

ElectricAccelerator Help

Version 10.0

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

ElectricAccelerator 10.0

Copyright © 2002–2017 Electric Cloud, Inc. All rights reserved.

Published 12/11/2017

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, ElectricFlow Deploy, ElectricFlow Release, ElectricInsight, and Electric Make are registered trademarks or trademarks of Electric Cloud, Incorporated.

Electric Cloud products—ElectricAccelerator, ElectricAccelerator Huddle, ElectricCommander, ElectricFlow, ElectricFlow Deploy, ElectricFlow Release, ElectricInsight, and Electric Make—are commonly referred to by their “short names”—Accelerator, Huddle, Commander, Flow, Deploy, Release, 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)	2
ElectricAccelerator Agent (Agent)	2
Cluster Manager	2
Electrify	2
Cluster Manager and Other Components	2
When to Use the ElectricAccelerator Cluster	3
LSF or SGE 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	16
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	27
Stopping, Starting, or Restarting the Daemon	27
Checking Whether the Daemon is Running	27
Using econfig	28

Using ecagent Tools	30
Using ecinstconf	31
Using eclock	34
Using the runagent.local File	36
Preventing Runaway Jobs	36
Configuring Stalled Job Detection Settings	36
Configuring Agent Log Rotation	37
System Maintenance Tasks	39
Database Configuration	40
Enabling LDAP	41
Import License	42
Importing a License	42
Using Comments	43
Adding a New Comment	43
Editing a Comment	43
Deleting a Comment	44
Stopping Builds	45
Stopping a Single Build from the ElectricAccelerator Web Interface	45
Stopping Multiple Builds from the Web Interface	45
Stopping a Build Using cmttool	45
Deleting Builds	47
Deleting a Single Build	47
Deleting a Group of Builds from the Build List	47
Creating Build Classes	48
Build Class Examples	48
Editing Build Classes	50
Deleting Build Classes	51
Deleting a Single Build Class	51
Deleting Multiple Build Classes	51
Agent Policies	52
Agent Allocation Policy	52
Preemption Policy	52
Agent Lock Interval (seconds)	54
Deep/Wide Agent Allocation Policy	54
Understanding a Cluster Sharing Policy	55
Modifying the Number of Agents per Host	56
Enabling Agents	57
Enabling a Single Agent	57

Enabling a Group of Agents	57
Disabling Agents	58
Disabling a Single Agent	58
Disabling a Group of Agents	58
Testing Agents	59
Checking the Status of a Single Agent	59
Checking the Status of a Group of Agents	59
Deleting Agents	60
Deleting a Single Agent	60
Deleting a Group of Agents from the Agent List	60
Removing an Agent from the Cluster Permanently	60
Creating Resources	61
Host Name and Glob-Style Pattern Example 1	61
Host Name and Glob-Style Pattern Example 2	61
Editing Resources	63
Deleting Resources	64
Deleting a Single Resource	64
Deleting Multiple Resources	64
Priority Pools	65
Using Priority Pools	65
Use Case 1—High Performance Builds	65
Use Case 2—Multiple Departments	66
Creating Filters	68
Editing Filters	70
Deleting a Filter	71
Generating Reports	72
Adding Custom Reports	73
Making Custom Reports Available to the Cluster Manager	73
Known Issues	73
The Home Page	74
Banner	74
Tabs	76
News Feed	76
Online Resources	76
Lightning Lessons	76
Agent Usage (last 24 hours) Report	76
Build Duration Report	76
Builds	77

Functionality	77
Column Descriptions	77
Build Details	79
Actions	79
Build Summary	80
Details Pane	80
Configuration Pane	81
Metrics Pane	81
Build Classes	82
About Build Classes	82
Functionality	82
Column Descriptions	82
Build Class Details	84
Functionality	84
Details	84
Comments	84
Build Classes- Create or Edit a Build Class	85
Field Descriptions	85
Agents	89
Functionality	89
Column Descriptions	89
Agent Details	91
Functionality	91
Status	91
Configuration	91
Agent Console	91
Comments	91
Agent Policies	92
Agent Allocation Policy	92
Preemption Policy	92
Agent Lock Interval (seconds)	94
Deep/Wide Agent Allocation Policy	94
Resources	95
Functionality	95
Column Descriptions	95
Resource Details	96
Functionality	96
Details	96

Comments	96
Resources - Create or Edit a Resource	97
Field Descriptions	97
Reports	98
Checking Your Permissions to Run Reports	99
Using Permanent Links to Reports	99
Downloading Report Data	99
Agent Usage Report	99
Build Duration Report	101
Build Duration vs. Conflict Report	106
Build Duration vs. Job Count Report	107
Build Summary Report	108
Build Wait Time Report	109
Cluster Utilization Report	113
Concurrent Builds Report	115
Effective Agent Allocation Summary Report	118
Effective Agent Allocation Per Day Report	120
JobCache License Allocation Report	121
JobCache Usage Summary Report	124
Resource Usage over Time Report	126
Sea Level Report	127
Server Load	128
About Server Load	128
Functionality	128
Column Descriptions	128
Realtime Server Metrics	130
Messages	131
	131
Message Policies	132
Administration	133
Filters	134
Saved Filters Panel	134
Functionality	134
Lists that Do Not Support Filters	134
Filters - Create or Edit a Filter	136
Permissions	137
Functionality	137
Column Descriptions	137

Edit User Permissions	142
Enable User	144
Edit Group Permissions	145
Enable Group	147
Users	148
Functionality	148
Column Descriptions	148
User Details	148
Users - Create or Edit Local Users	149
Creating a User	149
Field Descriptions	149
Editing a User	149
User Password	150
Groups	151
Functionality	151
Column Descriptions	151
Group Details	151
Group Members	152
Groups - Create or Edit a Local Group	153
Group Members	153
Creating a Group	153
Editing a Group	153
Removing Users From a Group	153
Add New User to Group	154
Adding a User to a Group	154
User Settings	155
Editing User Settings	155
Field Descriptions	155
Server Settings	156
Field and Menu Descriptions	156
About Resource Managers	157
Database Configuration	160
Licenses	161
License File Example	161
License File Property Descriptions	161
Import License	163
Importing a License	163
Cloud	164

Resource Manager Example	164
Resource Manager Information Section	165
Current Jobs or Instances Section	166
Error Codes	168

Welcome to ElectricAccelerator

About ElectricAccelerator

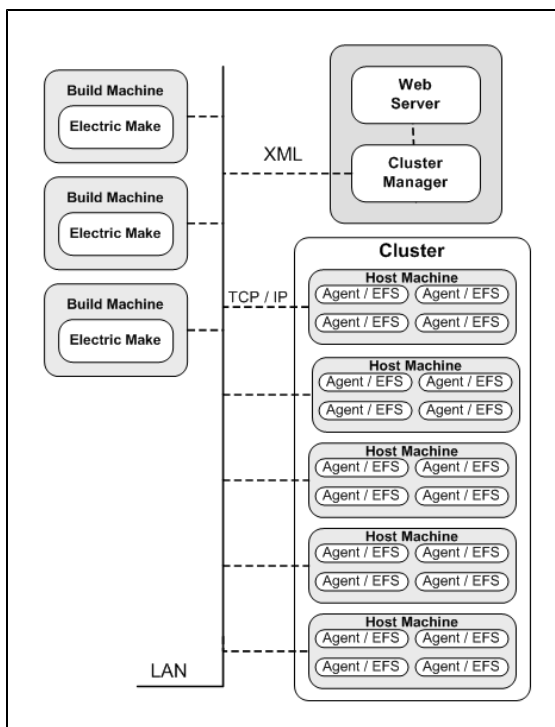
ElectricAccelerator® (“Accelerator”) is a software build accelerator that dramatically reduces software build times by distributing the build over a large cluster of inexpensive servers. Using a patented dependency management system, ElectricAccelerator identifies and fixes problems in real time that 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 (agents)
- 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.

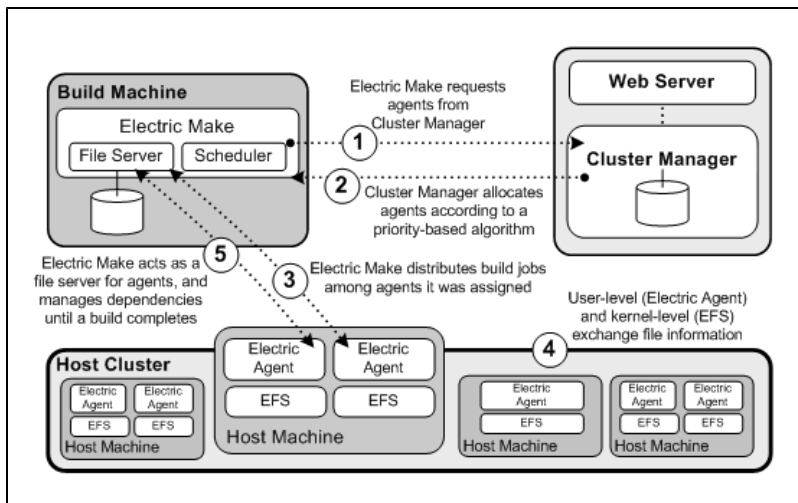
Cluster Manager also allows users to view upcoming and past builds, view the status of agents in the cluster, and run reports.

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. For details about Electrify, see the *ElectricAccelerator Electric Make User Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.

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 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.

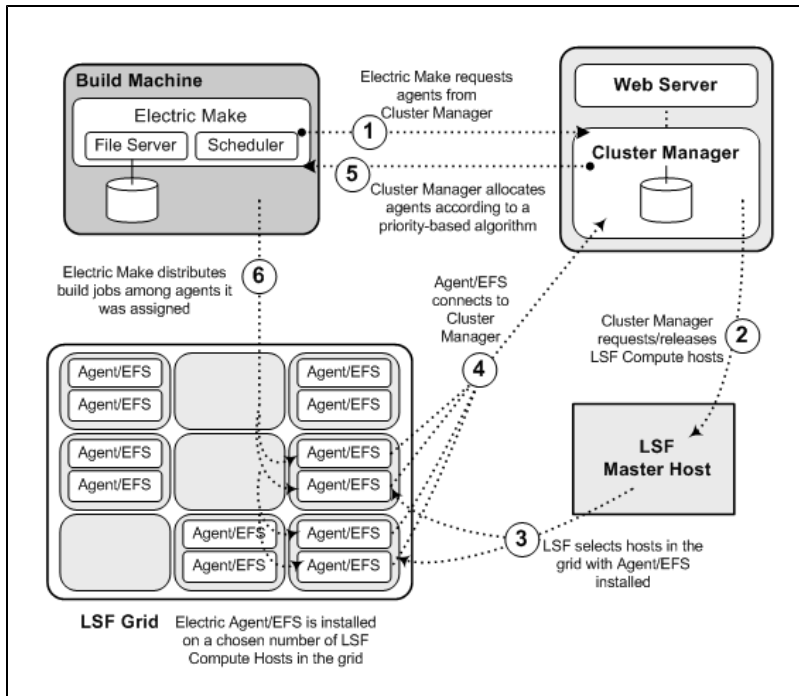
LSF or SGE Grid Integration

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

- 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 LSF or SGE 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 file names 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
<code>clusterdownload</code>	cm, agent, emake	Downloads files to the host. See Using Cluster Manager Administration Tools .
<code>clusterexec</code>	cm, agent, emake	Runs commands on agent hosts. See Using Cluster Manager Administration Tools .
<code>clusterupload</code>	cm, agent, emake	Uploads files to agent hosts. See Using Cluster Manager Administration Tools .
<code>cmtool</code>	cm, agent, emake	Command-line tool for Cluster Manager.
<code>ecagent_restart</code> <code>/etc/init.d/ecagent restart</code>	agent	Restarts the agent service or daemon. See Using ecagent Tools .
<code>ecagent_start</code> <code>/etc/init.d/ecagent start</code>	agent	Starts the agent service or daemon. See Using ecagent Tools .
<code>ecagent_stop</code> <code>/etc/init.d/ecagent stop</code>	agent	Stops the agent service or daemon. See Using ecagent Tools .
<code>ecbreakpoint</code>	agent	Debugging tool for problematic job steps and stalled processes during the build.
<code>eccc</code>	cm, agent, emake	A wrapper for SunForte Compiler Suite and Sun ONE Studio/Sun Workshop if you are experiencing excessive serialization in your build.

File	Location	Description
ecconfig	agent	Reconfigures agents. See Using ecconfig .
ecinstconf	cm, agent	Configures Cluster Manager and agent settings. See Using ecinstconf .
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 eMake component. For details about eMake, see the “Electric Make Command-Line Options, Environment Variables, and Configuration File” section in the <i>ElectricAccelerator Electric Make User Guide</i> at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html .

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>	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>	Label for the breakpoint. This is used in the web UI to identify the breakpoint
<code>--port=<value></code>	HTTP port to use to contact this application
<code>--printLabel</code>	Print the label when the breakpoint is reached
<code>--timeout=<value></code>	Number of seconds after which the breakpoint continues. The 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 breakpoint 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`.

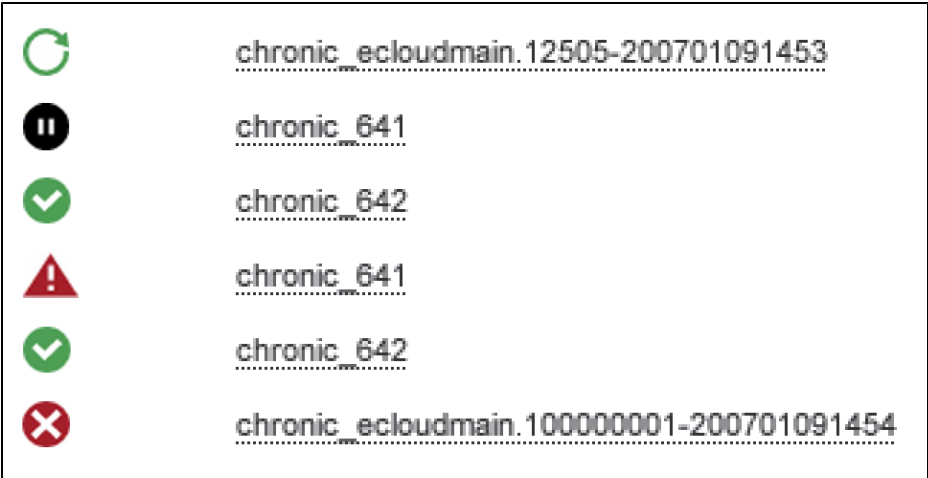


Figure 2: Paused job

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, the 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 `pstree -alp` was passed to the agent. The `pstree` command shows a hierarchical representation of the process tree and is often useful to see what commands agents are running. Note that `pstree` 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 `pstree`. 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: pstree -alp

Agent CommandShell Command

Type	Request	Response	Requested By	Created Time
S	pstree -alp	init,1 -acpid,2214 -agent,2324 /opt/ecloud/i686_Linux/bin/runagent -cm support-lin1:8030 -efsid 0 -numagents 4 -secureconsole n-d -execserver,1231 -sleep,1236 60 -agent,2341 /opt/ecloud/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/ecloud/i686_Linux/bin/runagent -cm support-lin1:8030 -efsid 2 -numagents 4 -secureconsole n-d -execserver,1237 -sleep,1241 60 -agent,2375 /opt/ecloud/i686_Linux/bin/runagent -cm support-lin1:8030 -efsid 3 -numagents 4 -secureconsole n-d -execserver,31813 -sleep,1222 60	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. For contact information, see <http://electric-cloud.com/support/>.

Type	Request	Response	Requested By	Created Time
A	session state	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	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 `ecloud 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.

Following is an 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.
<code>exec.xml</code>	Execute the specified command. The command is one simple command passed with the command GET parameter.
<code>agent.xml</code>	Execute the specified agent command. The command is one simple command passed with the command GET parameter.
<code>agentstat.xml</code>	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 Command	
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 Command	

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 `cmtool runAgentCmd` program to configure agent-side breakpoints (Figure 1).

```
cmtool runAgentCmd 'session breakpoint set {trigger_code} {command_to_execute}'
cmtool runAgentCmd 'session breakpoint get'
cmtool 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, <code>mergestreams</code> 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`—Lets you run arbitrary commands on a host machine or all hosts in a cluster
- `clusterupload`—Lets you upload executables and files (for example, compiler, libraries, and so on) to all hosts in a cluster
- `clusterdownload`—Lets you 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 backslash (\) 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</code> , <code>--help</code>	Prints a usage message summarizing information in this table

Switch Options	Description
--hosts=<value>	Set of hosts on which to run commands. <i>value</i> should be in the form <code>host1 host2</code>
--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, commands run on all hosts in 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 and --hosts 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
--good-hosts-only	Only use hosts with at least one pingable enabled agent in Cluster Manager. This option applies only when --cm is specified. Default behavior uses all hosts regardless of their state
--file=<value>	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
-k, --keep-going	Continue running commands after failure. If more than one command is specified, continue to run subsequent commands, even if an earlier command fails
--mergestreams=<0/1>	Default = true (merge <code>stdout</code> and <code>stderr</code> commands and write to <code>clusterexec stdout</code> stream). If set to false, <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
-v, --version	Displays <code>clusterexec</code> version
-s, --use-shell	Send the entire command string to the shell (<code>sh</code> on UNIX, <code>cmd</code> on Windows)
--timeout=<value>	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 agent service on the hosts
<code>startAgent</code>	Starts the agent service on the hosts
<code>restartAgent</code>	Restarts the 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>	Shuts down eRunner service on the hosts
<code>restartEranner</code>	Restarts eRunner service on the hosts
<code>printVersion</code>	Prints 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 <code>host1 host2[:port] ...</code>
<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 windows, solaris, or linux. 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

Switch Options	Description
<code>-r, --recursive</code>	Recurse into subdirectories
<code>-u, --update</code>	Update only. Do not overwrite newer files. File modification times are used
<code>-t, --times</code>	Preserve file timestamps
<code>-n, --dry-run</code>	Show what would have been transferred
<code>--existing</code>	Only update existing files
<code>--progress</code>	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
<code>--exclude=<pattern></code>	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
<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.

Option	Description
<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. For contact information, see <http://electric-cloud.com/support/>.

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 or set the agent Cluster Manager host
-agentcmport	Get or set the agent Cluster Manager port
-agents	Get or set the agent number
-agentreboot	Get or set the agent reboot selection
-agentdrive	Get or set the agent drive letter
-agentinfo1	Get or set the agent info
-agentinfo2	Get or set the agent info
-agentlogremove	Get or set the agent log removal
-cmport	Get or set the Cluster Manager listening port
-cmserverport	Get or set the Cluster Manager accelerator server listening port
-cmphpserverport	Get or set the accelerator server port that the Cluster Manager web UI should use (startup.php)
-cmacceleratoradmin	Get or set the accelerator server admin username
-cmserveradmin	Get or set the Cluster Manager admin mail address
-cmservername	Get or set the Cluster Manager name
-cmdbport	Get or set the Cluster Manager database port
-cmmaxclients	Get or set the cluster manager MaxClients setting, which controls the number of simultaneous requests the Cluster Manager will handle
-cmmaxrequestspchild	Get or set the cluster manager MaxRequestsPerChild setting

-cmrotatelogerror	Get or set the cluster manager error log rotation
-cmrotatelogcustom	Get or set the cluster manager custom log rotation
-cmmigratedb	Get or set the cluster manager database migration
-cmstatssoft	Get or set the cluster manager stats soft limit
-cmstatshard	Get or set the cluster manager stats hard limit
-cmstrictusers	Get or set the cluster manager strict user check
-cmallowcookies	Get or set the cluster manager cookie permission
-cmcookieexpiration	Get or set the cluster manager cookie expiration time
-cmcookiedomain	Get or set the cluster manager cookie domain
-cmemakeheartbeat	Get or set the cluster manager emake timeout
-cmconntimeout	Get or set the cluster manager connection timeout
-cmwebtimeout	Get or set the cluster manager web timeout
-cmwebmaxattempts	Get or set the cluster manager web max attempts
-cmmaxagents	Get or set the cluster manager max agents
-cmmaxbuilds	Get or set the cluster manager max builds
-cmversionmismatch	Get or set the cluster manager version mismatch file
-cmcharset	Get or set the cluster manager character set
-cmlogfilepath	Get or set the cluster manager build log file path
-cmlogfileprocessor	Get or set the cluster manager log file processor
-cmlogremove	Get or set the cluster manager log removal
-upgradehost	Get or set the agent upgrade host
-upgradeport	Get or set the agent upgrade port

-dbuser	Get or set the database root user
-dbpassword	Get or set the database root password
-dbbasedir	Get or set the database base directory
-dbdatadir	Get or set the database directory
-dbtmpdir	Get or set the database tmp director
-dbsocket	Get or set the database socket path
-dbserve	Get or set the database server path
-dbcharset	Get or set the database charset

Using eclock

Eclock is a tool that lets you 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 `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 lets you 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 or set the maximum file size for log files before they are rotated. The default is 1 GB.
<code>filename <path></code>	Get or 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 or 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.

Database Configuration

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.

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 technical support or your sales representative. For technical support contact information, see <http://electric-cloud.com/support/>.

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

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 pages:

- **Home**—While comments on this page are for server-specific information, you can use this section for other messages or notes. For example, system-wide notices, schedule announcements, and task assignments. **Home** page comments are available immediately each time you log in or are 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.

The **New Comment** page appears.

2. Type 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 comment. The solution is:

1. From the command line, 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`

If you retype: `cmtool getServerComments`, you will not see the offending comment. Any other comments already in the comment box are not affected, and you can add more comments.

Editing a Comment

Click **Edit Comment** to update a comment.

Deleting a Comment

Click **Delete Comment** to delete a comment.

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 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 option for those who are 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 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**.

Deleting a Group of Builds from the Build List

1. Go to the **Filter** dropdown menu and select the 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 opens 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 completed.

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 reselect 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 results 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 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 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

Preemption Policy Setting	Reassign the agent to A from B?	
	Unlocked or Lock Interval = 0	Lock Interval > 0 and time elapsed exceeds interval
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 want 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 agents list, a  (alive) icon in the **Status** column indicates an agent is alive. A  (not alive) icon indicates the agent is not responding, has timed out, or has been disabled. At any time, you can test one or all agents to update the status list.

Checking the Status of a Single Agent

Either Click the  (test agent) button in the **Status** column in the agents list or click the  button on the **Agent Details** page to test that agent.

Checking the Status of a Group of Agents

1. From the **Filter** drop-down menu, select the agent criteria.
2. Click the  (test filtered agents) button in the **Filtered Agents** section of the page.

After a few seconds, a  icon in the **Status** column appears to indicate an agent is alive. A  icon indicates the agent is not responding or has timed out.

3. If necessary, click the **View Messages** link in the **Actions** column to help you determine what the error icon means.
4. Click the **Agents** tab to return to the agents 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 a 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
- 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 opens 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. (Optional) Add a description.
4. Click **Add Host Mask Pattern**.

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

5. Click **OK** when all fields are completed.

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. Retype or reselect the information you want to change.

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

3. Click **OK**.

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 used by any build, but builds originating from a pool's "owner" always have first priority to use that pool's resources. Using Priority Pools lets you 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>": "</code> or <code>""</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).

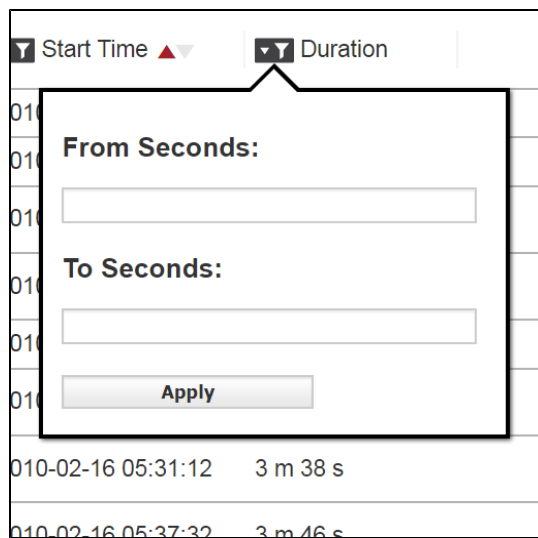
Creating Filters

To create a filter:



1. On the **Saved Filters** panel on a page that supports filtering, click the (column filtering) button at the top of a column whose values you want to filter.

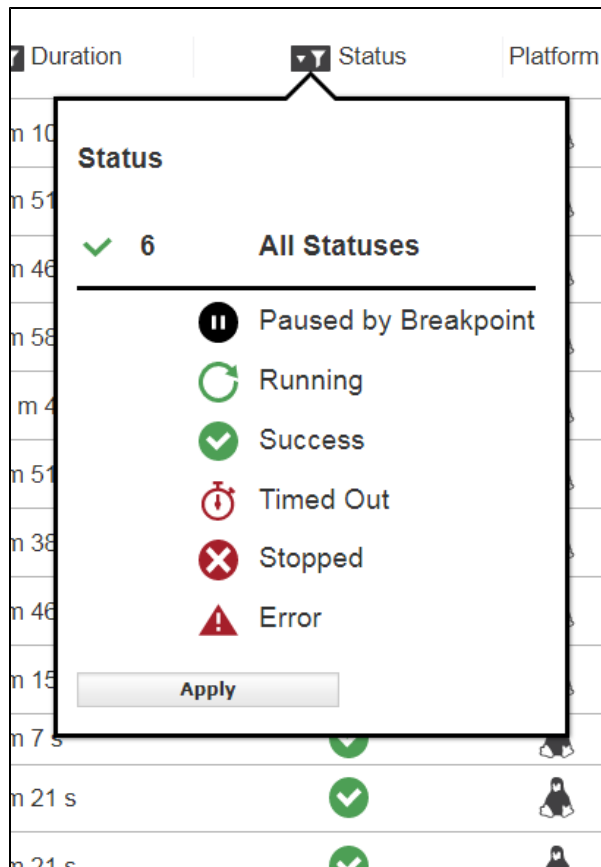
A dialog box appears that lets you select or enter the filtering specifics. For example, for the Duration column, a dialog box appears that lets you filter out builds whose durations (in seconds) are outside of a specific range:

A screenshot of a web application showing a table with columns 'Start Time' and 'Duration'. A dialog box is open over the 'Duration' column header. The dialog box has a title 'From Seconds:' with a text input field, a title 'To Seconds:' with a text input field, and an 'Apply' button at the bottom. The table below shows two rows of data: '010-02-16 05:31:12 3 m 38 s' and '010-02-16 05:37:32 3 m 46 s'.

Start Time	Duration
010-02-16 05:31:12	3 m 38 s
010-02-16 05:37:32	3 m 46 s

Also, for example, for the Status column, a dialog box appears that lets you filter out builds possessing

one or more specific statuses (such as builds that are stopped or timed out):



2. Repeat the prior step for additional columns as needed.

Save

3. Type a name for the filter into the **Save** field and then click the button. The filter is added to the **Saved Filters** list.

Editing Filters

To edit a filter:

1. Open the tab containing the filter that you want to modify.
For example, open **Builds > Build Classes**.
2. From the **Saved Filters** drop-down menu, choose the filter.



3. Click the (modify filtering) button for the column for which you want to filter or refilter.



For example, click the button at the top of the **Duration** column in the **Build Classes** tab. The filtering details dialog box appears, which lets you filter out builds whose durations (in seconds) are outside of a specific range:

Start Time ▲ ▼ Duration ▼

From Seconds:

To Seconds:

Apply

010-02-16 05:31:12	3 m 38 s
010-02-16 05:37:32	3 m 46 s

4. Enter or modify the filtering details in the dialog box and then click **Apply**.
See the [“Filters - Create or Edit a Filter”](#) section for field definitions.
5. Click **OK**.

The table is refiltered, and the filter changes are saved.

Deleting a Filter

1. Open the tab containing the filter that you want to delete.
For example, open **Builds > Build Classes**.
2. From the **Saved Filters** drop-down menu, choose the filter.

A rectangular button with a light gray background and a thin black border. The word "Delete" is centered on the button in a black, sans-serif font.

3. In the **Saved Filters** panel, click the button.
A popup that asks **Are you sure you want to delete this saved filter?** appears.
4. Click **Delete <filter_name>** to confirm.
The filter is deleted and no longer appears in the **Saved Filters** menu.

Generating Reports

1. Click the **Reports > Reports** subtab.

2. From the drop-down menu, choose the report that you want to generate.
3. Click the **From** field and choose the report start date from the calendar that appears.
4. Click the **To** field and choose the report end date from the calendar that appears.
5. (Optional) Complete any additional fields to customize the report.
6. (Optional) Customize the **Ranges** field.

If available, the **Ranges** field lets you customize which ranges that 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. [or] brackets denote that the number is inclusive. (or) brackets denote that the number is exclusive.

See [Reports](#) for more information.

7. 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.

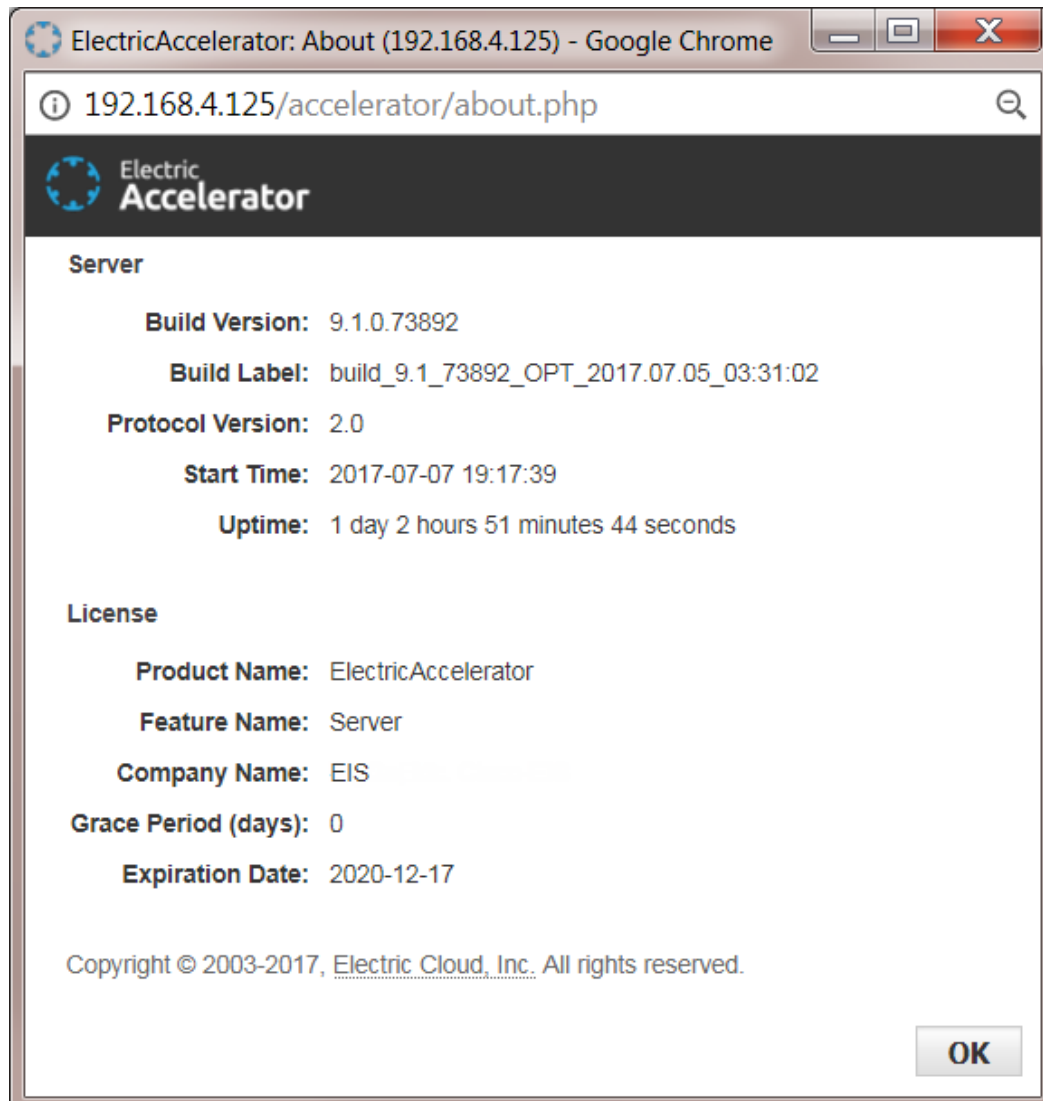
The Home Page

The **Home** page provides a news feed, links to documentation and other assistance, instruction for novice users, and real-time charts for agent usage and build durations during the past 24 hours.

Banner

The banner provides the following links:

- **Logout**—Logs out the current user
- **About**—Displays information about the Cluster Manager server (such as the product version) and information about the Cluster Manager license (such as the expiration date and the grace period for renewal)



- **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.

The banner also provides the following key statistics about the cluster:

- **Agents**—Number of agents that are currently running in the cluster
- **Running builds**—Number of builds that are running currently in the cluster

- **CPU Hours Used**—Total hours of CPU time used by all builds on all agent machines in the cluster since it was established
- **Developer Hours Saved**—Total hours of build time saved versus builds without build acceleration since the cluster was established
- **Days Remaining**—Number of days remaining in the Cluster Manager license

Tabs

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.

News Feed

This section provides information about new feature releases and patch releases as well as links to the release notes and software download information. If there are no new versions, information about the current release is displayed.

Online Resources

This section provides online resources for documentation and other forms of assistance such as the ElectricAccelerator Knowledge Base.

Lightning Lessons

This section provides a quick start guide and instructional videos for beginning users.

Agent Usage (last 24 hours) 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. For details, see the “[Agent Usage Report](#)” section.

Build Duration Report

This report is a scatter plot that shows the durations of builds over a specified time period. For details, see the “[Build Duration Report](#)” section.

Builds





The **Builds** page shows:

- Running and completed builds
- Status for each build (such as successful or ended because of an error)
- Additional build information

Functionality

















- Click a build name to open that build's [Build Details](#) page, which provides detailed information about the build, including access to the Build Log.

Note: To configure the **Builds** page so that users can see only their own builds, see the [Permissions](#) topic.

- After [filtering builds](#), you can then [stop](#) or [delete](#) the builds in the filtered results by clicking the  **Stop** or  **Delete** button respectively
- The **Actions** column lets you [delete](#) or [stop](#) a build or view its build-related messages by clicking the  or  button respectively
- You can also:
 - Click a column heading to sort the information in that column
 - Set the page refresh interval to keep the information current
 - Set the number of records you want to see per page

Column Descriptions






Column	Description
ID	Build ID number assigned by the Cluster Manager
Name	Build name as defined by the build class tag definition
Class	Build class containing the build. A default build class name is displayed if one is not assigned
User	User who started the build and the machine where the build was invoked. (Mouse over the field to see the IP address)
Start Time	Date and time when the build was started
Duration	Build duration (the elapsed time between the start and end time)

Column	Description
Status	<p>Build's result or status:</p> <p> Paused by a breakpoint</p> <p> Running</p> <p> Completed successfully</p> <p> 100 %</p> <p>Agent availability (using a percentage bar) during a running build. On mouse over, Requested indicates the number of separate build steps for which eMake identified and requested agents</p> <p> Timed out. For details about possible causes, see KB article KBEA-00039 "Fixing builds that the Cluster Manager reports as timed out."</p> <p> Stopped manually</p> <p> Failed or encountered an error</p> <p> Running but waiting for a build license. This appears when the number of concurrent builds has reached the license limit. The build will resume when another build stops running for any reason (because it completed successfully, completed with an error, was stopped from the Cluster Manager, timed out, and so on).</p>
Platform	<p>Operating system platform of the machine running the build:</p> <p> Linux</p> <p> Windows</p>
Priority	<p>Build priority for acquiring agents when the build runs:</p> <p> High (three green bars)</p> <p> Normal (two yellow and one white bar)</p> <p> Low (one blue and two white bars)</p>
Actions	<p> Opens the Messages page for viewing all system log entries for the build</p> <p> Deletes a non-running build</p> <p> Stops a running build</p>

Build Details








This page displays additional build properties, including the full command line, environment, and eMake version for a specific build.

Actions

-  **View Messages**
 - Opens the **Messages** page for viewing all system log entries for the build.
-  **Download Build Log**
 - Downloads the annotation file for a completed build. For details about the annotation file, see the “Annotation” chapter in the *ElectricAccelerator Electric Make User Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.
-  **Delete Build**
 - Deletes a non-running build.
-  **Stop Build**
 - Stops a running build.
-  **New Comment**
 - lets you add a new build comment as described in [Using Comments](#).

On the left side of the page, a pulldown menu lets you select **Details**, **Configuration**, or **Metrics** information for your build.

Build Summary

Field	Description
Status	<p>Build's result or status:</p> <p> Paused by a breakpoint</p> <p> Running</p> <p> Completed successfully</p> <p> Timed out. For details about possible causes, see KB article KBEA-00039 "Fixing builds that the Cluster Manager reports as timed out."</p> <p> Stopped</p> <p> Failed or encountered an error</p> <p> Running but waiting for a build license. This appears when the number of concurrent builds has reached the license limit. The build will resume when another build stops running for any reason (because it completed successfully, completed with an error, was stopped from the Cluster Manager, timed out, and so on).</p>
Name	Build name as defined by the build class tag definition
Status	Text-based version of the build result or status.
Start Time	Date and time the build was started
Duration	Build duration (the elapsed time between the start and end time)
Conflicts	Number of conflicts in the build

Details Pane

This pane displays detailed information about the build and the machine that invoked the build. If a breakpoint is active and was triggered, this information appears here also. For details about using ecbreakpoint, see the ["Using Breakpoints"](#) section.

- **Build ID**—Build ID number assigned by the Cluster Manager
- **Build Class**—Build class to which the build belongs. A default build class name is displayed if one was not assigned
- **Job Count**—Number of jobs run during the build
- **Priority**—Priority at which the build was run
- **Min Agents**—Minimum number of agents used during the build

- Max Agents—Maximum number of agents that can be used by the build
- Effective Agent Alloc—Percentage of time the build was able to get all the agents it requested that were available for it to use. 100% means the build had all requested agents 100% of the time, while a lower percentage means that there were periods where agents that could have been used by this build were allocated to other builds
- Wait Time—Time that the build spent waiting for agents to become available
- IP set—If eMake was directed to use an alternate set of IP addresses to access agent hosts, this field indicates which set was used. See the Multi-Interface Configuration document under ElectricAccelerator Technical Notes for more information. http://docs.electric-cloud.com/technicalnotes_doc/PDF/ea_multiinterface_technote1_0.pdf
- Workload—Sum of agent usage over all agents (in seconds). This value also appears in the annotation file
- Host Name—Name of the machine from which the build was started
- IP Address—IP address of the machine from which the build was started
- User Name—Effective name of the user that started the build. This will be the same as the OS User Name unless `--emake-impersonate-user` was specified
- OS User Name—Actual name of the user that started the build. This will be the same as the User Name unless `--emake-impersonate-user` was specified
- User Label—Label portion of the build name
- Platform—Platform on which the build ran
- History Exists—Whether or not a history file existed for the build.
- Command Line—Command line used to invoke eMake
- Resource Request—Resource on which the build ran.
- CWD—Directory from which eMake ran
- EMake Version—Electric Make version information
- Build Log Directory—Location of the build log
- History File—Full path to the history file that was input into the build

Configuration Pane

This pane includes several build properties and displays build environment variables. AnnoDetail lists the annotation options used for the build. These options might have been specified on the eMake command line, or they might be inherited from the build class

Metrics Pane

This pane includes more than 50 metrics characterizing different aspects of build performance. For definitions of these metrics, see the “Metric Values in Annotation Files” section in the “Annotation” chapter of the *ElectricAccelerator Electric Make User Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.

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 opens 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 lets you [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> <code>QA_BUILD_1234_20060123185958</code>
Resource	Resource that the build class uses.

Column	Description
Platform	Operating system that the build class uses/supports.
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	<p>Edit Build Class—Opens the corresponding Edit Build Class page. Your existing build class specifications populate the appropriate fields.</p> <p>Delete Build Class—Deletes the corresponding build class.</p>

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**—Opens 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 lets you 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 details about user-defined variables, see the “Electric Make Command-Line Options, Environment Variables, and Configuration File” section in the “Electric Make Basics” chapter of the <i>ElectricAccelerator Electric Make User Guide</i> at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html. For details about constructing tag definitions, see the “Tag Definitions” section in the “Additional Electric Make Settings</p>

Field	Description
	and Features” chapter of the <i>ElectricAccelerator Electric Make User Guide</i> at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html .
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 .
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.)






Field	Description
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 or a newer version, 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>For more information about JobCache, see the <i>ElectricAccelerator Electric Make User Guide</i> at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.</p>
Annotation Upload	<p>Checkbox that determines whether to upload the annotation file to the Cluster Manager.</p>
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 JobCache 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* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.

Agents







The **Agents** page lets you view all configured agents. If an agent is missing, check the hardware for proper connection and setup or whether the agent is assigned to the correct Cluster Manager host name.

Functionality

- Click an agent name in the **Name** column to open the [Agent Details](#) page, where more information is available for that specific agent
- Click a host in the **Host** column to filter the agent list to agents on that host only
- Click a build in the **Build** column to open that build's [Build Details](#) page
- Click an agent in the **Enabled** column to enable or disable that agent
- Click the  (test agent) button to test the agent in that row. For details, see [Testing Agents](#)
- After [filtering agents](#), you can then [enable](#), [disable](#), [test](#), or [delete](#) the agents in the filtered results by clicking the  Enable  Disable  Test, or  Stop button respectively.
- The **Actions** column lets you 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 the information current
 - Set the number of records you want to see per page

Column Descriptions


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

Column	Description
Enabled	Whether the agent is available for use. If the agent is enabled and alive, the Cluster Manager can use the agent
Status	<div>  Agent is alive (is running and connected to the Cluster Manager) </div> <div>  Agent is not alive (cannot connect to the Cluster Manager) </div> <div>  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 </div>
Actions	<div>  Opens the Messages page for viewing all system log entries for the agent </div> <div>  Deletes the agent </div> <div>  Tests the agent </div>

Agent Details

This page displays information for a specific agent.

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 cannot contact the agent, this section provides troubleshooting information.

Configuration

This section provides detailed information about the agent's configuration, including port numbers, IP address, agent host name, 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**

lets you 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 results 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 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 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

Preemption Policy Setting	Reassign the agent to A from B?	
	Unlocked or Lock Interval = 0	Lock Interval > 0 and time elapsed exceeds interval
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 opens the resource's [Resource Details](#) page.
- You can [create](#) a new resource or [delete](#) existing resources.
- The **Actions** column lets you [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	Name of the resource. Click a resource name to go to the Resource Details page for that resource.
Hosts	Semicolon-delimited list of host name glob-style patterns.
Actions	Edit Resource —Opens 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**—Opens the [Edit Resource](#) page.

Details

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

Comments

New Comment lets you 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 the resource name and a list of host names or masks.

Field	Description
Name	<p>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 lets you 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:

Report	Description
Agent Usage	Stacked area chart that shows the agent license count and agents available versus agent demand in a specified time period
Build Duration	Scatter plot that shows the durations of builds over a specified time period
Build Duration vs. Conflict	Scatter plot of build duration versus the number of conflicts per build
Build Duration vs. Job Count	Scatter plot of build duration versus the number of jobs per build
Build Summary	Table that shows aggregate statistics (average wait time and total workload) that are grouped by either build class or build name
Build Wait Time	Scatter plot that shows the wait times for builds over a specified time periodCluster
Utilization	Two stacked, stepped line charts that represent average agent demand and average builds running over a specified time period
Concurrent Builds	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	Pie chart that shows the effective agent allocation percentage that is reported by eMake for a specified time period
Effective Agent Allocation Per Day	Stacked bar chart that shows the effective agent allocation percentage that is reported by eMake per day
JobCache License Allocation	Skyline chart that shows concurrent builds at any time (stacked by JobCache license request results) to show how many were able to use JobCache

Report	Description
JobCache Usage Summary	Several pie charts that show job cache requests and their results
Resource Usage over Time	Chart that shows the number of simultaneous builds that ran on each resource over a specified time period
Sea Level	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

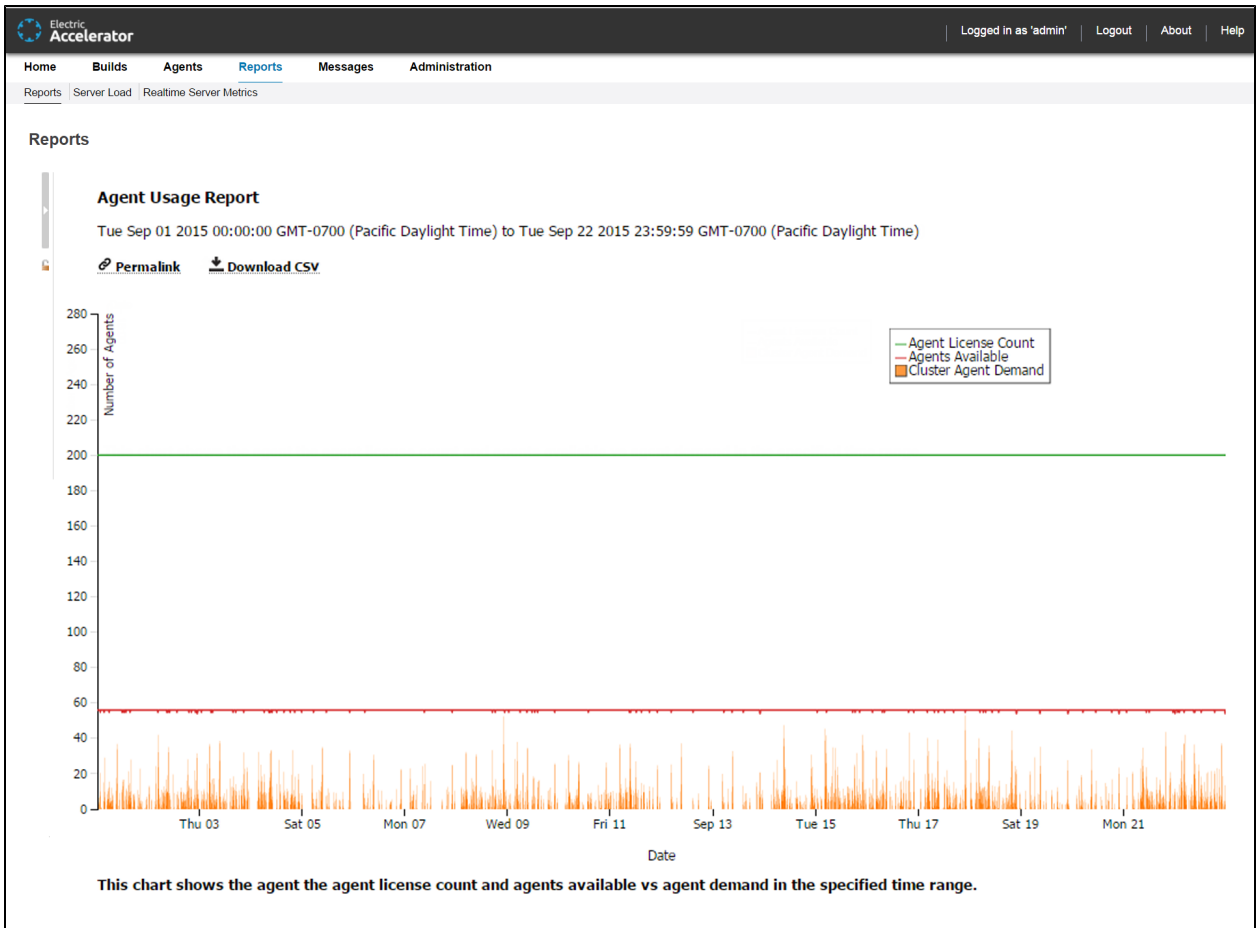
Most reports include a **Permalink** button, which provides permanent, direct access to a report. Click the **Permalink** button to browse to the version of the report with a permanent URL, which you can bookmark in your browser or send to others for convenient viewing.

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:

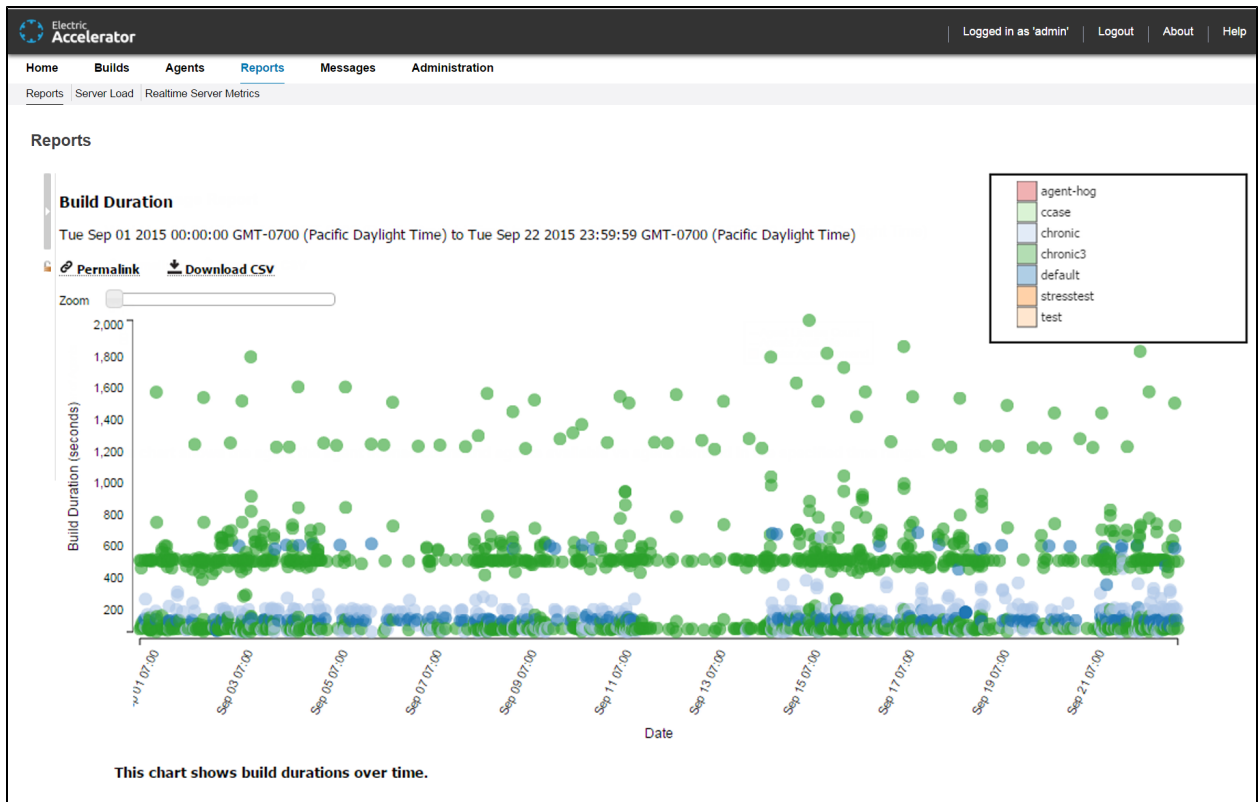


You can input one or more resource names. If no resource name is specified, the cumulative usage for the entire Cluster is shown.

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

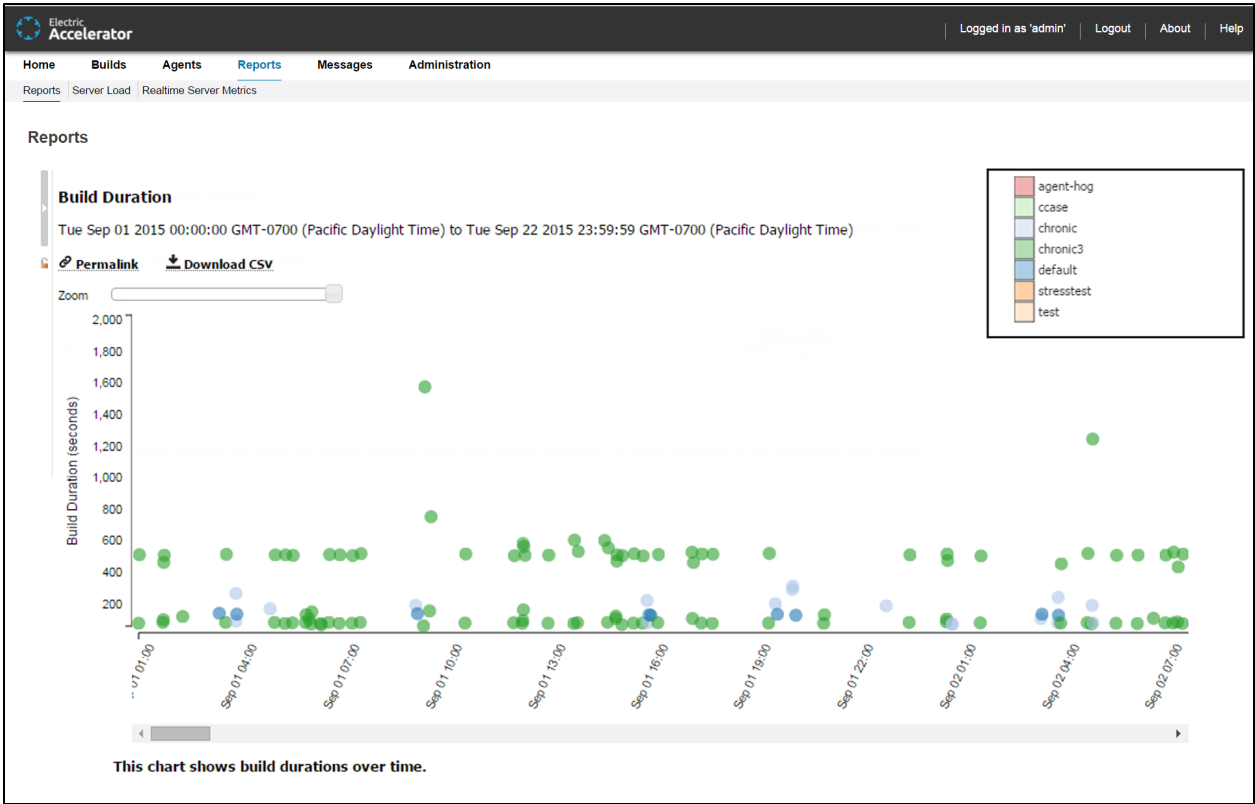
This report is a scatter plot that shows the durations of builds over a specified time period:



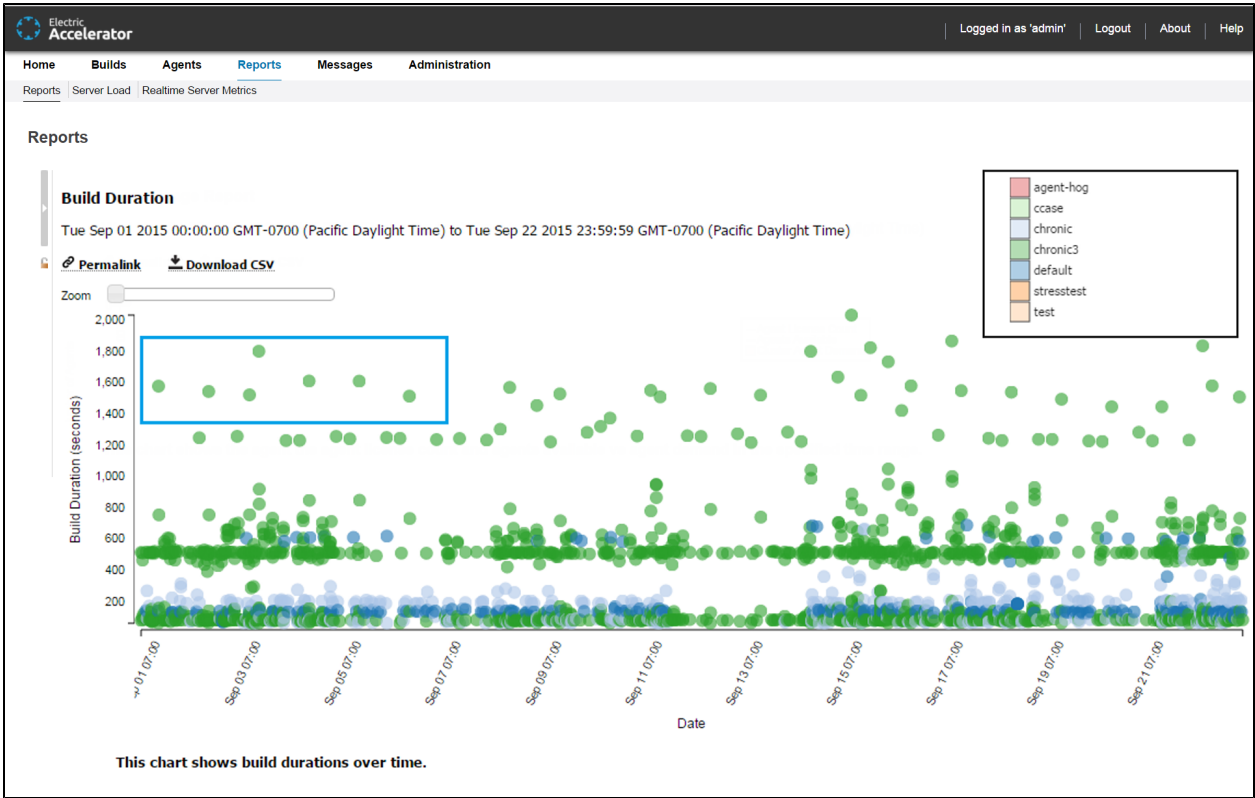
You can input a build class or a build name along with a date range. You can also specify whether you want builds (points) to be colored by build class or build name.

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:

Electric Accelerator

Home Builds Agents Reports Messages Administration

Reports Server Load Realtime Server Metrics

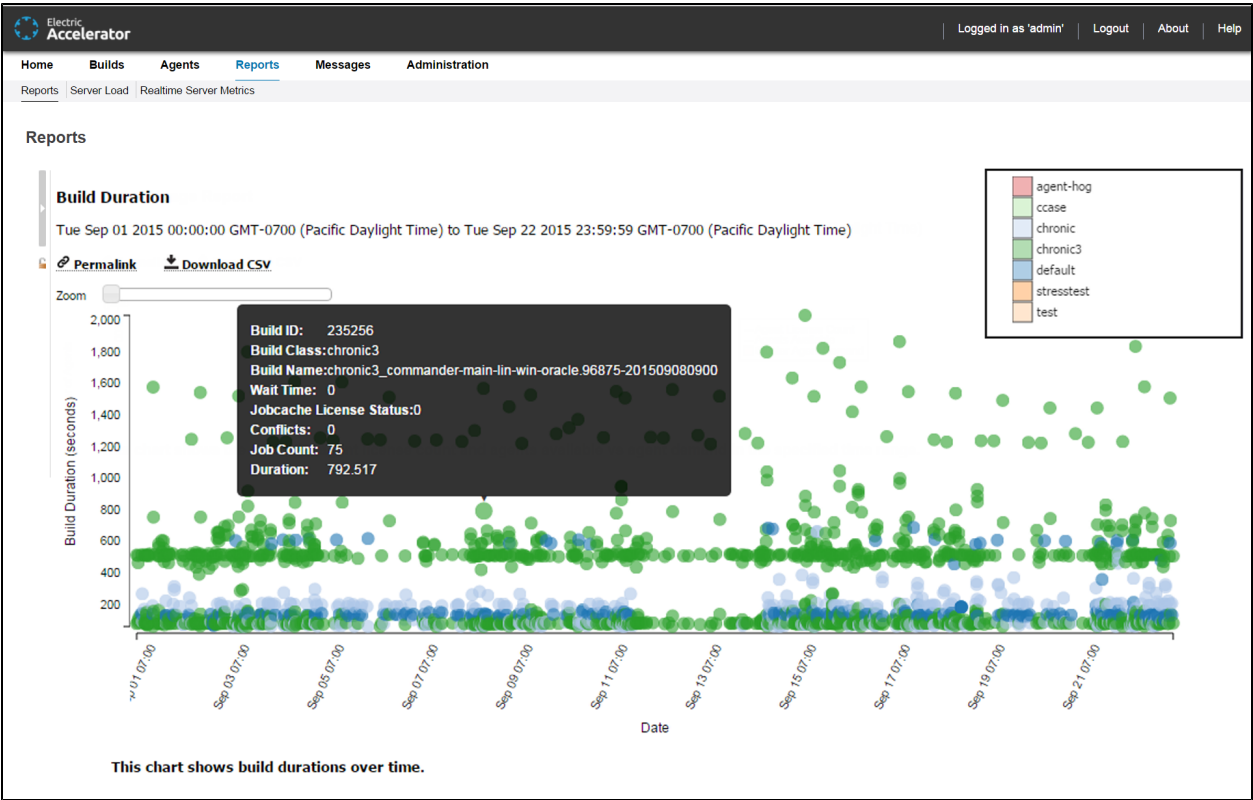
Reports

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:

Electric Accelerator

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

HomeBuildsAgentsReportsMessagesAdministration

BuildsBuild Classes

Build Details

View MessagesDownload Build LogDelete Build

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

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-annofile=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

New Comment

Comment

Modified By

Last Modified

Actions

No records

Go

Build Duration vs. Conflict Report

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

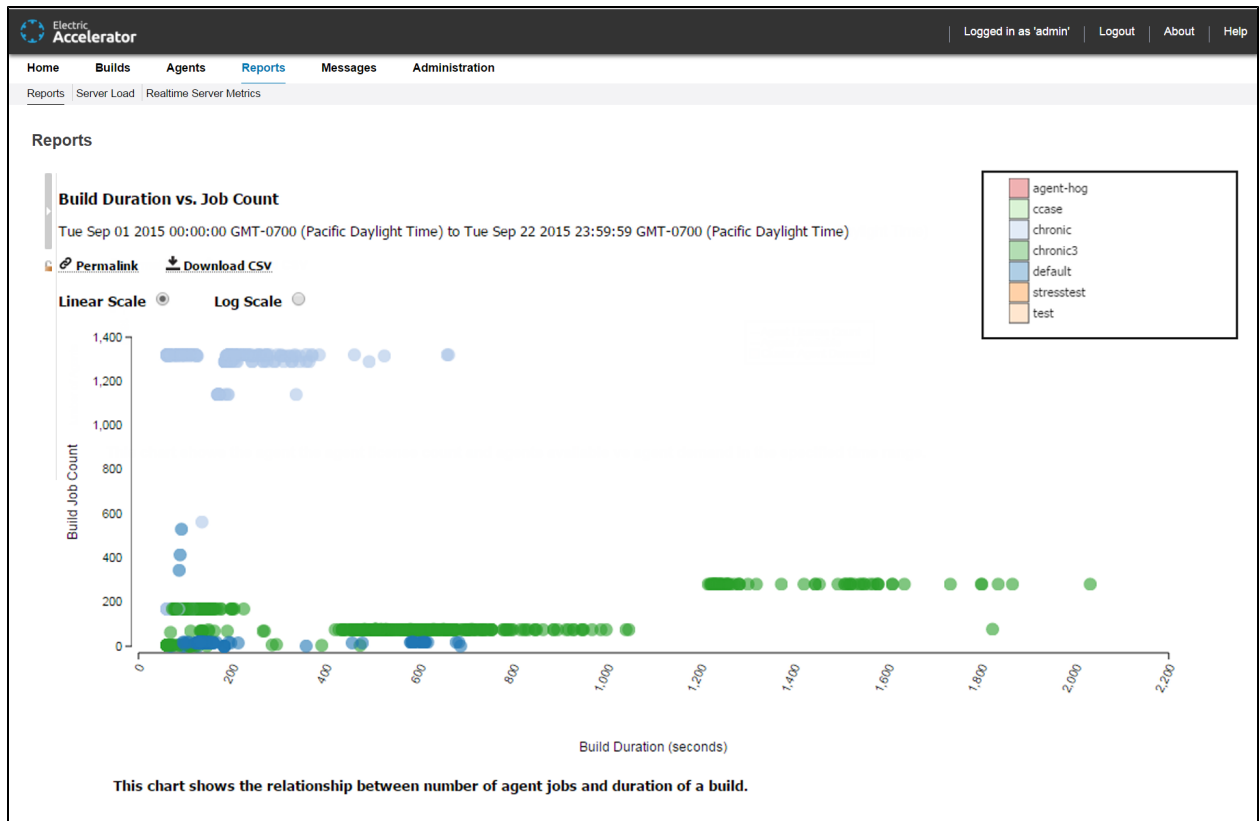


You can input a build class or one or more build names along with a date range. Also, you can specify a threshold for conflicts in a build. Builds with conflicts lower than the threshold are filtered out. You can also select whether to color the report by build class or build name.

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:

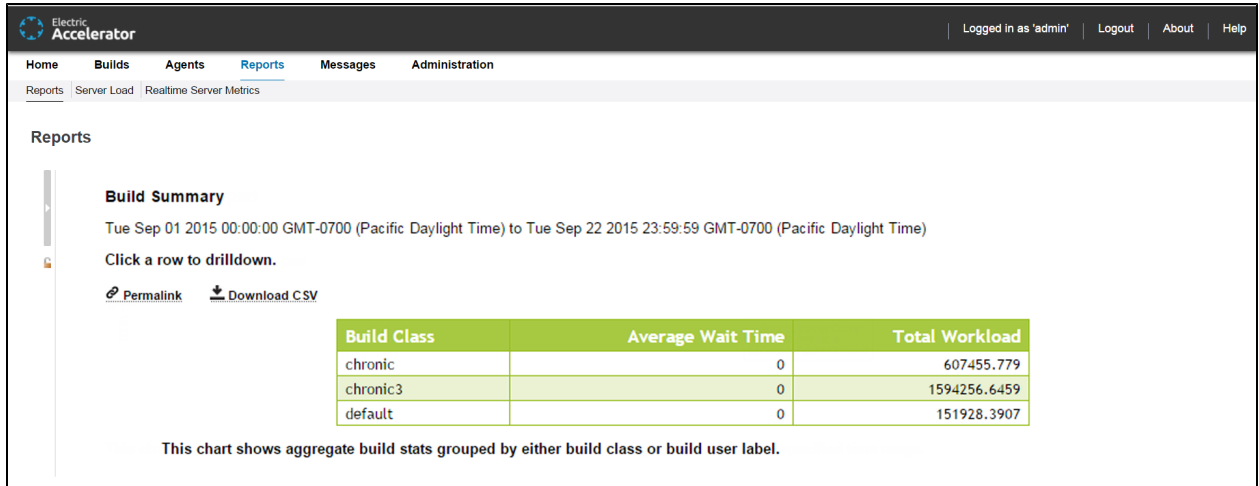


You can input a build class or one or more build names along with a date range. You can also select whether to color the report by build class or build name.

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.

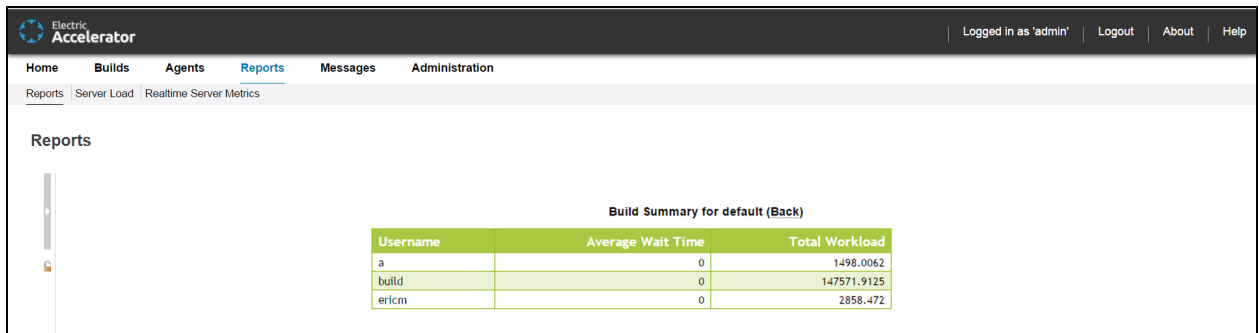
Build Summary Report

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:



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.

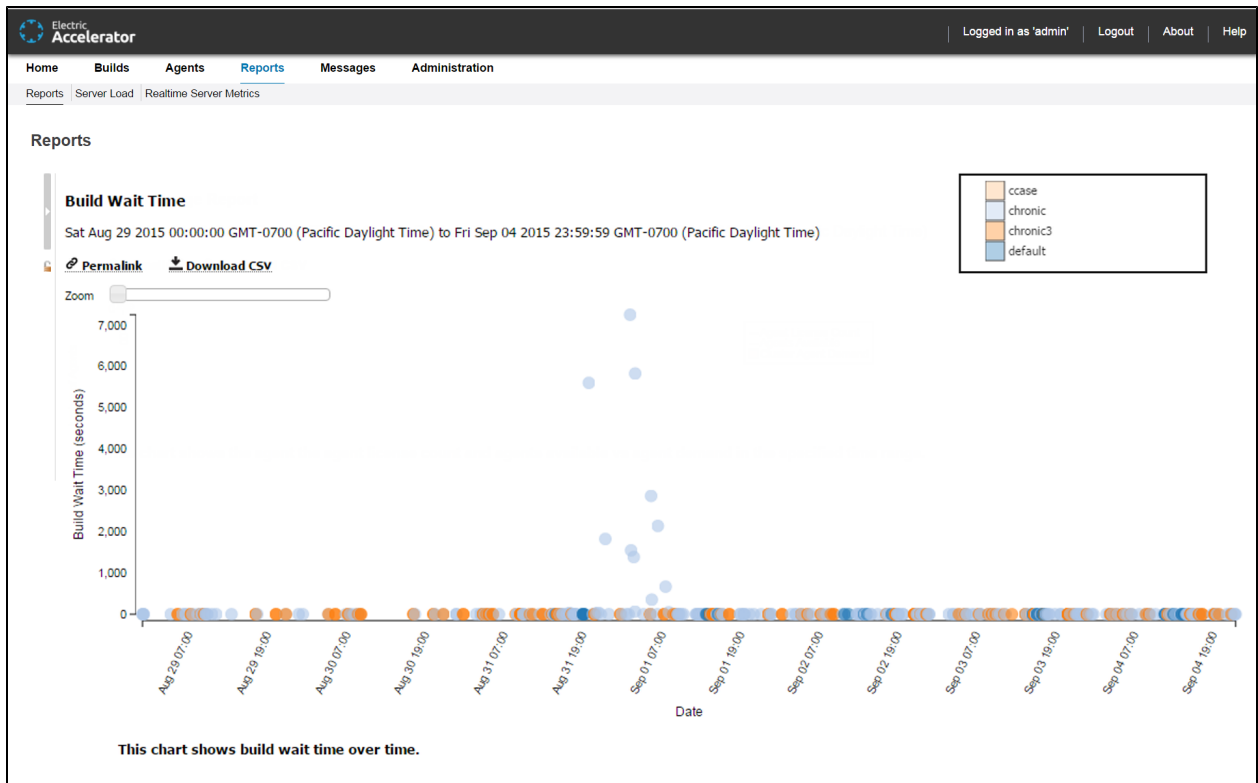
You can click a build class to see builds for that class grouped by user:



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:

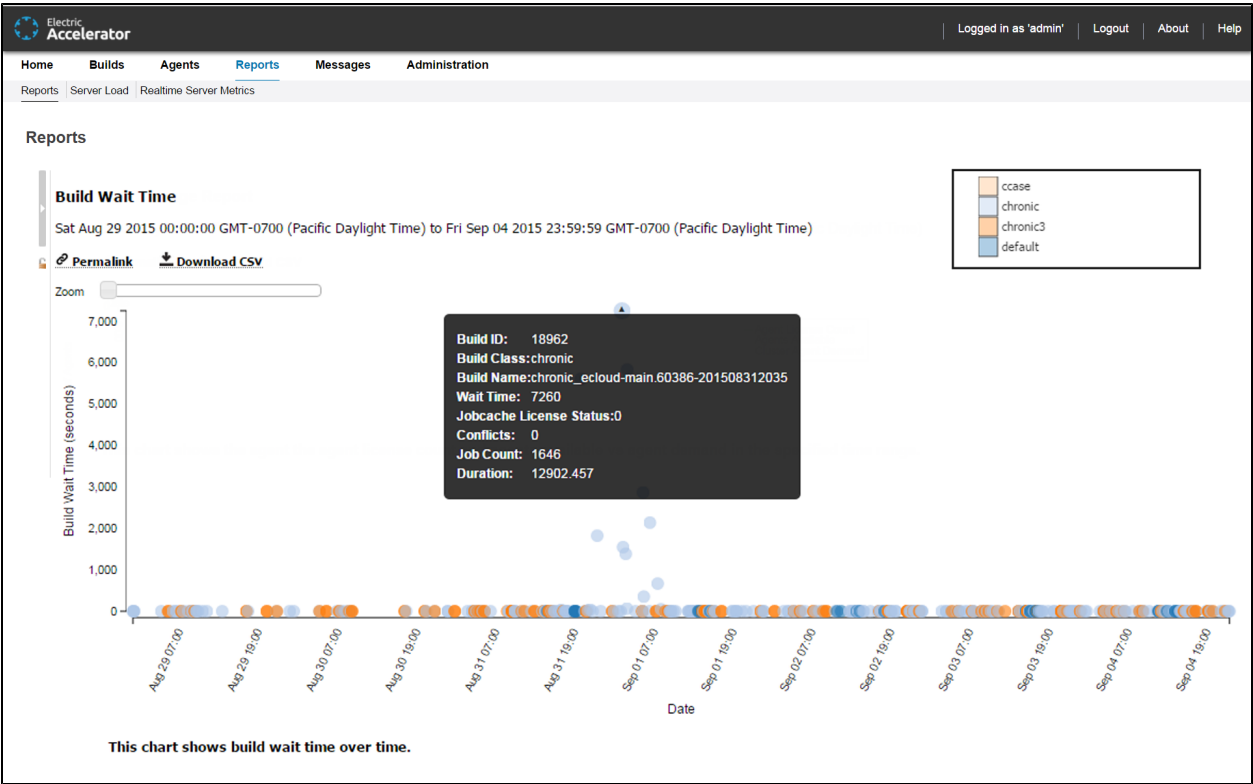


Wait time is the time when eMake has no agents available to run the build. A high number of builds with a non-zero wait time indicates that the Cluster might be too small.

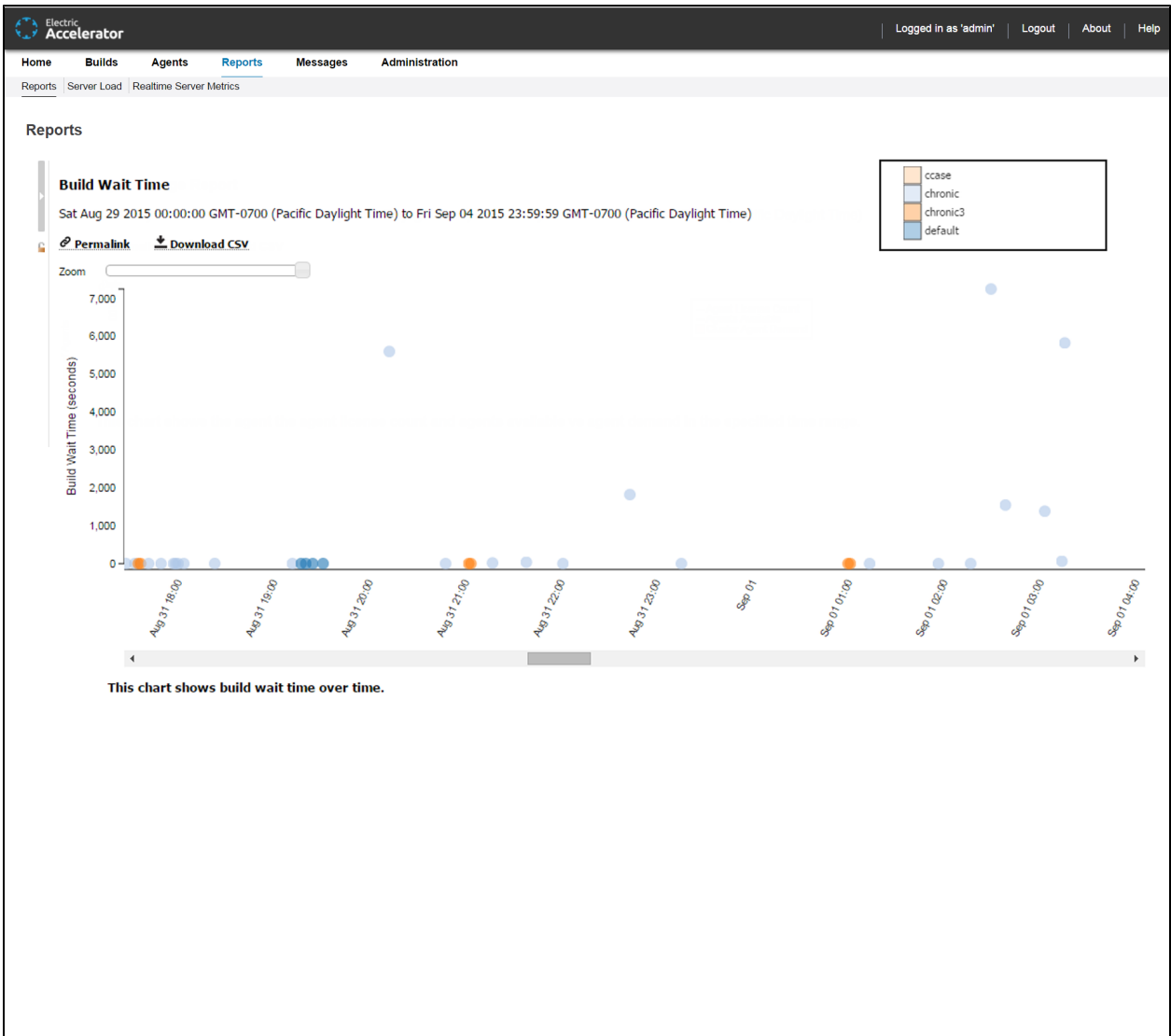
You can specify a build class or one or more build names along with a date range as input.

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:

Electric Accelerator

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

HomeBuildsAgentsReportsMessagesAdministration

BuildsBuild Classes

Build Details

View MessagesDownload Build LogDelete Build

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

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-annofile=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/2/3/5/2/5/2/35256

History File:emake-linux.data

Comments

New Comment

Comment

Modified By

Last Modified

Actions

No records

Go

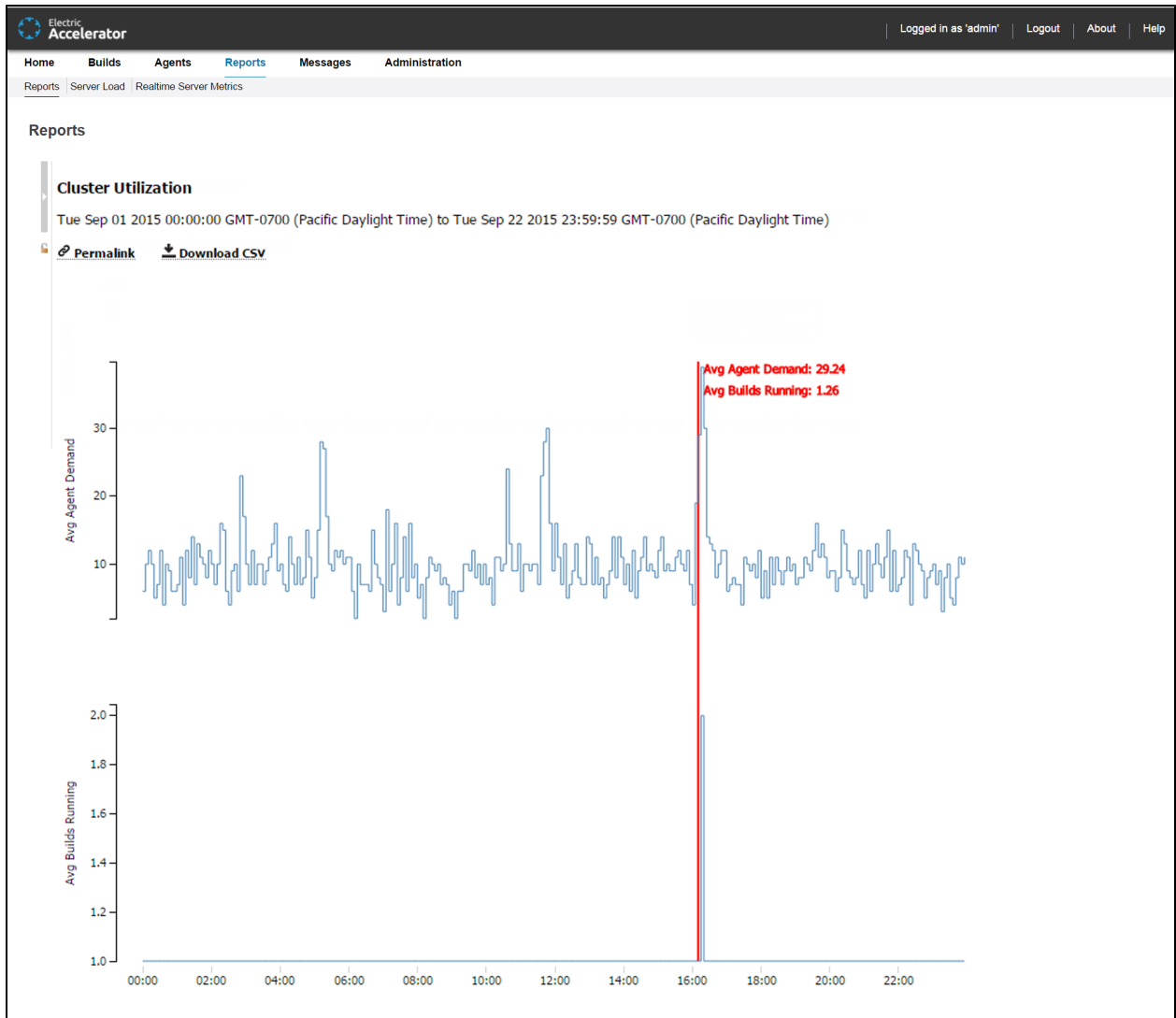
Cluster Utilization Report

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.

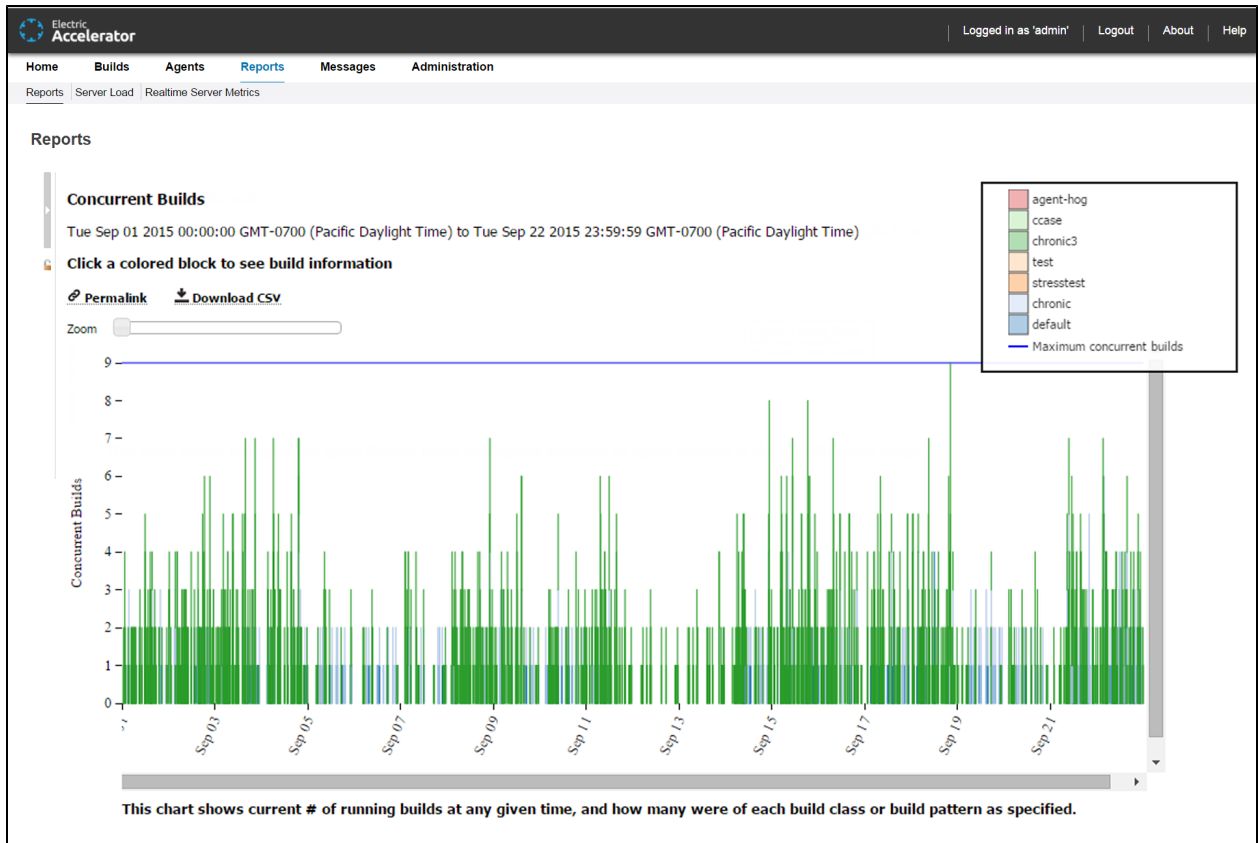
You can hover over the chart to see the average agent demand and average builds running at any point of time:



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.

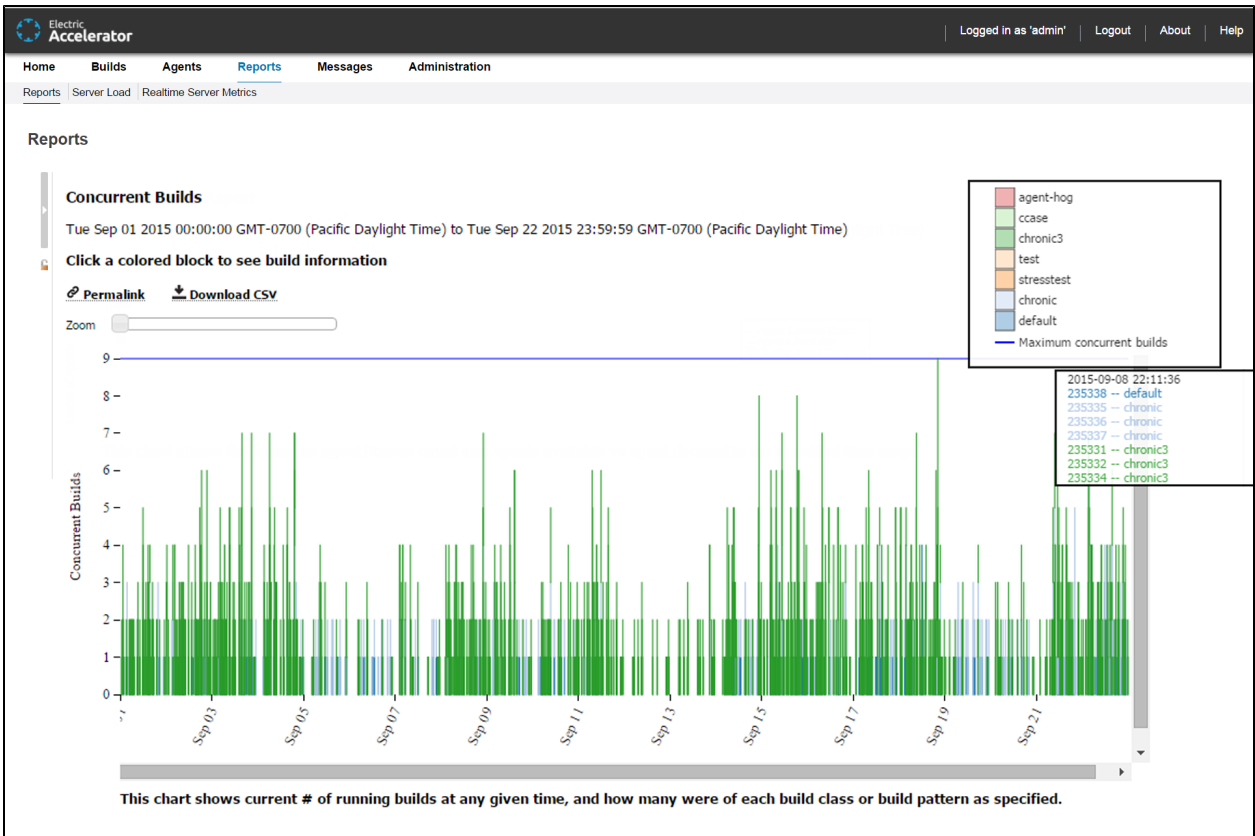
Concurrent Builds Report

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:

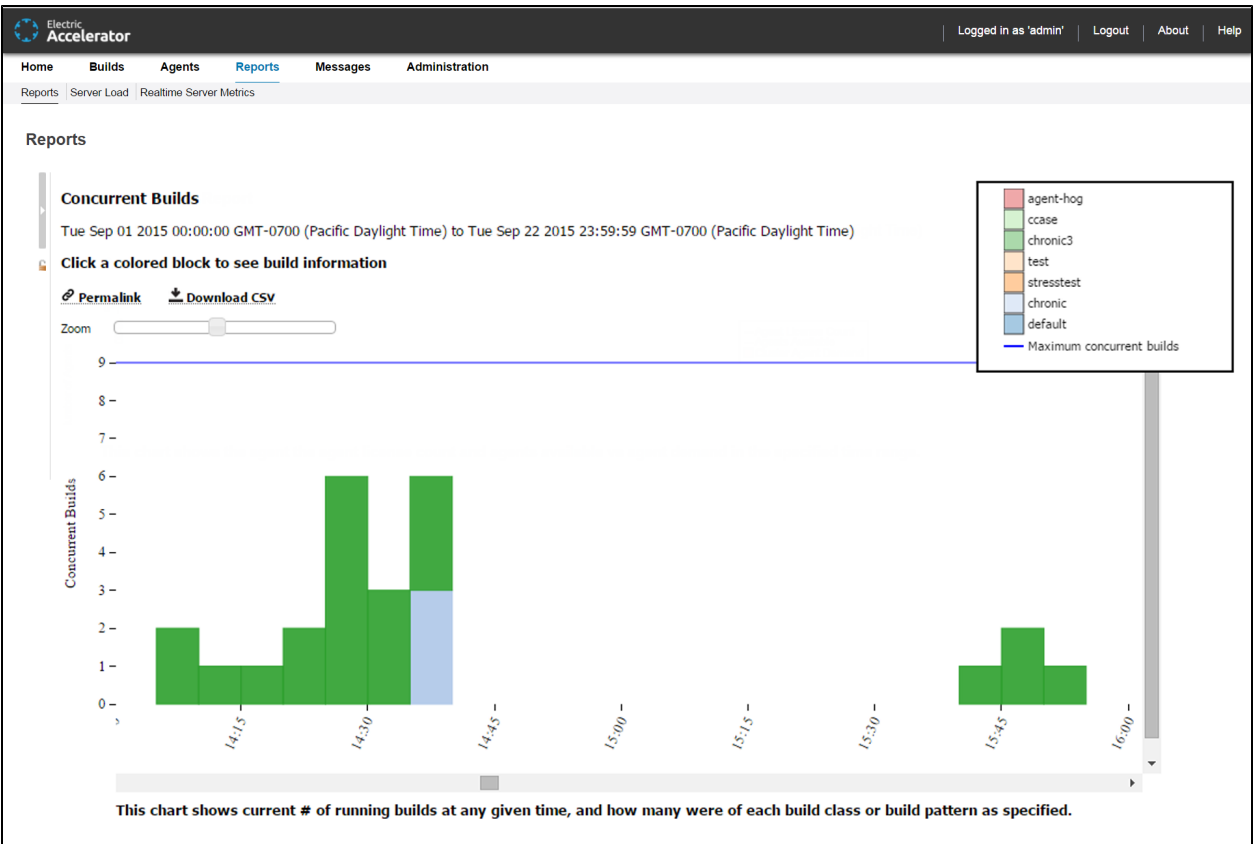


You can input a date range, a sample interval size (the default is five minutes), and whether to group by build class (the default) or by build name. You can omit builds under a specified number of seconds. You can specify a series of build name patterns.

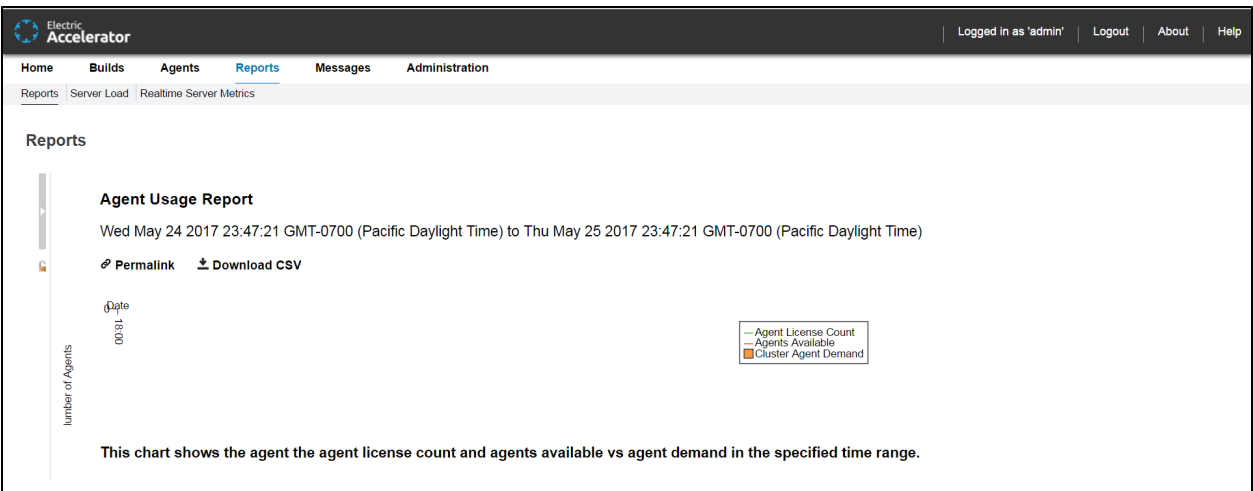
You can hover over a stack to see a list of concurrent builds for that stack:



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

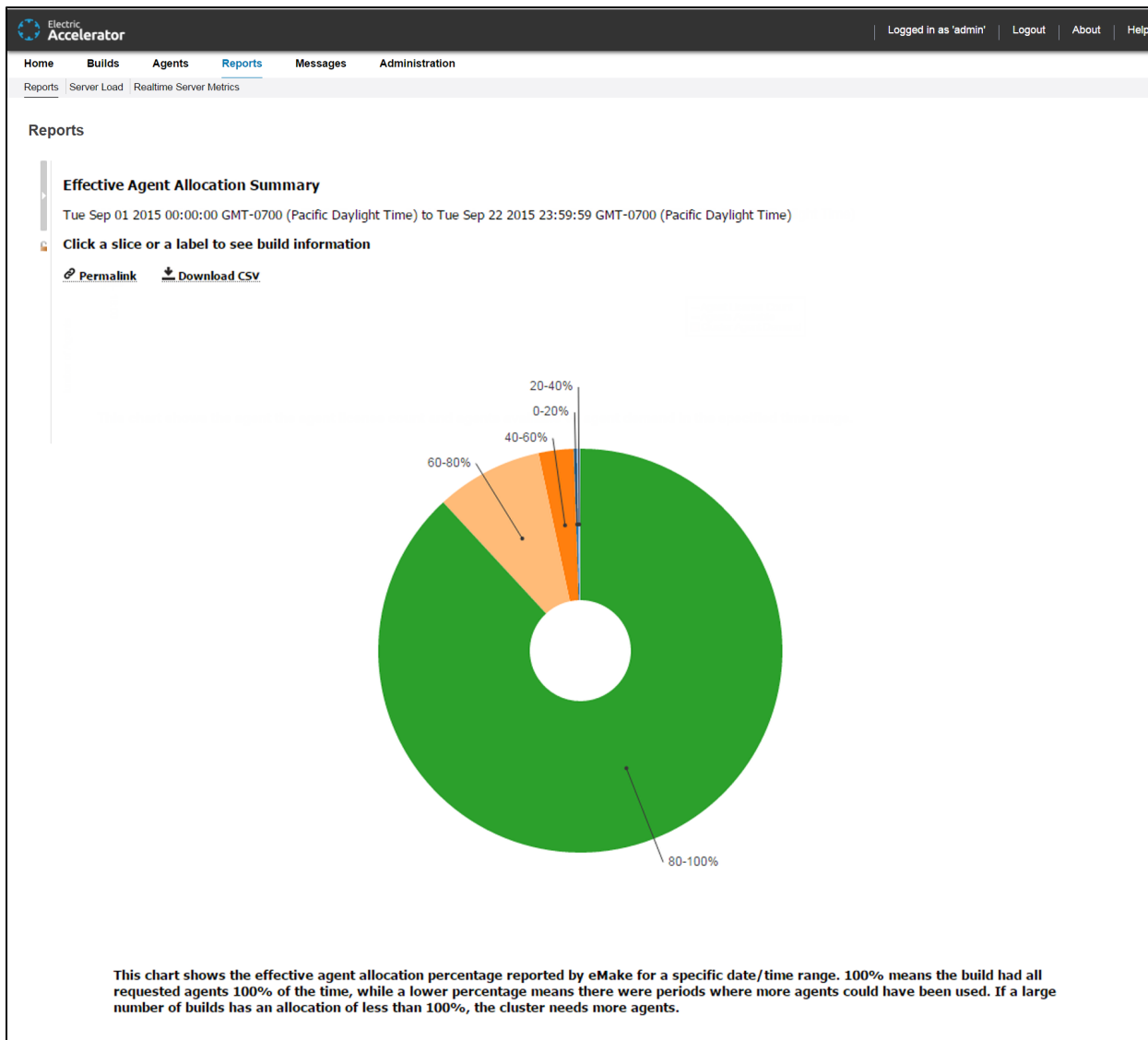


You can click to see a table of builds for each build event (when a build starts or when a build ends):



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:

Electric Accelerator

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

HomeBuildsAgentsReportsMessagesAdministration

ReportsServer LoadRealtime Server Metrics

Reports

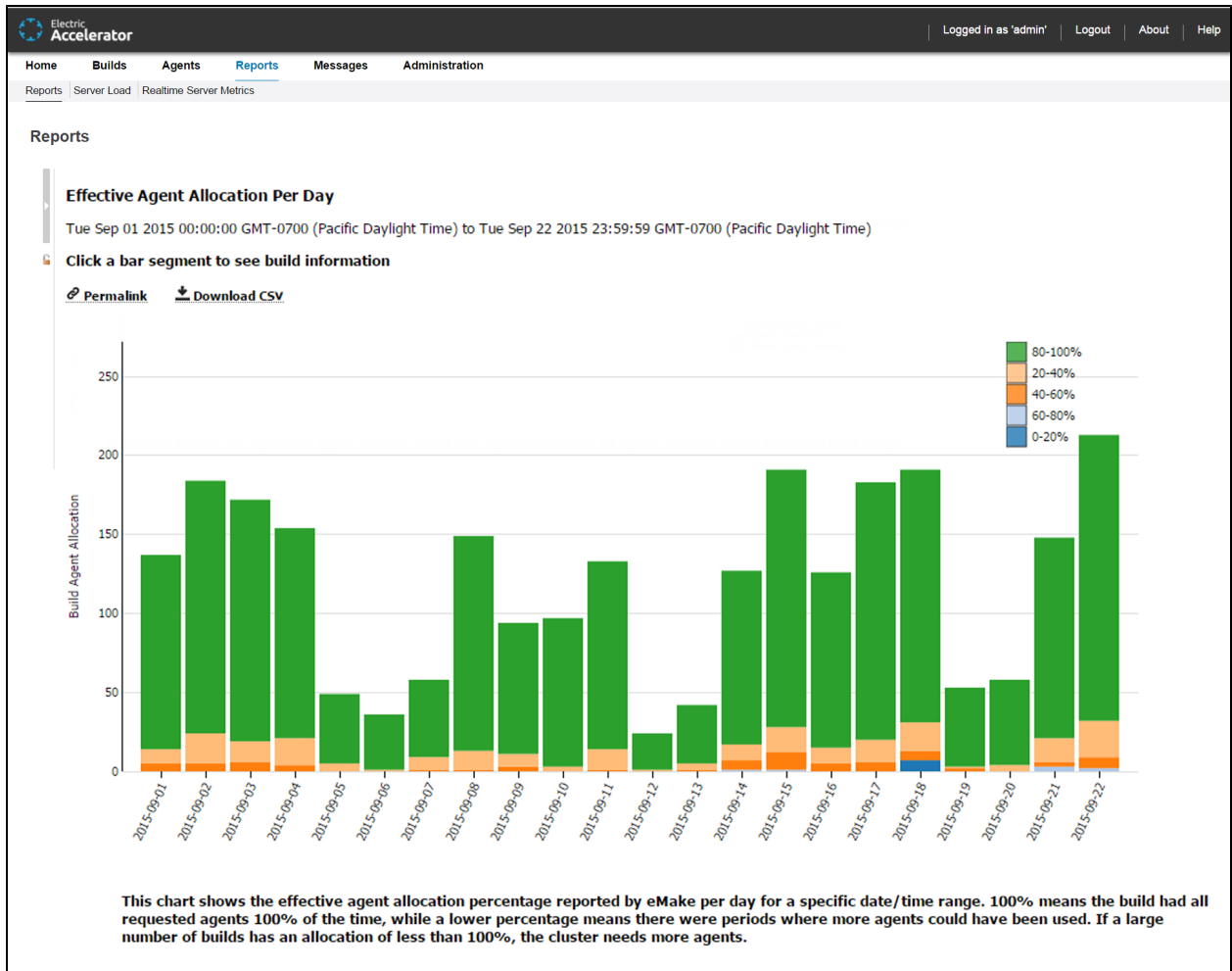
Builds with 60-80% agent allocation (Back to chart)

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

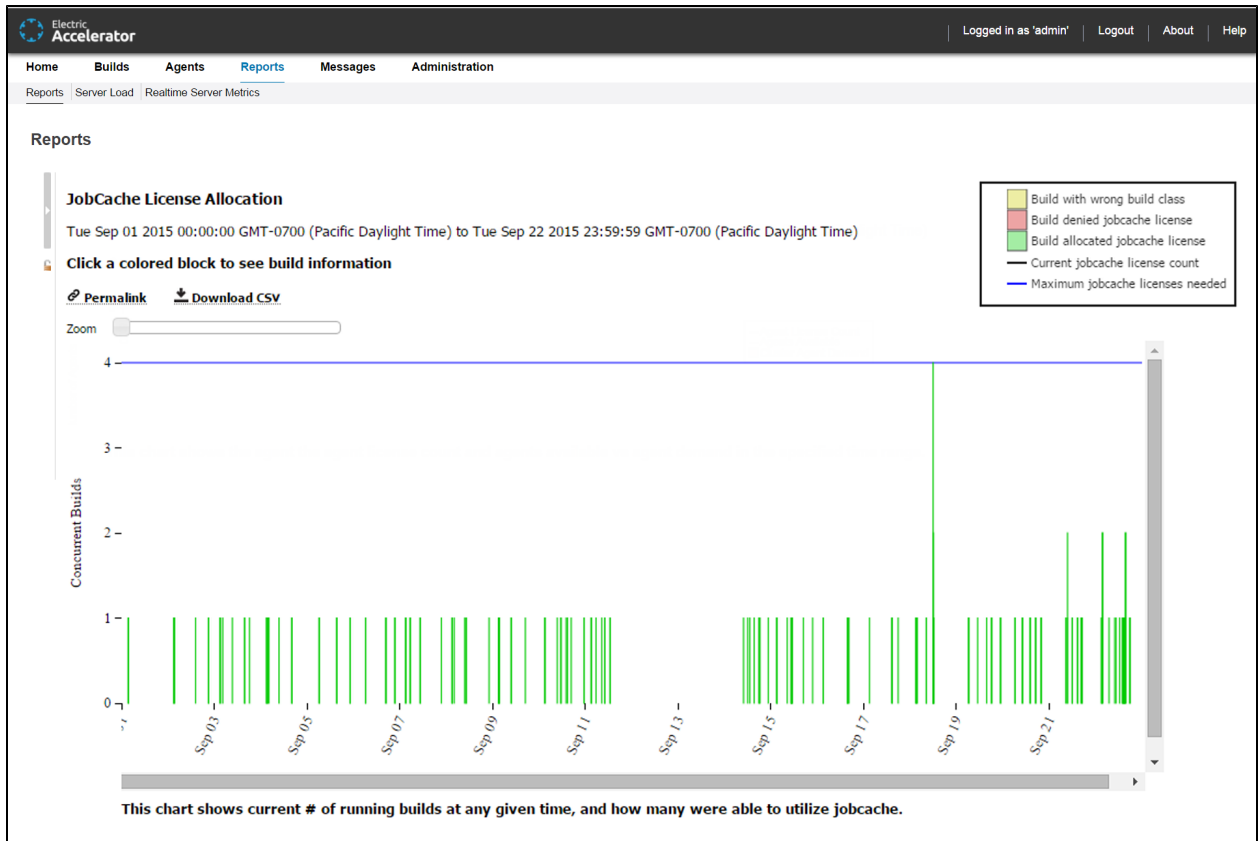
Effective Agent Allocation Per Day Report

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

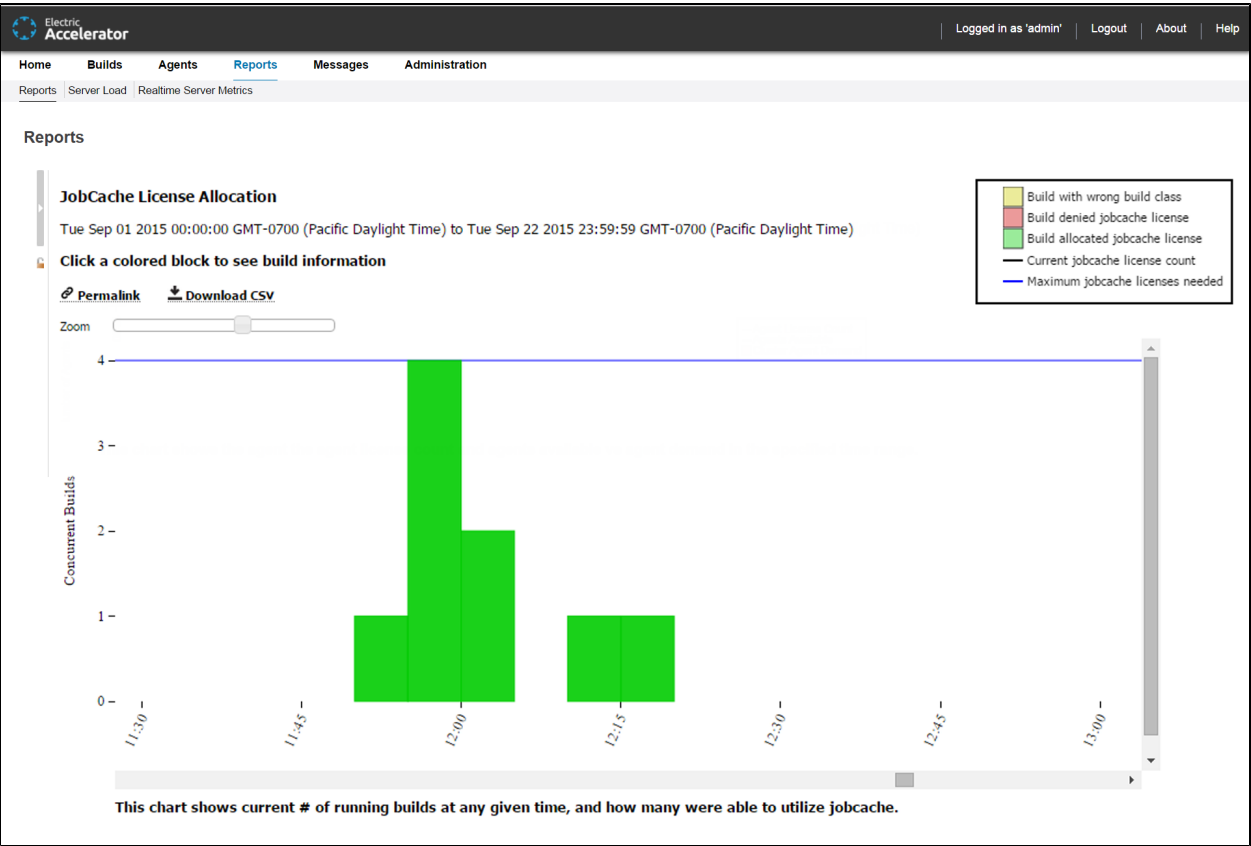
This report is a skyline chart that shows concurrent builds at any time, stacked by JobCache license request results, to show how many were able to use JobCache:



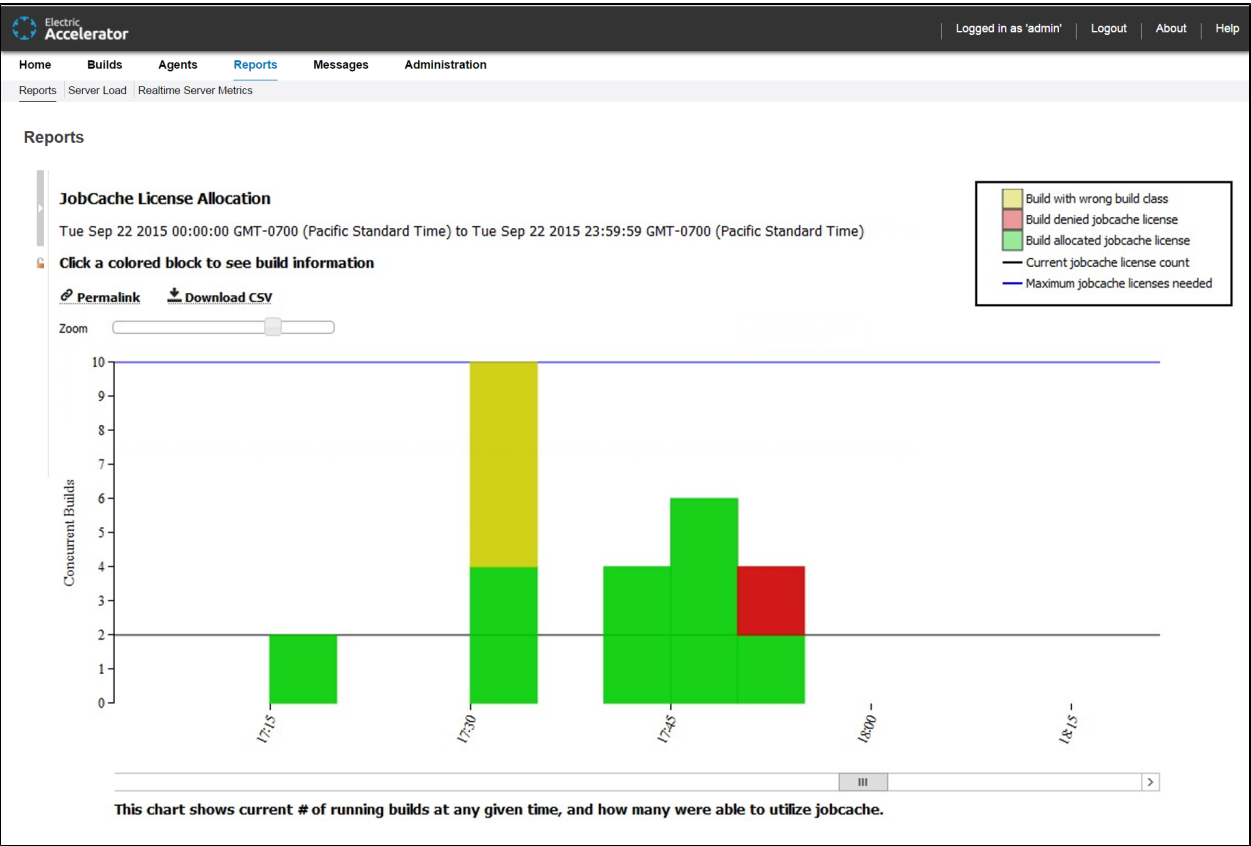
The chart includes two color-coded lines indicating the current JobCache license count and the maximum JobCache licenses needed.

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.

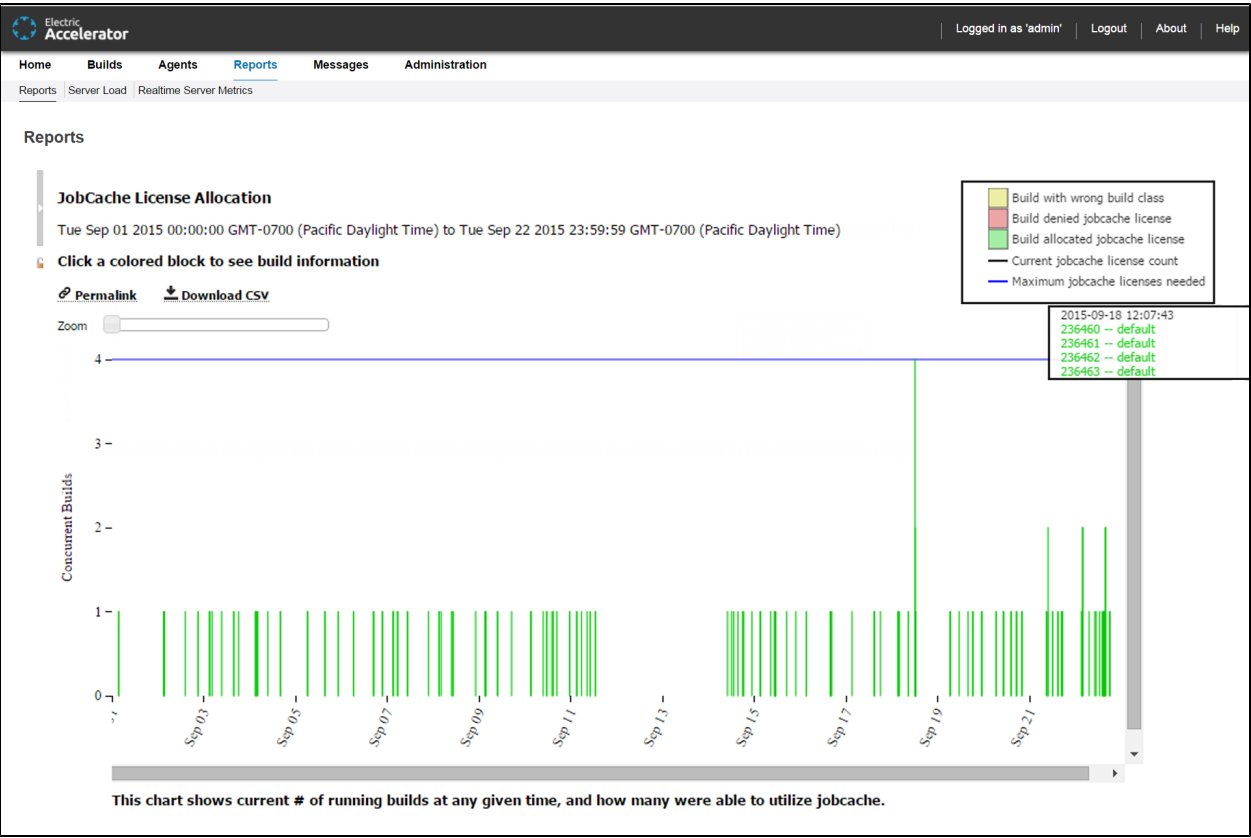
You can use the **Zoom** slider to zoom in to a particular stack or set of stacks:



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:

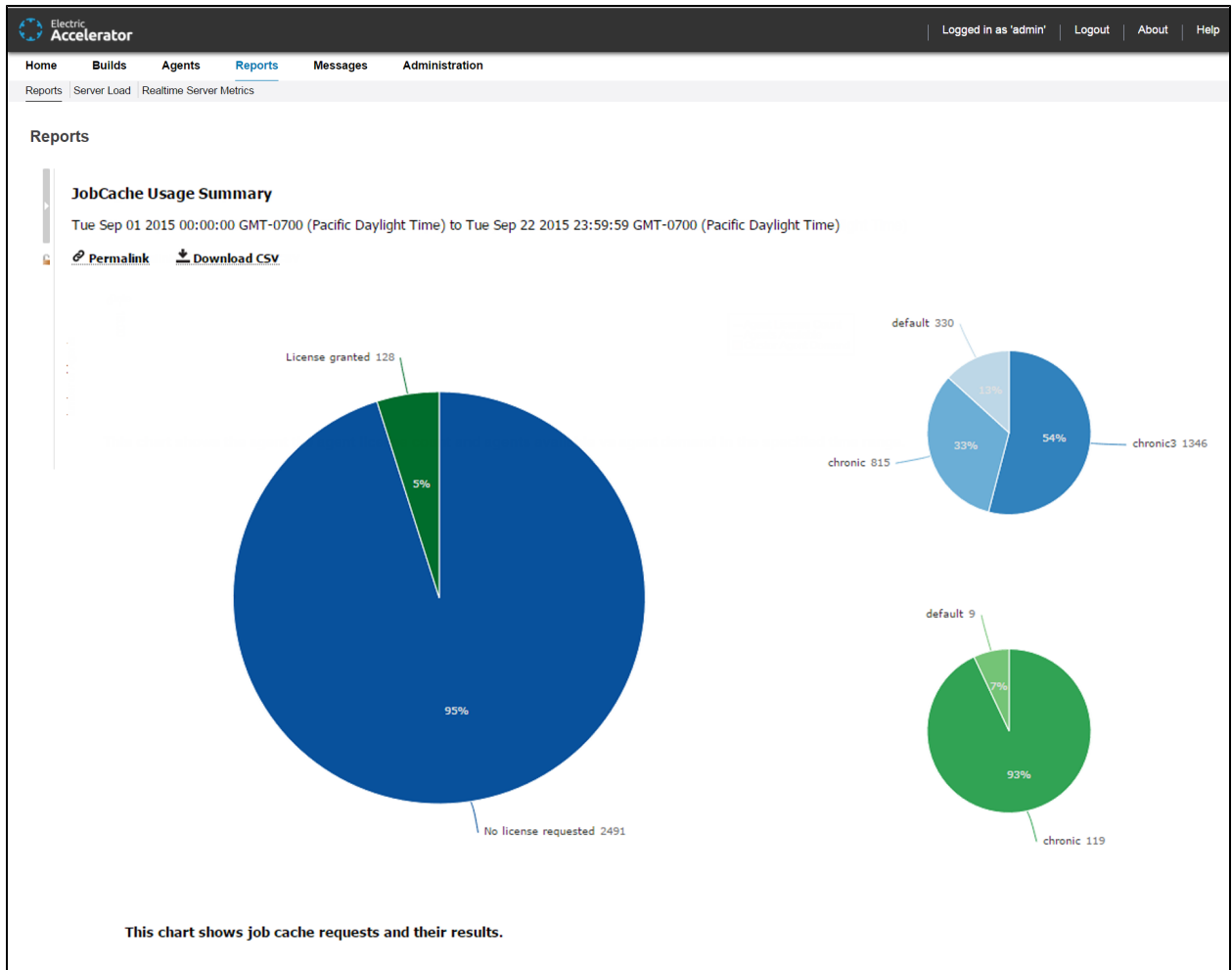


You can click a stack to see a table of builds for that build event:

Builds 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 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

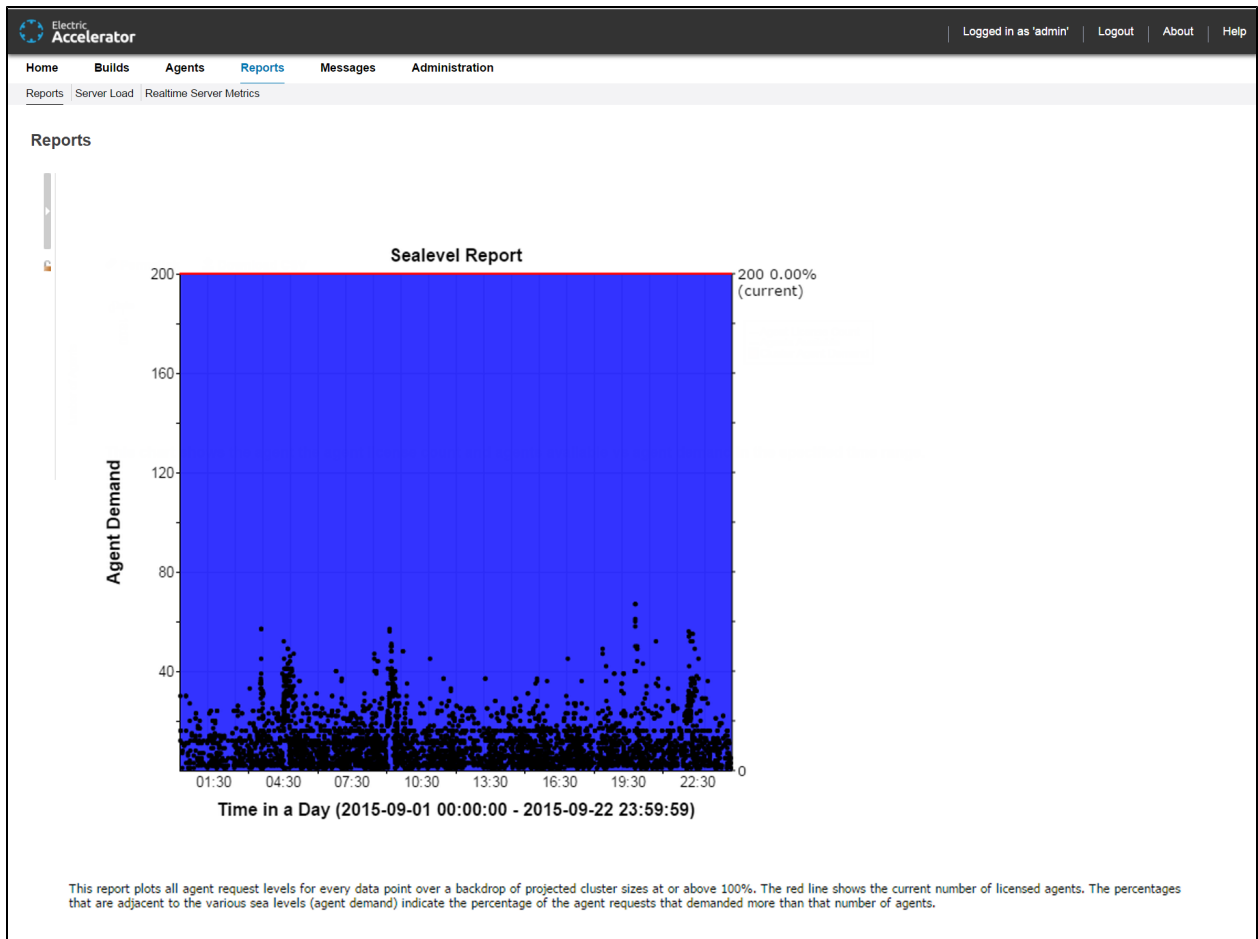
This report is a chart that shows the number of simultaneous builds that ran on each resource over a specified time period:



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	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	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	Average number of agents that are used for all builds

Column	Description
Agent Cluster Shortage	Number of 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 and is unavailable for individual resource statistics
Agent License Shortage	Number of agents that 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	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 five builds are 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	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	Description
Severity	Levels are:  Debug  Info  Warning  Error
ID	Global message ID.
Text	Text of the message.
Time	Time of the message.
Build	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	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 Delete Message in this column to delete the corresponding message.

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

A filter lets you drill down to a relevant set of information. The Cluster Manager provides a set of predefined filters for commonly requested information such as broken builds, lists of builds for a particular build class, lists of broken agents, and error messages.

You can choose a predefined filter or a custom filter from the **Saved Filters** drop-down menu that appears in lists that support filtering. Users can create their own custom filters.

Saved Filters Panel

The **Saved Filters** panel appears on the left side of a page. The panel lists all ElectricAccelerator-supplied filters and filters that you create for your own use only.

The **Saved Filters** panel has the following functions:

- **My Saved Filters**—Lists the custom filters that were created by the currently logged in account
- **Legacy Filters**—Lists the predefined filters

A rectangular button with the word "Save" in a sans-serif font.

- button—Saves your modifications to the selected filter

A rectangular button with the word "Delete" in a sans-serif font.

- button—Deletes the selected filter. A popup asks you to confirm the deletion



- (permalink) button—Provides permanent, direct access to a filter. Click the button to browse to the version of the filter that has a permanent URL. You can bookmark a permalink in your browser or send it to others for convenient viewing.

Functionality



- Clicking a (column filtering) button at the top of a column opens a dialog box where you can adjust the filtering by choosing values from the values available in that column. For example, you can adjust the filtering for the **Status** column so that the table includes only the successful builds.
- The **Actions** column lets you [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 that you want to see per page

Lists that Do Not Support Filters

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

- **Users and Groups**—No filtering because they follow a different model (they are a combination of local and LDAP/Active Directory based users)
- **Licenses**—No filtering because there are few licenses in the system
- **Current LSF jobs**—No filtering because the set of jobs is usually small

Filters - Create or Edit a Filter

See the [“Creating Filters”](#) section for information about creating a filter and the [“Editing Filters”](#) section for information about editing a filter.

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 a group. The table information is view-only. You use this page to set or modify permissions on other pages.

Functionality

- You can enable permissions for multiple users or groups at once.
- The **Actions** column lets you edit or clear permissions for the user or group in that row.
- You can 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	Name of the user or group.

Column	Description
Access	Yes or No for permission to access the server.

Column	Description
<p>Impersonate</p>	<p>Yes or No for permission to impersonate a different user or 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 Cluster Manager looks for a user defined in Administration > Users for a matching user. If unsuccessful, the Cluster Manager uses the "default" user, whose Full Name is Anonymous.</p> <p>Next, the Cluster Manager 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>
<p>Invoke</p>	<p>Yes or No for permission to invoke eMake.</p>

The next nine columns display None, Read, Modify, or Full permission for the user or 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
Buils	Permission to read, write, or delete build entries
Agents	Permission to read, write, or delete agent entries
Build Classes	Permission to read, write, or delete build class entries
Messages	Permission to read, write, or delete message log entries
Administration	Permission to read, write, 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 or preemption policy)
User	None or Modify for permission to allow a user to change their own User Settings, Message Policies, filters, and Build Class notification
Break Points	<p>Full or Modify Permission is available for users who must use breakpoint functionality</p> <p>Read lets users view breakpoint information in the ElectricAccelerator web interface.</p> <p>Modify lets users interact with the breakpoint. Users can click Agent Command and Shell Command, which update the breakpoint by adding records to the database.</p> <p>Full lets users perform all breakpoint operations. Users can click Retry and Continue, which deletes the current breakpoint. If a breakpoint launches, and the user stops the build by killing the process, breakpoint information remains on the Build Details page, but the build is finished. Then there is a link to delete the breakpoint manually.</p> <p>See the Using Breakpoints help topic.</p>
Actions	<p>Edit User Permissions—Opens an edit page to change permissions for either a user or a group</p> <p>Delete User Permissions—Deletes 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	Specifies permission 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	Specifies permission to invoke Electric Make.
Builds	Specifies permission to read, write, and delete build entries.
Agents	Specifies permission to read, write, and delete agent entries.
Build Classes	Specifies permission to read, write, and delete build class entries.
Messages	Specifies permission to read, write, and delete message log entries.
Administration	Specifies permission to read, write, and delete entries in the Administration section (users, groups, and so on).
Reports	Specifies permission to access reports.

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

Permission	Description
Resources	Specifies permissions to perform resource management (for example, build priorities or preemption policy)
User	Specifies permissions to change their own settings and filters
Break Points	<p>Specifies permissions to access, read, write, or delete breakpoint feature/functions</p> <p>Read lets users view breakpoint information in the web interface.</p> <p>Modify lets users interact with the breakpoint. Users can click Agent Command and Shell Command, which update the breakpoint by putting records in the database.</p> <p>Full lets users perform all breakpoint operations. Users can click Retry and Continue, which delete the current breakpoint. If a breakpoint launches, and the user stops the build by killing the process, breakpoint information remains on the Build Details page, but the build is finished. Then there is a link to delete the breakpoint manually.</p> <p>See the Using Breakpoints help topic for information on using breakpoints.</p>

Click **OK** to save changes.

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 need access to ElectricAccelerator, click the **Permissions** subtab, then the **Enable User** link.

Functionality

- Clicking a user name opens the user's **User Details** page.
- You can create a new user or delete existing users.
- The **Actions** column lets you 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	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 opens the User Details or User Settings page, depending on whether you click your own user name or another user's name
Real Name	Real name of the user. For example, the real name for the “msmith” user name might be Mary Smith
Email	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 —Opens 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	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	Real name of the user. For example, the real name for the "msmith" user name might be Mary Smith
Email	Normal email address used to send email to this user
Password	Password for this user account

Editing a User

1. On the **Users** page or **User Details** page, click **Edit User**.

The **Edit User** page appears.

2. Select and retype the information you need to change or add new information.

For a description of all fields, see the "Field Descriptions" table above.

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. The default ElectricAccelerator user group is named Administrators. 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 opens the group's **Group Details** page.
- You can create a new group or delete existing groups.
- The **Actions** column lets you 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	Name of the group. Clicking a group name opens the Group Details page where you can see the list of members included in that group.
Source	Whether the source is Local for groups created in ElectricAccelerator or LDAP/Active Directory for external groups
Actions	Edit Group —Opens the corresponding Edit Group page. Add New User To Group —Opens 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 retype 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**.

Field Descriptions

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

Server Settings

Use this page to set server preferences for resource management, message logging, notifications, and transmission of anonymous usage statistics to Electric Cloud.

Field and Menu Descriptions

Field	Description
Default Priority	<p>Class build default priority and default boost values. A build class has a default priority and a default boost value of Normal and 0 respectively by default.</p> <p>The range for possible boost values is an integer from -10 to +10. A higher value means that builds in that class have preference over builds with the same priority and a lower value (for using available agents).</p> <p>If you change the default priority or boost values, a new build class will have those values. However, you can change these settings individually for each build class via the Edit Build Class or New Build Class page.</p>
Resource Manager Type	<p>Specifies whether the Cluster Manager will use a resource manager, and if so, which one to use:</p> <ul style="list-style-type: none">• None (default)• Built-In• Priority Pool• LSF• Sun Grid Engine (SGE)• Amazon EC2 <p>For details about these options, see About Resource Managers below.</p> <p>If you select either the built-in or priority pool resource manager, a new Resources subtab appears under the Agents tab. If you select either the LSF, SGE, or Amazon Elastic Compute Cloud (EC2) resource manager, a new Cloud subtab appears under the Administration tab.</p> <p>If you cannot select either the LSF, SGE, or Amazon EC2 resource manager (because they are grayed out), see the “Configuring ElectricAccelerator” chapter in the <i>ElectricAccelerator Installation and Configuration Guide</i> at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html for details about enabling them.</p>
Resource Statistics Interval	Interval in minutes to collect resource usage statistics

Field	Description
Resource Statistics Keep	Number of days of resource usage statistics to keep
Message Log Keep	Number of days to keep message log entries. You should keep 30 days of message logs (but you should monitor message log sizes to ensure sufficient storage space)
Email Interval	Number of minutes between email notifications
Email Item Limit	Maximum number of messages in an email notification
Email From	Value to use in the From header element for the email notification
Email Prefix	String used to prefix subject lines
Report Usage Statistics?	<p>Opt in to or out of transmission of anonymous build performance and feature usage data to Electric Cloud. (Opt-out is not available for trial usage.) This data is used to plan the strategic direction of the Accelerator product line. Proprietary user data is never transmitted.</p> <p>Examples of data that is transmitted are build start times and durations, total workload, and conflict counts. Examples of data that is never transmitted are file names, file contents, and command-line arguments.</p>

Click **OK** to save your settings.

About Resource Managers

Accelerator resource management lets you select a subset of available agents for running builds. For example, a build might require a certain hardware feature. If so, Accelerator must use agents only on hosts containing that hardware.

Note: eMake automatically selects agents running on a compatible operating system. Resource managers do not override this feature, which means you can select a resource definition that contains incompatible hosts. Likewise, a resource definition can contain hosts with incompatible operating systems. In this case, Accelerator selects only those hosts with an operating system that satisfies the resource definition.

The following sections describe the resource managers that are available through the Accelerator web UI. (You can enable an additional resource manager through cmtool.)

Built-In

This resource manager is integrated with Accelerator and provides a straightforward way to associate hosts with a resource name. For example, for a build that requires a certain hardware feature, you can instruct Accelerator to use agents on hosts containing only that hardware.

Priority Pool

This resource manager lets you group resources into pools that can be prioritized differently among groups. For details about how to set up and use priority pools, see [Priority Pools](#).

LSF

This resource manager uses LSF grid management software to select agents for your build. For configuration details, see the *ElectricAccelerator Installation and Configuration Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.

Sun Grid Engine (SGE)

This resource manager uses SGE grid management software to select agents for your build. For configuration details, see the *ElectricAccelerator Installation and Configuration Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.

Amazon EC2

This resource manager uses the Amazon Web Services (AWS) EC2 service to select agents for your build. For configuration details, see the *ElectricAccelerator Installation and Configuration Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html.

If you choose the Amazon EC2 resource manager, the following additional menu and fields appear:

- **Region** pulldown menu—Choose the region, which is a named set of AWS resources in the same geographical area.
- **AWS Access Key ID** field—Enter the access key ID, which is the unique identifier that is associated with a secret access key. The access key ID and the secret access key are used together to sign programmatic AWS requests cryptographically.
- **AWS Secret Key** field—Enter the secret access key, which is used in conjunction with the access key ID to cryptographically sign programmatic AWS requests. Signing a request identifies the sender and prevents request alteration. You can generate secret access keys for your AWS account, individual IAM users, and temporary sessions.

To connect to AWS, you must choose a region, then enter the access key ID and secret access key, and then click **Connect**. Upon a successful connection, the following menus become enabled:

- **Instance Type**—Specification that defines the memory, CPU, storage capacity, and usage cost for an instance.
- **Security Group**—Named set of allowed inbound network connections for an instance. (Security groups in Amazon Virtual Private Cloud (VPC) also include support for outbound connections.)
- **Key Pair**—Set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key. You use the private key to create a digital signature, and then AWS uses the corresponding public key to validate the signature.
- **Availability Zone**—Distinct location within a region that is insulated from failures in other Availability Zones.
- **Subnet ID**—Segment of the IP address range of a VPC that Amazon EC2 instances can be attached to. You can create subnets to group instances according to security and operational needs.

After their corresponding cluster resource is done being used, the agents become idle and available for a subsequent build for 15 minutes until they are terminated. If a new build starts that uses that resource, new instances with new agents are provisioned.

For more information about Amazon EC2 and the terms used above, see the [Amazon Elastic Compute Cloud Documentation](#).

Database Configuration

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.

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.

License File Example

```
<?xml version="1.0" encoding="UTF-8"?>

<licenses>
  <license>
    <customerName>Electric Cloud, Inc.</customerName>
    <productName>ElectricAccelerator</productName>
    <featureName>Server</featureName>
    <expirationDate>2017-09-30</expirationDate>
    <gracePeriod>0</gracePeriod>
    <evaluation>false</evaluation>
    <properties>
      <property>
        <propertyName>maxAgents</propertyName>
        <propertyValue>128</propertyValue>
      </property>
      <property>
        <propertyName>jobcacheMax</propertyName>
        <propertyValue>5</propertyValue>
      </property>
      <property>
        <propertyName>concurrentBuilds</propertyName>
        <propertyValue>5</propertyValue>
      </property>
    </properties>
    <signature><signature></signature>
  </license>
</licenses>
```

License File Property 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. For contact information, see http://electric-cloud.com/support/ .)

Column	Description
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.
Concurrent Builds Max	<p>Maximum number of builds that are licensed to occur simultaneously. This number is read from the <code>concurrentBuilds</code> property in the current license file.</p> <p>If the license file has a <code><concurrentBuilds></code> property, the Cluster Manager limits the number of concurrent builds to that number. Any build that exceeds that number will queue until a license is available. If there is no <code><concurrentBuilds></code> property, or if the property is present but the value is <code>unlimited</code>, then the Cluster Manager places no limit on the number of concurrent builds.</p> <p>eMake reports delays because of license limitations via the <code>WARNING EC1321: No build license available, waiting...</code> message. A <code>WARNING EC1322: Build license allocated, proceeding.</code> message appears when a build in the queue is given a concurrent build license.</p> <p>A performance metric named <code>noLicenseWaitTime</code> indicates the time spent waiting for a build license because the number of concurrent builds has reached the license limit.</p>
Jobcache Max	<p>Maximum number of builds that are licensed to use the JobCache feature simultaneously. For first-time installations of Accelerator or most upgrades to Accelerator 9.1, this number is equal to the number of concurrent builds.</p> <p>This number is read from the <code>jobcacheMax</code> property in the current license file.</p> <p>Simultaneous builds in excess of this number will occur without using this feature.</p>
Actions	<p>View License—Opens the License Details page.</p> <p>Delete License—Deletes the Accelerator license. (A popup appears that asks you to confirm the deletion.)</p>

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 technical support or your sales representative. For technical support contact information, see <http://electric-cloud.com/support/>.

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 or instances running on LSF, SGE, or Amazon EC2 machines. The Cluster Manager is the only Accelerator component that communicates directly with LSF, SGE, or Amazon EC2 resource management software.

This page is absent until you configure the Cluster Manager server for LSF, SGE, or Amazon EC2 resource management software. If you have not done so, see the *ElectricAccelerator Installation and Configuration Guide* at http://docs.electric-cloud.com/accelerator_doc/AcceleratorIndex.html for instructions.

Resource Manager Example

The type of content in this tab varies with the type of resource manager that is enabled. Following is an example of this tab after the server is configured for the Amazon EC2 resource manager:

Electric Accelerator

Logged in as 'admin'

Logout

About

Help

Home

Builds

Agents

Reports

Messages

Administration

Permissions

Users

Groups

User Settings

Server Settings

Cloud

Database Configuration

Licenses

Amazon EC2 Information

Messages: 2017-12-11T21:23:31.571 DEBUG ip-172-31-43-113-1: hostName 'ip-172-31-43-113' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:25:04.609 DEBUG ip-172-31-39-42-1: hostName 'ip-172-31-39-42' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:25:27.710 DEBUG ip-172-31-37-218-1: hostName 'ip-172-31-37-218' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:25:31.576 DEBUG ip-172-31-43-113-1: hostName 'ip-172-31-43-113' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:27:04.617 DEBUG ip-172-31-39-42-1: hostName 'ip-172-31-39-42' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:27:27.716 DEBUG ip-172-31-37-218-1: hostName 'ip-172-31-37-218' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:27:31.582 DEBUG ip-172-31-43-113-1: hostName 'ip-172-31-43-113' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:43:31.655 DEBUG ip-172-31-43-113-1: hostName 'ip-172-31-43-113' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:45:04.707 DEBUG ip-172-31-39-42-1: hostName 'ip-172-31-39-42' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:45:27.827 DEBUG ip-172-31-37-218-1: hostName 'ip-172-31-37-218' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:45:31.664 DEBUG ip-172-31-43-113-1: hostName 'ip-172-31-43-113' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

2017-12-11T21:47:04.715 DEBUG ip-172-31-39-42-1: hostName 'ip-172-31-39-42' is in (CloudResource[linux, cfulmerAgent]) already. No more processing is needed.

Current Instances

Instance ID	Instance Status	Host Name	Platform	AMI Name	Submitted
i-05148034585d8e8b5	running	ip-172-31-37-218	Linux	cfulmerAgent	2017-12-11 20:52:51
i-0831f645ff9393a24	running	ip-172-31-43-113	Linux	cfulmerAgent	2017-12-11 20:54:51
i-0a9d5e214d8194a0a	running	ip-172-31-39-42	Linux	cfulmerAgent	2017-12-11 20:56:07
i-0831f645ff9393a05	running	ip-172-31-43-218	Linux	cfulmerAgent	2017-12-11 20:57:51
i-0a9d5e214d8194a24	running	ip-172-31-39-72	Linux	cfulmerAgent	2017-12-11 20:58:07

1 thru 5

Resource Manager Information Section

This section of the page contains messages about actions performed on LSF, SGE, or Amazon EC2. The title of this section indicates the resource manager type (for example, **Amazon EC2 Information**).

Current Jobs or Instances Section

This section of the page contains information about active LSF or SGE jobs or Amazon EC2 instances. The title of this section indicates whether the information is for jobs or instances (for example, **Current Instances** applies to Amazon EC2).

LSF or SGE Jobs

Column	Description
Job ID	ID assigned for each job that is running
Job Status	Job status returned from LSF or SGE (such as pending, running, done, or unknown)
Host Name	Name of the LSF or SGE host running the <code>agentconnect</code> script to communicate with the Cluster Manager
Platform	Platform of the LSF or SGE host running the <code>agentconnect</code> script
Resource Request	If you did not define a resource in your build class, the LSF or SGE master host chooses any available server, and the column will be blank. If you defined a resource name in the build class, this name is used by the LSF or SGE master host to choose servers to run the job
Stalled	Indicates if the job is stalled
Submitted	Date and time the job was submitted

Amazon EC2 Instances

Column	Description
Instance ID	Amazon EC2 instance ID assigned for each job that is running
Instance Status	Job status returned from Amazon EC2 (such as pending, running, done, or unknown)
Host Name	Internal DNS name of the instance. This is the name of the Amazon EC2 host running the <code>agentconnect</code> script to communicate with the Cluster Manager when the instance is launched

Column	Description
Platform	Platform of the Amazon EC2 instance
AMI Name	Amazon Machine Image (AMI) name. The AMI is the resource that you specified via the <code>--emake-resource</code> option when invoking eMake. This is the image that is used to launch the instance
Submitted	Date and time the instance was launched

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.