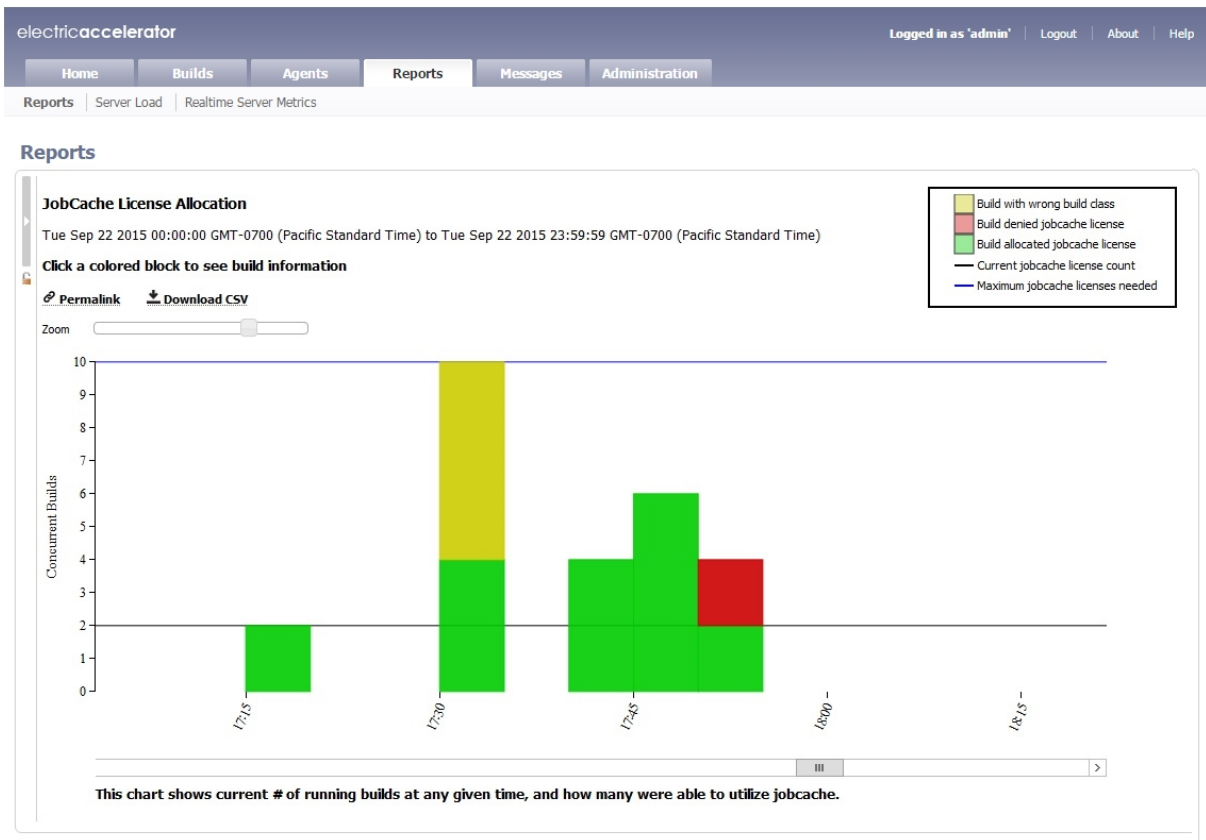
The chart includes two color-coded lines indicating the current JobCache license count and the maximum JobCache licenses needed.

You can input a build class or a build name. You can also specify a build duration threshold in which builds under that duration are not included in the report.

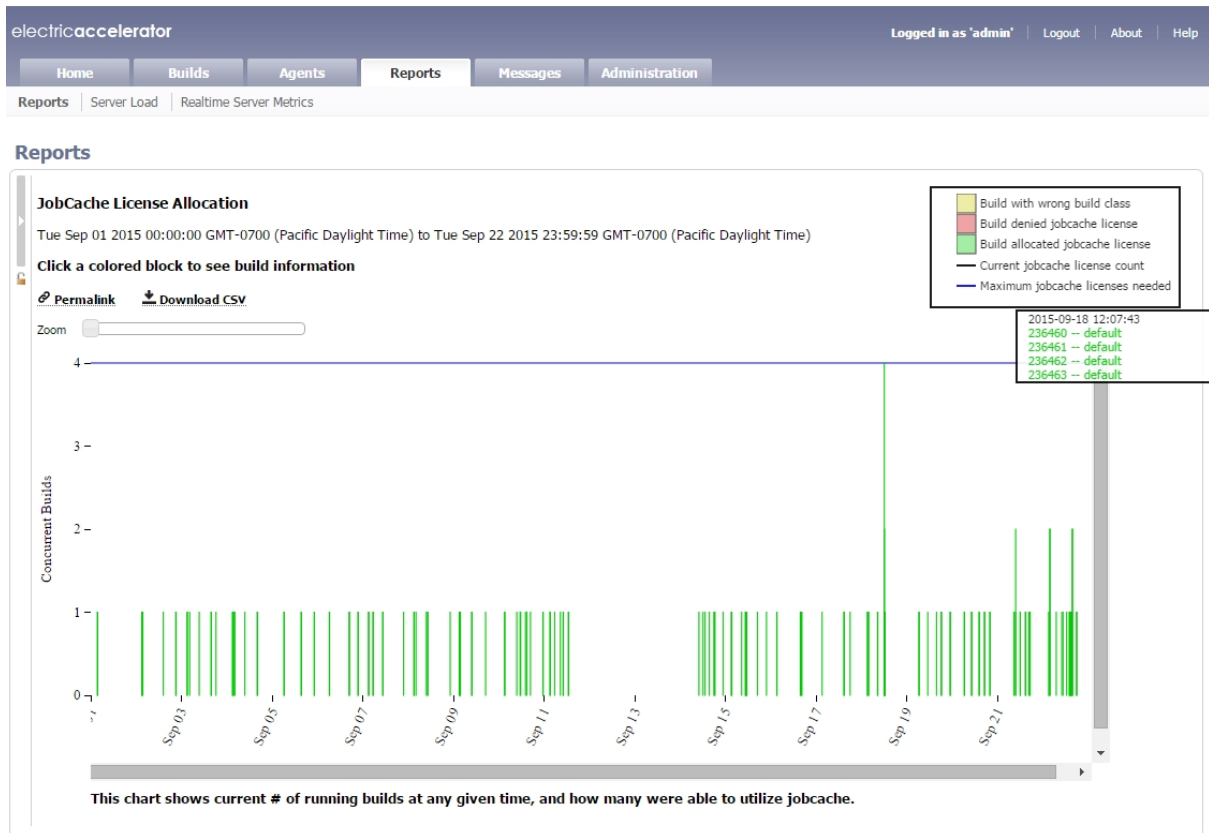
You can use the **Zoom** slider to zoom in to a particular stack or set of stacks:



Each build is color coded to indicate whether the build was allocated or denied a JobCache license or was in a build class not enabled to use JobCache:



You can hover over a build event to see a list of builds in that stack:



You can click a stack to see a table of builds for that build event:

electricaccelerator

Logged in as 'admin' | Logout | About | Help

Home Builds Agents Reports Messages Administration

Reports Server Load Realtime Server Metrics

Reports

Buils running at 2015-09-18 12:07:43 (Back to chart)

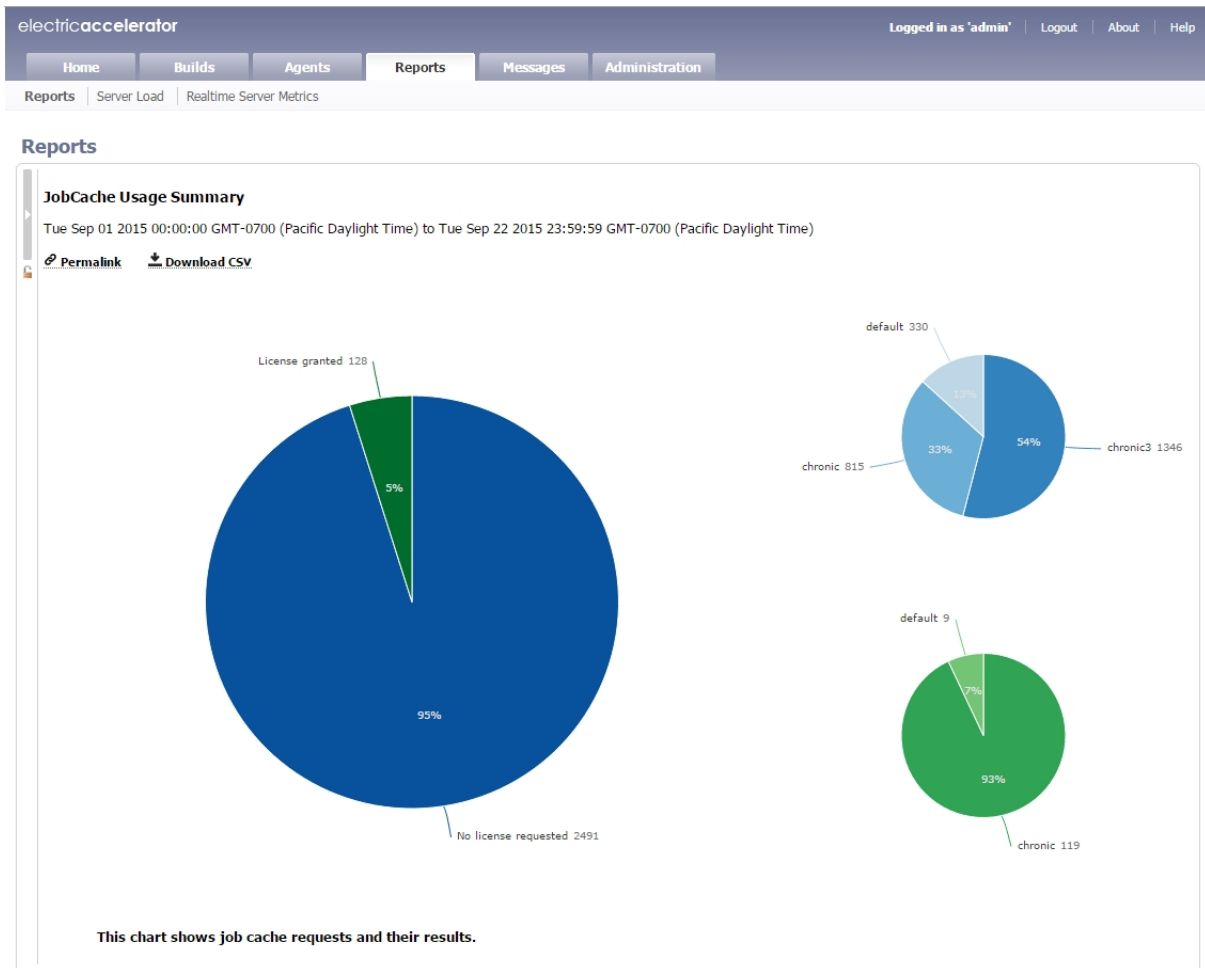
Download CSV

Build ID	Build Class	Duration
236460	default	183.444
236461	default	182.459
236462	default	182.474
236463	default	183.508

JobCache is an add-on feature. It requires a separately-purchased license and is available only on Linux and Solaris platforms.

JobCache Usage Summary Report

This report is several pie charts that show job cache requests and their results:



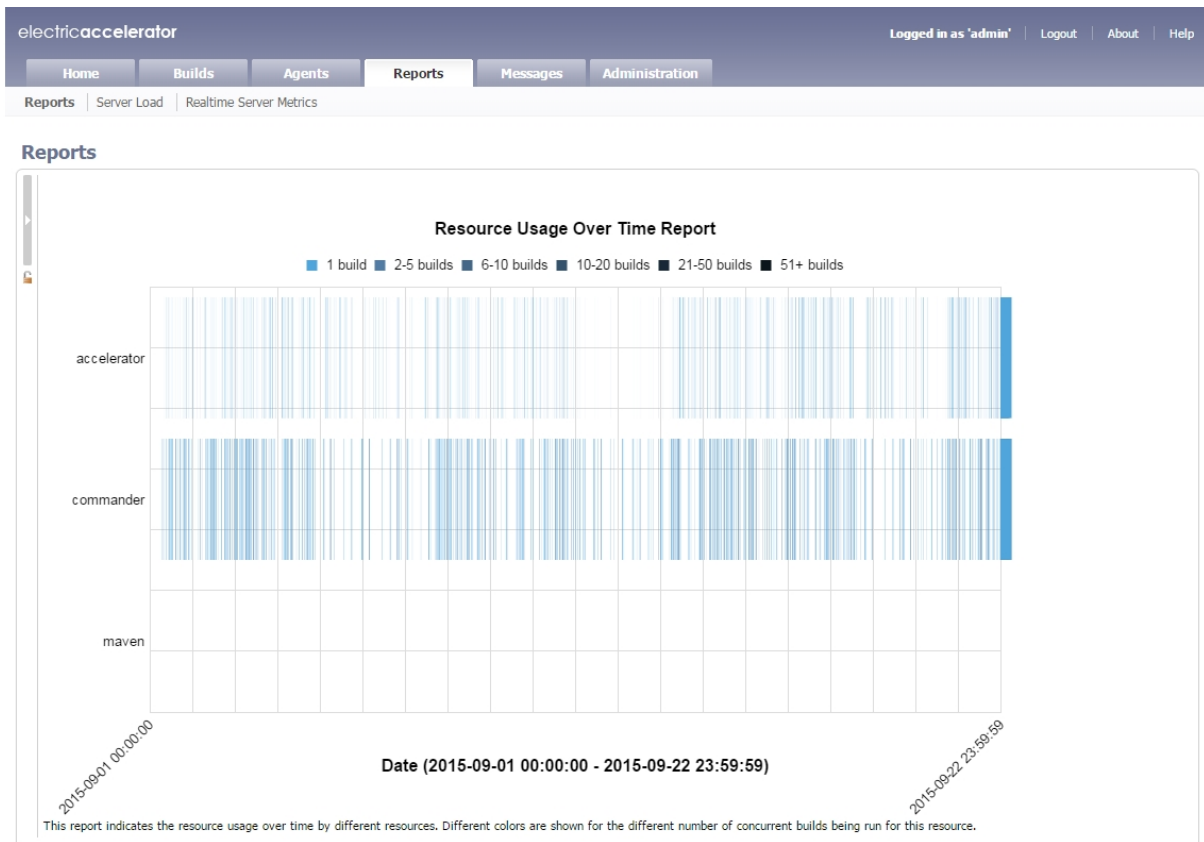
The large chart will have one to three segments (determined by actual JobCache usage). The large chart displays the percentages of builds that

- Did not request a license.
- Requested and were granted a license.
- Were denied a license because of license limits or build class configuration.

Each small chart corresponds to a segment of the large chart and represents a breakdown based on build class.

Resource Usage over Time Report

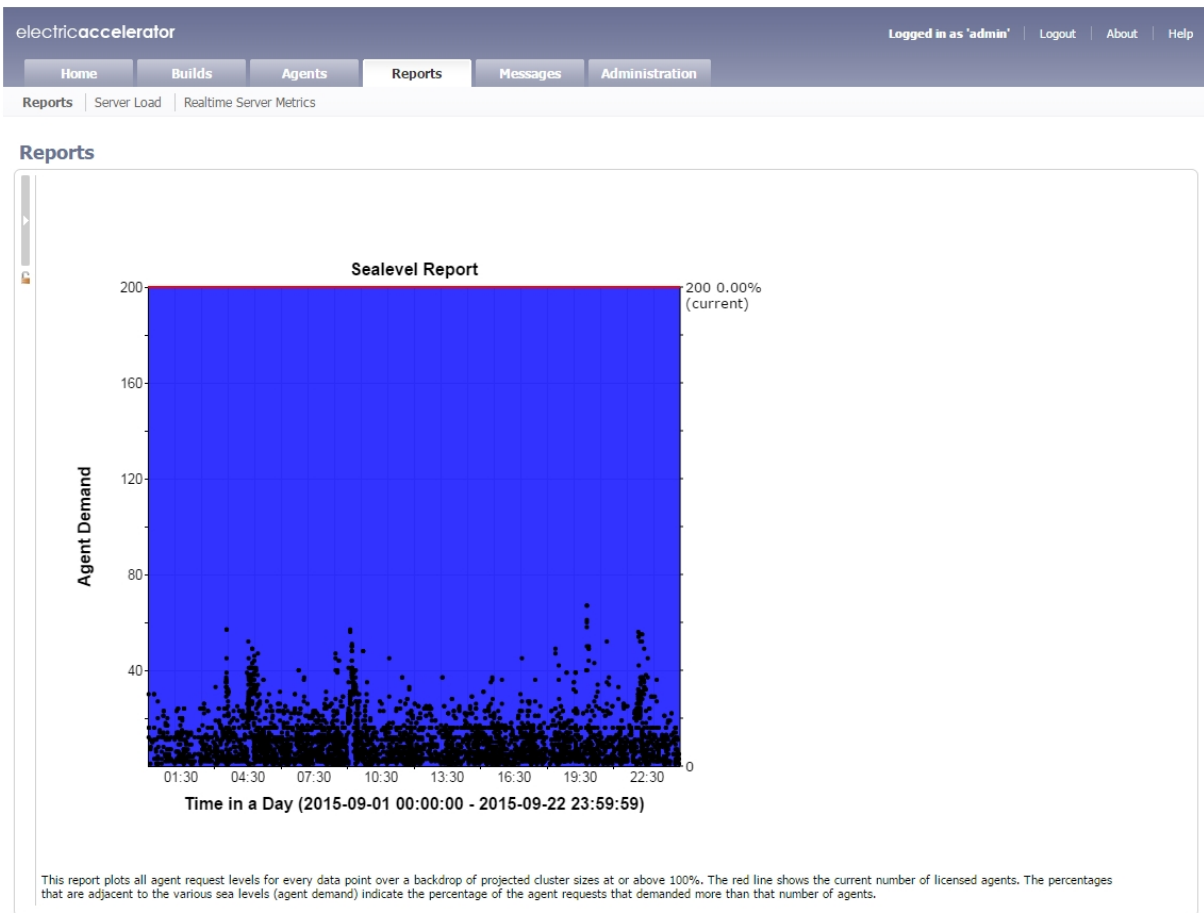
This report is a chart that shows the number of simultaneous builds that ran on each resource over a specified time period:



Each color represents the number of concurrent builds that are running for that resource.

Sea Level Report

This report is a plot of agent requests against a backdrop of projected cluster sizes at or above 100%:



The red line shows the current number of licensed agents. The percentages that are adjacent to the various sea levels (agent demand) indicate the percentage of the agent requests that requested more than that number of agents.

To view this report, you must have “read” permission for resources and reports. For details, see the “[Checking Your Permissions to Run Reports](#)” section above. If you do not have “read” permission for both, reports are still available but the following popup appears: Error: AccessDenied: User <username> does not have the required ResourcesRead privileges to perform this operation.

Server Load

About Server Load

- Server load information is grouped by the resource specification.
- The resource name "Cluster" is used to designate the server load for all resources.
- Server load statistics are written to disk at regular intervals.

The Current Server Load section displays agent information at-a-glance.

Functionality

- Click a column heading to sort the information in that column.
- Set the page refresh interval to keep your information current.
- Set the number of records you want to see per page.

Column Descriptions

Column	Description
Resource Name	This attribute comes from the resource specification for each build job. The resource is specified either on the eMake command line (<code>--emake-resource=</code>) or as defined in the build class.
Create Time	Server load statistics are written to disk at regular intervals. This interval is defined on the Administration > Server Settings page. The default value is 5 minutes. The Create Time attribute identifies when the interval began. The statistics show the average value over the interval starting at Create Time. For example, if the interval is set for 5 minutes and Create Time shows 10:00am, Server Load statistics are the average values from 10:00am to 10:05am.
Agents Available	This value shows the average number of enabled and active agents in the cluster over the specified time period. This value is available <i>only</i> for cluster statistics—it is not available for individual resource statistics.
Agent Demand	This value shows the average number of agents all builds could have used if those agents were available. For example, if two builds are using two different resources and each build could use 15 agents, the cluster load shows an Agent Demand of 30 agents and each resource shows 15.
Agents in Use	This value shows the average number of agents that are used for all builds.
Agent Cluster Shortage	This value shows how many additional agents could have been used by the builds over the given time period. This value is filled in <i>only</i> for cluster statistics—it is not available for individual resource statistics.

Column	Description
Agent License Shortage	This value shows how many agents could have been used to accelerate builds if licensed agents were available. If this number exceeds 10 percent of the Agent Demand, consider purchasing additional Agent Licenses to ensure the best possible performance.
Avg Build Duration	This value shows how long the current builds have been running. For example, if [during the time period] the average is 5 builds running concurrently and the builds run in about a minute, this value will be approximately 60 seconds. However, if 5 builds have been running for half an hour, the value will be approximately 30 minutes. Use this value to determine if there are several quick-running builds or a few long-running builds over any particular time period.
Avg # Builds Running	This value shows the average number of builds that ran simultaneously over the specific time period.

Realtime Server Metrics

This page can display a number of server metrics:




- AgentRequest Elapsed Time
- DispatchARQueue Size
- DispatchQueue Size
- JVM Free Memory
- Load Average
- Number of Concurrent Builds
- Number of Received Requests

To view this page, you must have “read” permission for Administration and Reports. If a user does not have “read” permission for both, Realtime Server Metrics is not displayed under the Reports tab.

Messages

This page displays messages from the Cluster Manager, agents, and eMake. Messages are listed by the global message ID.

Column Descriptions

Column	Description
Severity	Levels are:  Info  Warning  Error
ID	This is the message identifier.
Text	This is the text of the message.
Time	This is the time of the message.
Build	This is the build that reported the message. The field is empty if no message was reported by the build. Click the Build name in this column to go to the Build Details page for more information.
Agent	This is the agent that reported the message. The field is empty if no message was reported by the agent. Click the Agent name in this column to go to the Agent Details page for more information.
Actions	Click the Delete Message link in this column to delete the corresponding message.

Using the Custom SQL Filter

To use the Custom SQL filter, type the SQL you want to run in the Where field and click **Run Filter**.

The following fields are available for custom SQL filters:

- id
- text
- severity
- agent_name
- create_time
- build_id

Examples

To find messages with an ID greater than 16, use the custom SQL: `id > 16`

To find connection failure messages, use: `text like 'Unable to connect%'`

For examples of filters that use the following fields see:

- severity: Warnings
- agent_name: Messages By Agent
- create_time: Messages By Date
- build_id: Messages By Build ID

Message Policies

This page assumes User permissions and email notification information is already configured. This subtab is viewable only if you have Modify permission defined in the User field on the Permissions page. The administrator can edit a user's permission on the Administration > Permissions page.

1. Select the **Watch Messages** checkbox if you want to be notified when new messages are entered into the message log.
2. Select the appropriate button for the appropriate notification level.
Severity levels for email notifications are: Error, Warning, or Info.
3. Click **OK** to save the settings.

Administration

The Administration tab opens to the User Settings page and displays the following subtabs:

- [Filters](#)
- [Permissions](#)
- [Users](#)
- [Groups](#)
- [User Settings](#)
- [Server Settings](#)
- [Database Configuration](#)
- [Licenses](#)

Filters

Most lists in the ElectricAccelerator web interface support filtering. The exceptions are:

- Users and Groups, which follow a different model because they are a combination of local and LDAP/Active directory based users.
- Licenses do not support filters because there are a small number of them in the system.
- Current LSF jobs because the set of jobs is expected to be small.

A filter allows drill-down to a relevant set of information. The Cluster Manager supplies a set of predefined filters for commonly requested information, for example, broken builds, list of builds for a particular build class, list of broken agents, error messages, and many more. You can choose a predefined filter or a global filter from the Filter drop-down menu. Global filters require the user to have permission to use them. Users can create their own custom filters also.

The Filter panel is usually displayed on the left-side of the web page. You can collapse or expand this panel by clicking the adjacent vertical "gray bar" icon. The system remembers the Filter panel expansion status on a per session basis. After a filter runs, the filter section automatically collapses.

The Filters page lists all ElectricAccelerator-supplied filters, Global filters, and any filters you create. You can create filters for your use only or create additional Global filters, which can be shared by those users who have appropriate permissions.

Filter Panel

The Filter panel always has the following functions:

- **Run Filter**—Runs the selected filter.
- **No Filter**—Refreshes the list to display all items.
- **Run Filter => XML**—Downloads filtered records to a local XML file. The browser provides a dialog to specify a location on the local drive.
- **Page => XML**—Exports the current page to a local XML file. The browser provides a dialog to specify a location on the local drive.



- To close (or open) the Filter panel, click this icon
- To prevent the Filter panel from collapsing (or expanding) when running filters, click this icon

The Filter panel on pages with item lists also has the following functions:

- **New Filter**—Allows you to specify a new filter.
- **Edit Filter**—Allows you to open the currently selected filter on the Edit Filter page, so you can make the changes you need.
- **Manage Filters**—Takes you to the Administration page for filters, where all available filters are listed.

Functionality

- Clicking a filter Name in the first column takes to you that filter's [Edit Filter](#) page, where you can edit the filter.

- The Actions column allows you to [edit](#) or [delete](#) the filter in that row.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.
 - Set the number of records you want to see per page.

Column Descriptions

Column	Description
Name	This is the name of the filter.
Table	This is the name of the table where this filter is set to run.
Global	Yes or No specifies whether or not this filter is global.
Order By	This column displays the SQL order by clause for the associated table.
Query	This is the SQL query for the associated table.
Actions	Edit Filter —Takes you to the Edit Filter page for the filter in that row. Delete Filter —Deletes the filter in that row.

Filters - Create or Edit a Filter

See [Creating Filters](#) for information about creating a filter, and [Editing Filters](#) for information about editing filter.

Field Descriptions

Field	Description
Name	This is a short string that uniquely identifies the filter.
Table	Use the drop-down menu to select a short string to identify the table being filtered. For example, builds, agents, and messages can be chosen from the table list.
Global	Select this checkbox if this is intended to be a global filter. A filter can be globally accessible or not, depending on whether or not you have administrator privileges.
Order By	This is a SQL order by clause statement for the associated table.
Query	This is a SQL query for the associated table. The query value is a SQL WHERE clause. Comparisons can be between static values (for example "enabled = 1" to get enabled agents), or a comparison could be a user-defined parameter.

The format for these parameters is `{{<label>|<type>}}`

where

- `<label>` is the unique label to use for the parameter
- `<type>` is one of: `datetime`, `datetimeam`, `datetimepm`, `integer`, `integerrequired`, `float`, `floatrequired`, `buildclass`, `table`, `shorttext`, `shorttextrequired`, `noescape`, `noescaperequired`, `text`, or `textrequired`. Depending on the type, different kinds of UIs are displayed for entering the value.

For strings, the definition must be enclosed in "quotes" to maintain correct SQL syntax. You may want to refer to existing filters to understand the syntax. You can also use `{{currentUser}}` to insert the name of the currently logged in user.

Permissions

This page contains a table for Users, Groups, and Administrator. The column headings, links above each table, and links in the Action column are the same for both tables. The same permissions can be granted [or not] to an individual user or to a group. The table information is "view only." You can set or modify permissions on other web pages, accessed by clicking the appropriate links on this page.

Functionality

- You can enable permissions for multiple users or groups at once.
- The Actions column allows you to edit or clear permissions for the user or group in that row.
- You can choose to limit access to build information to the build owner and the administrator only. After this takes effect, only a user's own builds are visible on the Builds page.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.
 - Set the number of records you want to see per page.

Column Descriptions

The following column heading descriptions apply to either table.

Column	Description
Name	This is the name of the user or group.

Column	Description
Access	Yes or No for permission to access the server.
Impersonate	<p>Yes or No for permission to impersonate a different user/group to run eMake. Impersonate changes the user recorded by the Cluster Manager (CM) and that user's permissions. This option does not affect OS user permissions.</p> <p>The simplest way to impersonate a user that can invoke eMake successfully is to meet the following conditions:</p> <ul style="list-style-type: none"> • The Cluster Manager must match your login user name to a user defined in Administration > Users • The user defined on the Users page must have Impersonate permission • The impersonated user must have Invoke permission <p>The following example progresses through the steps to meet each condition.</p> <p>Example: The CM gets your user name from eMake, "mjones". Then the CM looks for a user defined in Administration > Users for a matching user. If unsuccessful, the CM uses the "default" user, whose Full Name is Anonymous.</p> <p>Next, the CM looks in Administration > Permissions and matches the name to permissions. Only Impersonate and Invoke permissions impact the eMake client.</p> <p>The "default" user has Invoke permission by default, but does not have Impersonate permission. Running eMake results in this error message:</p> <pre>% emake --emake-cm bxb-step-002 --emake-impersonate-user=test ERROR EC3140: Couldn't start cluster build: NoImpersonateAccess: User 'Anonymous' is not allowed to impersonate other users</pre> <p>If you create two users, "mjones" and "test", each with default permissions, the following error results when you run a build:</p> <pre>% emake --emake-cm bxb-step-002 --emake-impersonate-user=test ERROR EC3140: Couldn't start cluster build: NoImpersonateAccess: User 'mjones' is not allowed to impersonate other users</pre> <p>If you add Impersonate permission to the user "mjones" and run a build, you get the following error:</p> <pre>% emake --emake-cm bxb-step-002 --emake-impersonate-user=test ERROR EC3139: Couldn't start cluster build: NoBuildAccess: User 'test' does not have permission to run builds</pre> <p>Adding Invoke permission to "test" allows the build to succeed. In the Builds tab, the User@Machine column contains test@Machine instead of mjones@Machine.</p>
Invoke	Yes or No for permission to invoke eMake.

The next nine columns display None, Read, Modify, or Full permission for the user/group:

Note: Modify permission allows Read and Write privileges. Full permission allows Read, Write, and Delete privileges. Permission selections affect which information/options appear in the web interface.

Column	Description
Builds	Permission to read, write, and/or delete build entries.
Agents	Permission to read, write, and/or delete agent entries.
Build Classes	Permission to read, write, and/or delete build class entries.
Messages	Permission to read, write, and/or delete message log entries.
Administration	Permission to read, write, and/or delete entries in the Administration section (users, groups, and so on).
Reports	Permission to access reports.
Resources	Permission to perform Resource Management activities (for example, build priorities, preemption policy).
User	None or Modify for permission to allow a user to change his/her own User Settings, Message Policies, filters, and Build Class notification.
Break Points	<p>Full or Modify Permission is available for users who must access break point functionality.</p> <p>Read allows users to view break point information in the ElectricAccelerator web interface.</p> <p>Modify allows users to interact with the break point. Users can click Agent Command and Shell Command, which update the break point by adding records to the database.</p> <p>Full allows users to perform all break point operations. Users can click Retry and Continue, which delete the current break point. If a break point launches, and the user stops the build by killing the process, break point information remains on the Build Details page, but the build is finished. Then there is a link to delete the break point manually.</p> <p>See the Using Breakpoints help topic.</p>
Actions	<p>Edit Permissions—Takes you to an edit page to change permissions for either a user or a group.</p> <p>Clear Permissions—Clears all permissions for a single user or group.</p>

Edit User Permissions

Immediately after the page title, Edit User Permissions, you can see the user name you selected for permission modifications.

Permissions control these respective features:

Permission	Description
Server Access	Are members permitted to access the server?
Impersonate user	Are members allowed to impersonate a different user to run Electric Make? See the Permissions help topic for more information.
Invoke EMake	Are members permitted to invoke Electric Make?
Builds	Are members permitted to read, write, and/or delete build entries?
Agents	Are members permitted to read, write, and/or delete agent entries?
Build Classes	Are members permitted to read, write, and/or delete build class entries?
Messages	Are members permitted to read, write, and/or delete message log entries?
Administration	Are members permitted to read, write, and/or delete entries in the Administration section (users, groups, and so on)?
Reports	Are members permitted to access reports?
Resources	Are members permitted to perform resource management (for example, build priorities, preemption policy)?
User	Are members permitted to change their own settings and filters?
Break Points	<p>Are members permitted to access, read, write, or delete break point feature/functions?</p> <p>Read allows users to view break point information in the web interface.</p> <p>Modify allows users to interact with the break point. Users can click Agent Command and Shell Command, which update the break point by putting records in the database.</p> <p>Full allows users to perform all break point operations. Users can click Retry and Continue, which delete the current break point. If a break point launches, and the user stops the build by killing the process, break point information remains on the Build Details page, but the build is finished. Then there is a link to delete the break point manually.</p> <p>See the Using Breakpoints help topic for information on using breakpoints.</p>

Click **OK** to save your changes for this user.

Enable User

Use this page to see your LDAP user list and select users who need access to ElectricAccelerator.

1. User Name Filter—In this field, type a letter and a wildcard character for faster searching to find users in your LDAP list.
For example, you might type "s*" to get a list of all user names that begin with "s."
2. Click the **Run Filter** button.
3. After your filtered appears, use the checkboxes in the first column to select users who need access to ElectricAccelerator.
4. After selecting users, click the **Enable Users** link at the top of the table to see these users added to the Permissions page.

When your LDAP users are included in the Permissions page Users table, you can assign required permissions to them for ElectricAccelerator access.

Edit Group Permissions

Immediately after the page title, Edit Group Permissions, you can see the group name you selected for permission modifications.

Permissions control these respective features:

Permission	Description
Server Access	Are members permitted to access the server?
Impersonate User	Are members allowed to impersonate a different user to run Electric Make? See the Permissions help topic for more information.
Invoke EMake	Are members permitted to invoke Electric Make?
Builds	Are members permitted to read, write, and/or delete build entries?
Agents	Are members permitted to read, write, and/or delete agent entries?
Build Classes	Are members permitted to read, write, and/or delete build class entries?
Messages	Are members permitted to read, write, and/or delete message log entries?
Administration	Are members permitted to read, write, and/or delete entries in the Administration section (users, groups, and so on)?
Reports	Are members permitted to access reports?
Resources	Are members permitted to perform resource management (for example, build priorities, preemption policy)?
User	Are members permitted to change their own settings and filters?
Break Points	<p>Are members permitted to access, read, write, or delete break point feature/functions?</p> <p>Read allows users to view break point information in the web interface.</p> <p>Modify allows users to interact with the break point. Users can click Agent Command and Shell Command, which update the break point by putting records in the database.</p> <p>Full allows users to perform all break point operations. Users can click Retry and Continue, which delete the current break point. If a break point launches, and the user stops the build by killing the process, break point information remains on the Build Details page, but the build is finished. Then there is a link to delete the break point manually.</p> <p>See the Using Breakpoints help topic for information on using breakpoints.</p>

Click **OK** to save your changes for this group.

Enable Group

Use this page to select groups, local or LDAP groups, who need access to ElectricAccelerator.

1. Select the appropriate checkboxes in the first column to select the groups that need access to ElectricAccelerator.
2. After selecting groups, click the **Enable Groups** link at the top of the table to see these groups added to the Permissions page.

When your LDAP groups are included in the Permissions page Groups table, you can assign required permissions to them for ElectricAccelerator access.

Users

Use this page to manage the user list. All local users created in ElectricAccelerator, are listed on this page. "admin" is the default ElectricAccelerator user. The admin user cannot be deleted and always has full-permissions (except for Impersonate and Invoke, which are No by default).

Note: If you have LDAP configured users who you want to have access to ElectricAccelerator, click the Permissions subtab, then the **Enable User** link.

Functionality

- Clicking a user name takes you to the user's User Details page.
- You can create a new user or delete existing users.
- The Actions column allows you to edit or delete the user in that row.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.

Column Descriptions

Column	Description
Name	This is the name of the user, recognized by the system. For example, if Mary Smith is the user's real name, her system user name might be "msmith." Clicking a user name takes you to the User Details or User Settings page, depending on whether you click your own user name or another user's name.
Real Name	This is the real name of the user. For example, the real name for the "msmith" user name might be Mary Smith.
Email	This is the normal email address used to send mail to this user.
Source	The source is Local for users created in ElectricAccelerator. LDAP users are listed on the Permissions page.
Actions	Edit User —Takes you to the corresponding Edit User page. Delete User —Deletes the corresponding user.

User Details

This page displays the user's name at the top of the page and shows that user's details. For a complete description of these fields, see [Users - create new or edit existing Local Users](#).

Users - Create or Edit Local Users

This page displays the user's name at the top of the page and also in the Name field in the Detail section.

Creating a User

1. On the Users page, click **New User**.
2. Type a unique name for a *local* user.
3. Complete the remaining fields.
4. Click **OK** to create the user.

Field Descriptions

Field	Description
Name	This is the name of the user, recognized by the system. For example, if Mary Smith is the user's real name, her system user name might be "msmith."
Full Name	This is the real name of the user. For example, the real name for the "msmith" user name might be Mary Smith.
Email	This is the normal email address used to send email to this user.
Password	Enter a secret value for this user account and re-type the value to confirm the new password.

Editing a User

1. On the Users page or User Details page, click **Edit User**.
The Edit User page appears.
2. Select and re-type the information you need to change or add new information.
For a description of all fields, see the preceding "Field Descriptions" section.
3. Click **OK** to save your modifications.
The user is added to the list on the Users page.

If this user is a member of a group, the Groups field displays the group name.

User Password

Type-in the new password and type it in again for confirmation.

Click **OK** to save the change.

Groups

Use this page to manage local user groups. All user groups created in ElectricAccelerator are listed on this page. "Administrators" is the default ElectricAccelerator user group. This group cannot be deleted and always has full permissions.

Suggestion: If you do not currently use groups, you may want to define groups in terms of development teams or your products.

After you create a group, you can assign permissions to that group. Each member of the group will have the same permissions selected for the group, which means when you add a new user to the group, that new user will automatically have all permissions granted to the group.

Note: If you have LDAP groups that need access to ElectricAccelerator, click the Permissions subtab, then the **Enable Group** link.

Functionality

- Clicking a group name takes you to the group's Group Details page.
- You can create a new group or delete existing groups.
- The Actions column allows you to edit, delete, or add a user to the group in that row.
- You can also:
 - Click a column heading to sort the information in that column.
 - Set the page refresh interval to keep your information current.

Column Descriptions

Column	Description
Name	This is the name of the group. Clicking a group name takes you to the Group Details page where you can see the list of members included in that group.
Source	The source is Local for groups created in ElectricAccelerator or LDAP/Active Directory for external groups.
Actions	Edit Group —Takes you to the corresponding Edit Group page. Add New User To Group —Takes you to the corresponding Add New User To Group page. Delete Group —Deletes the corresponding group.

Group Details

This page displays the name of the group and all members in that group.

Note: You may remove group members individually or remove multiple group members at the same time.

Group Members

This table lists all members who belong to this group.

To add a member, click the **Add New User To Group** link to go to that page.

Groups - Create or Edit a Local Group

Group Members

This table displays the members who belong to the group named on the page.

Creating a Group

1. On the Groups page, click **New Group**.
2. Type a unique name for a *local* group.
3. Click **OK** to create the group.

Editing a Group

1. On the Groups page or Group Details page, click **Edit Group**.
The Edit Group page appears.
2. Select and re-type the information you need to change or add new information.
3. Click **OK** to save your modifications.

See [Add New User to Group](#) for information about adding a user to the group.

Removing Users From a Group

1. Select the checkboxes in the first column for the users you want to remove from the group.
2. Click **Remove Users From Group**.

Add New User to Group

Use this page to filter user names to add to your selected group.

The group name you selected appears at the top of the table. You can add local or LDAP users to local groups, but ElectricAccelerator **cannot** add users to LDAP groups—you must use LDAP to add users to LDAP groups.

Adding a User to a Group

1. In the User Name Filter field, type a letter and a wildcard character for faster searching to find users in your LDAP list.

For example, you might type "s*" to get a list of all user names that begin with "s."

2. Click the **Run Filter** button.

This displays your filtered list.

3. Use the checkboxes in the first column to select users who need access to ElectricAccelerator.
4. Click the **Add Users To Group** link at the top of the table.

Now the selected users are added to the group.

If the group already has assigned permissions, all new group members will automatically have the same permissions.

User Settings

This page opens to the currently logged in user's User Settings page.

Editing User Settings

If you have edit permissions, you can change user settings:

1. Go to the Users page and click **Edit User** in the appropriate table row.

Note: To confirm you have edit permissions, the web page title will be Edit User Settings, not User Settings.

2. Make any necessary changes.

If you need to change the password, click **Edit Password** [at the top of the page] to go to the User Password page.

3. Click **OK** to save user setting changes.

Field Descriptions

Field	Description
Name	This is the name of the currently logged-in user whose settings can be edited.
Full Name	This is the full real-world name of the user. For example, Mary Jane Smith.
EMail	This is the email address assigned to the user.
Groups	These are the groups where the user is a member.

Server Settings

Use this page to set up your server preferences for resource management and notifications.

Field Descriptions

Field	Description
Default Priority	This is the class build priority level. Classes have default priorities and boost values. Boost values range from -10 to +10 (default=0). A higher boost value means builds in that class have preference over builds with the same priority and lower boost [for using available agents]. If you change the default settings, each new build class you create will have the "new" default values you specified on this page. However, you may change these settings individually for each build class on the Edit Build Class or New Build Class page.
Resource Manager Type	<p>Select none if you do not want the Cluster Manager to use a resource manager.</p> <p>For information about how to set up and use Priority Pools, review Priority Pools information.</p> <ul style="list-style-type: none"> • If you choose to use a resource manager, a new Resources subtab is available to you under the Agents tab. Conversely, if you select "none," you will not see or have access to the Resources subtab. • If you cannot select the grid radio button because it is "grayed-out," go to the <i>ElectricAccelerator Installation and Configuration Guide</i>, and review the "Enabling Resource Management" section in the Initial Configuration Tasks chapter, and the Grid chapter.
Resource Statistics Interval	In minute units, this is the interval to collect statistics on resource usage.
Resource Statistics Keep	This is the number of days of Resource usage statistics to keep.
Message Log Keep	This is the number of days to keep message log entries. You may want to keep 30 days of message logs, but monitor the size of your message logs to make sure you have enough storage space available.
Email Interval	The number of minutes between email notifications.
Email Item Limit	The maximum number of messages an email notification can contain.
Email From	The value to use in the From header element for the email notification.
Email Prefix	A string used to prefix subject lines.

Note: A field marked with an asterisk denotes a required field.

Click **OK** to save your settings.

About Resource Managers

The ElectricAccelerator resource management feature allows you to select a subset of available agents when running builds. For example, you might have a build that requires a certain hardware feature. If so, ElectricAccelerator must use agents only on hosts containing that hardware.

The following resource managers are available through the ElectricAccelerator web UI:

- built in—This resource manager is integrated with ElectricAccelerator and provides a straight forward way to associate hosts with a resource name.
- grid—This resource manager uses grid management software to select agents for your build, see the Grid chapter in the *ElectricAccelerator Installation and Configuration Guide*.
- priority pools—This resource manager enables you to group resources into pools that can be prioritized differently among groups. See the [Priority Pools](#) topic for details.

To select a resource manager for your build, go to Administration > Server Settings. You can enable an additional resource manager through cmtool only.

Note: eMake automatically selects agents running on a compatible operating system. The resource manager does not override this feature, which means it is possible to select a resource definition that contains incompatible hosts. Likewise, a resource definition can contain hosts with incompatible operating systems. In this case, ElectricAccelerator selects only those hosts with the correct operating system that satisfies the resource definition.

Database Configuration

Configuring Accelerator to Use an External Database

Before converting to an external database using MariaDB, Microsoft SQL Server, MySQL, or Oracle, ensure the database is up and running correctly. (Electric Cloud bundles MariaDB and installs it as the default local database on the Cluster Manager.)

- For Oracle, ensure that when you create a user, Roles has **Resource** added.
- For Microsoft SQL Server, ensure that when you create a user, Server Roles has **sysadmin** (or another server role that allows the necessary actions) enabled.

1. Fill in the fields as follows:

- Database Name—Database instance name.
- Database Type—MariaDB, Microsoft SQL Server, MySQL, or Oracle.
- Host Name—Machine name where the database is installed.
- Port—Database port number. The default port is filled in automatically after Database Type is selected. Type a different port number if applicable.
- User Name—Unique name of the user that is used to access the database.
- Password and Confirm—Secret value that identifies an account for a particular user.

The Cluster Manager tests the database connection before saving `database.properties`. If the connection is unsuccessful, an error appears, and `database.properties` is not saved.

2. Re-import your license.
3. Restart the Cluster Manager service.

Note: After changing the database type, your original database service might continue to start each time the Cluster Manager service starts. Manually disable the database service.

Backing Up and Restoring a Database

You use the `exportData` and `importData` `cmtool` commands to back up and restore a database.

Note: Manual migration using the `exportData` and `importData` commands is recommended only to replicate data between Cluster Manager instances running the same version of the Cluster Manager. For example, you could use it to make a backup dump and then restore it to a new instance of the Cluster Manager of the same version.

Backing Up Your Database

Run the following command to export the Cluster Manager data to a file:

```
cmtool exportData <file_name>
```

where `<file_name>` is the file name or path to export to. If you use a file name, the destination is the current working directory of the Java process (`/opt/ecloud/i686_Linux` or `C:\ECloud\i686_win32` by default. If you use a path, the Cluster Manager Java user (`eacmuser`) must have execute and write access to that path.

Note: This is a full database dump, which might take substantial time for a large database.

Restoring Your Database

Run the following command to import the Cluster Manager data from a file:

```
cmtool importData <file_name>
```

where *<file_name>* is the name of the file to import. The file's path is relative to the current working directory of the Java process (/opt/eccloud/i686_Linux or C:\ECloud\i686_win32 by default).

Note: This command imports a full database dump, which might take substantial time for a large database.

Make sure that you

- manually delete any old or unused agents from the agents list.
- update the license file after import, if it has previously expired.

Licenses

This page displays all license information known to the Cluster Manager and shows the usage to which you are entitled. The Cluster Manager reads this information from the current license file. This page typically displays a single license.

Note: You must re-import your license after changing database types.

Column Descriptions

Column	Description
Product Name	Product name as listed in the current license file. For ElectricAccelerator installations, the value is <code>ElectricAccelerator</code> .
Feature Name	Feature name as listed in the current license file. For ElectricAccelerator installations, the value is <code>Server</code> .
Customer Name	Name of the organization that is granted the license. (If the name is incorrect, contact Electric Cloud technical support.)
Grace Period	Number of days that you can use ElectricAccelerator after the expiration date. However, you will be warned frequently during the grace period that your license has expired.
Expiration Date	Last day on which this license is valid. After your license expires (and after any grace period), you might be able to use certain ElectricAccelerator features, but you cannot run a build. If this field is blank, your license is perpetual.
Max Agents	Maximum number of agents that the current license file allows.
Jobcache Max	Maximum number of builds that are licensed to use the JobCache feature simultaneously. This number is read from the <code>jobcacheMax</code> property in the current license file. (JobCache is an add-on option. It is available only on Linux and Solaris platforms and requires separately-purchased licensing.) Simultaneous builds in excess of this number will occur without using this feature.
Actions	View License —Opens the License Details page. Delete License —Deletes the ElectricAccelerator license. (A popup appears that asks you to confirm the deletion.)

Import License

Use this page to provide a license to the ElectricAccelerator server.

You should already have received a license from Electric Cloud. If you did not receive a license, contact Electric Cloud Customer Support or your sales representative.

Importing a License

Do one of the following:

- Browse for and upload the license file.
- Follow these steps:
 1. Use a text editor to open the license file.
 2. Copy the text into the Data field.
 3. Click **OK**.
 4. Go back to the License page to see your license information.

Cloud

Use this page to track jobs running on grid machines. To use this page, you must have already configured the Cluster Manager server for grid management software. If you have not completed the grid configuration process, see the Configuring LSF section in the *ElectricAccelerator Installation and Configuration Guide*.

The Cluster Manager is the only ElectricAccelerator component that communicates directly with grid management software.

CloudHostManager Information

This section contains messages about actions performed on the grid.

Current Jobs

Column heading descriptions:

Column	Description
Job ID	The grid assigns a job ID for each job that runs.
Job Status	The job status is returned from the grid. Status can be PEND, RUNNING, EXIT, DONE, or UNKNOWN.
Host Name	The name of the grid host running the <code>agentconnect</code> script to communicate with the ElectricAccelerator Cluster Manager.
Platform	The platform of the grid host running the <code>agentconnect</code> script.
Resource Request	If you did not define a resource in your ElectricAccelerator build class, the grid master host chooses any available server and the column will be blank (no name displayed). If you defined a resource name in the ElectricAccelerator build class, this name is used by the grid master host to choose servers to run the job.
Stalled	Indicates if the job is stalled.
Submitted	This is the date and time the grid job was submitted.

Error Codes

Error code documentation has moved and is now available exclusively online. See the *ElectricAccelerator Error Messages Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html. The guide provides descriptions and troubleshooting information for common error messages in ElectricAccelerator.