



## **ElectricAccelerator Help**

**Version 8.0**

**Electric Cloud, Inc.**  
35 South Market Street, Suite 100  
San Jose, CA 95113  
[www.electric-cloud.com](http://www.electric-cloud.com)

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

Published 4/1/2015

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

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

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

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

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

### **Trademarks**

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

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

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

# Contents

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

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

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

<b>Build Class Details</b>	<b>74</b>
Functionality	74
Details	74
Comments	74
<b>Build Classes - Create or Edit a Build Class</b>	<b>75</b>
Field Descriptions	75
<b>Agents</b>	<b>78</b>
Functionality	78
Column Descriptions	78
<b>Agent Details</b>	<b>80</b>
Functionality	80
Status	80
Configuration	80
Agent Console	80
Comments	80
<b>Agent Policies</b>	<b>81</b>
Agent Allocation Policy	81
Preemption Policy	81
Agent Lock Interval (seconds)	82
Deep/Wide Agent Allocation Policy	82
<b>Resources</b>	<b>83</b>
Functionality	83
Column Descriptions	83
<b>Resource Details</b>	<b>84</b>
Functionality	84
Details	84
Comments	84
<b>Resources - Create or Edit a Resource</b>	<b>85</b>
Field Descriptions	85
<b>Reports</b>	<b>86</b>
Report Descriptions	86
Agent Usage	86
Build Agent Allocation	86
Build Agent Allocation Per Day	86
Build Conflicts	87
Build Durations	87
Build Results	87
Build Results Per Day	87
Build Usage	87
Build Users	87
Build Wait Time	87
Cluster Usage	87
Cluster Utilization	88
Sea Level	88
Resource Usage over Time	88

Realtime Server Metrics .....	88
<b>Server Load .....</b>	<b>89</b>
About Server Load .....	89
Functionality .....	89
Column Descriptions .....	89
<b>Realtime Server Metrics .....</b>	<b>91</b>
<b>Messages .....</b>	<b>92</b>
Column Descriptions .....	92
Using the Custom SQL Filter .....	92
<b>Message Policies .....</b>	<b>94</b>
<b>Administration .....</b>	<b>95</b>
<b>Filters .....</b>	<b>96</b>
Filter Panel .....	96
Functionality .....	96
Column Descriptions .....	97
<b>Filters - Create or Edit a Filter .....</b>	<b>98</b>
Field Descriptions .....	98
<b>Permissions .....</b>	<b>99</b>
Functionality .....	99
Column Descriptions .....	99
<b>Edit User Permissions .....</b>	<b>102</b>
<b>Enable User .....</b>	<b>103</b>
<b>Edit Group Permissions .....</b>	<b>104</b>
<b>Enable Group .....</b>	<b>105</b>
<b>Users .....</b>	<b>106</b>
Functionality .....	106
Column Descriptions .....	106
User Details .....	106
<b>Users - Create or Edit Local Users .....</b>	<b>107</b>
Creating a User .....	107
Field Descriptions .....	107
Editing a User .....	107
<b>User Password .....</b>	<b>108</b>
<b>Groups .....</b>	<b>109</b>
Functionality .....	109
Column Descriptions .....	109
Group Details .....	109
Group Members .....	109
<b>Groups - Create or Edit a Local Group .....</b>	<b>110</b>
Group Members .....	110
Creating a Group .....	110
Editing a Group .....	110
Removing Users From a Group .....	110

<b>Add New User to Group</b>	<b>111</b>
Adding a User to a Group	111
<b>User Settings</b>	<b>112</b>
Editing User Settings	112
Field Descriptions	112
<b>Server Settings</b>	<b>113</b>
Field Descriptions	113
About Resource Managers	114
<b>Database Configuration</b>	<b>115</b>
Configuring Accelerator to Use an External Database	115
Backing Up and Restoring a Database	115
<b>Licenses</b>	<b>117</b>
Column Descriptions	117
<b>Import License</b>	<b>118</b>
Importing a License	118
<b>Cloud</b>	<b>119</b>
CloudHostManager Information	119
Current Jobs	119
<b>Error Codes</b>	<b>120</b>
EC1001	129
EC1002	129
EC1003	129
EC1004	130
EC1005	130
EC1006	131
EC1007	131
EC1010	132
EC1015	132
EC1016	133
EC1020	133
EC1021	134
EC1030	134
EC1031	135
EC1038	135
EC1039	136
EC1040	136
EC1042	137
EC1043	137
EC1045	138
EC1049	138
EC1050	138
EC1051	139
EC1052	139
EC1053	140
EC1054	140



EC1056 .....	140
EC1057 .....	141
EC1058 .....	141
EC1060 .....	142
EC1061 .....	142
EC1063 .....	142
EC1064 .....	143
EC1065 .....	143
EC1068 .....	144
EC1069 .....	144
EC1070 .....	144
EC1071 .....	145
EC1072 .....	145
EC1073 .....	145
EC1075 .....	146
EC1076 .....	146
EC1077 .....	147
EC1100 .....	147
EC1115 .....	148
EC1118 .....	148
EC1119 .....	148
EC1124 .....	149
EC1130 .....	149
EC1137 .....	150
EC1138 .....	150
EC1140 .....	151
EC1144 .....	151
EC1145 .....	152
EC1151 .....	152
EC1152 .....	153
EC1166 .....	153
EC1170 .....	153
EC1171 .....	154
EC1172 .....	154
EC1173 .....	155
EC1174 .....	155
EC1175 .....	156
EC1176 .....	156
EC1182 .....	156
EC1183 .....	157
EC1184 .....	157
EC1185 .....	157
EC1186 .....	158
EC1187 .....	158
EC1188 .....	159
EC1189 .....	159
EC1190 .....	159

EC1289 .....	160
EC2002 .....	160
EC2093 .....	160
EC2096 .....	161
EC2128 .....	161
EC2129 .....	162
EC2130 .....	162
EC2131 .....	163
EC2132 .....	163
EC2133 .....	163
EC2134 .....	164
EC2135 .....	164
EC2136 .....	164
EC2137 .....	165
EC3015 .....	165
EC3017 .....	165
EC3018 .....	166
EC3019 .....	166
EC3020 .....	166
EC3101 .....	167
EC3102 .....	167
EC3148 .....	168

# Welcome to ElectricAccelerator

---

## About ElectricAccelerator

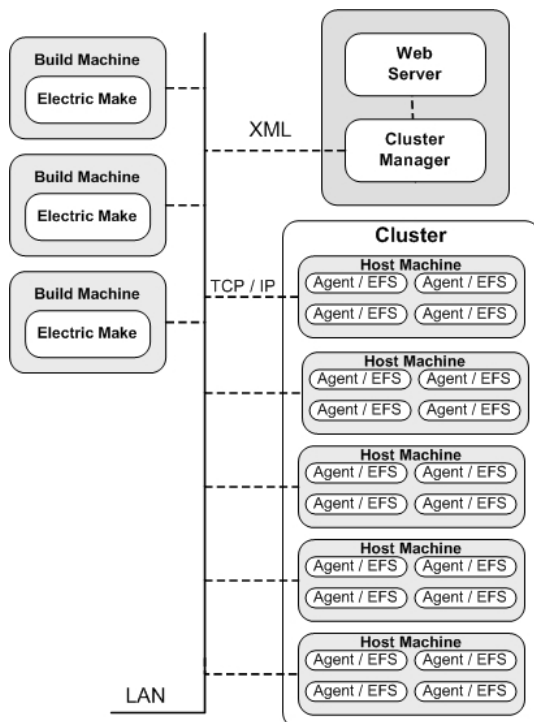
ElectricAccelerator® is a software build accelerator that dramatically reduces software build times by distributing the build over a large cluster of inexpensive servers. Using a patented dependency management system, ElectricAccelerator identifies and fixes problems in real time that would break traditional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

ElectricAccelerator components:

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

## Electric Make (eMake)

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



## Electric File System (EFS)

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

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

## ElectricAccelerator Agent (Agent)

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

## Cluster Manager

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

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

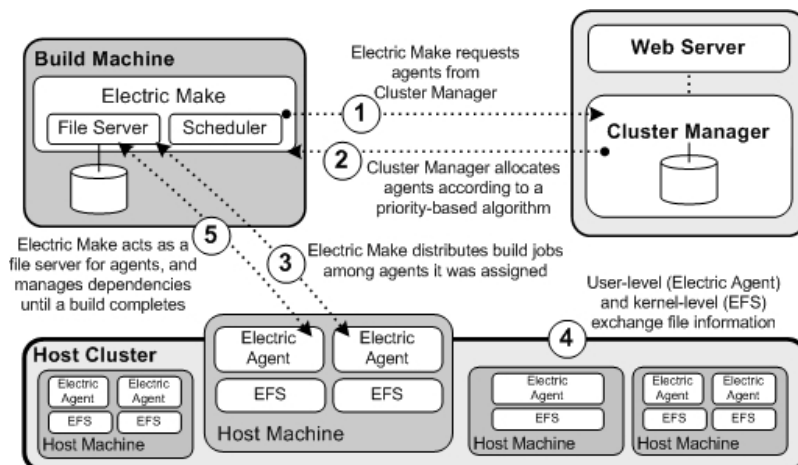
Cluster Manager also allows users to view upcoming and past builds, view the status of Agents in the cluster, and run reports.

## Electrify

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

## Cluster Manager and Other Components

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



## When to Use the ElectricAccelerator Cluster

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

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

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

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

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

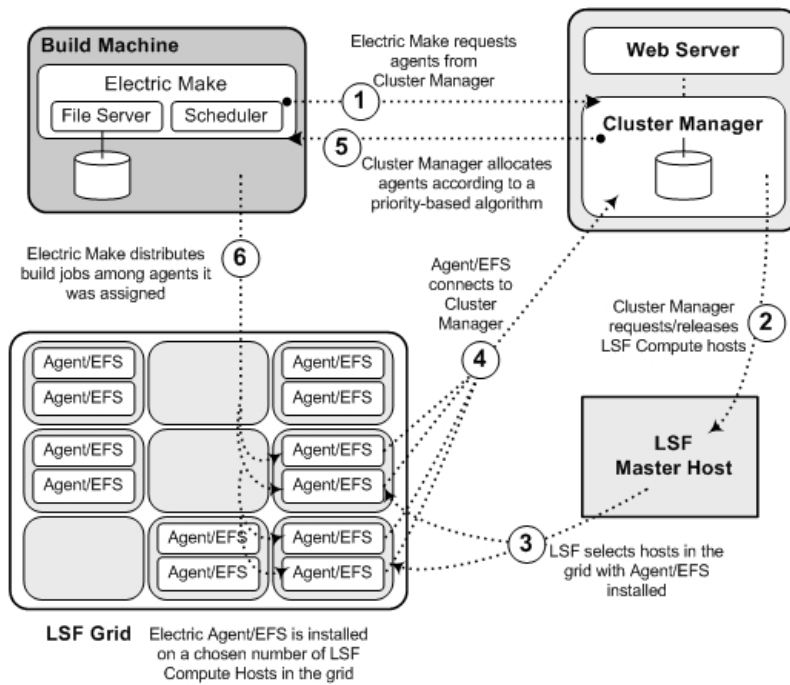
## Grid Integration

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

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

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

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



## ElectricAccelerator Installed Tools/Files

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

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

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

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



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

**Table 1:** `ecbreakpoint` command-line options

### Using ecbreakpoint

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

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

a:
    @ echo Before breakpoint
    ecbreakpoint

b:
    @ echo Some other job
```

**Figure 1:** Makefile using the `ecbreakpoint` program

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







	<code>chronic_ecloudmain.12505-200701091453</code>
	<code>chronic_641</code>
	<code>chronic_642</code>
	<code>chronic_641</code>
	<code>chronic_642</code>
	<code>chronic_ecloudmain.100000001-200701091454</code>

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	<a href="#">Retry</a>	<a href="#">Continue</a> <a href="#">Agent Command</a> <a href="#">Shell Command</a>

Figure 3: Breakpoint UI on the Build Details page

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

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

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

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

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

Break Point Details – 7

Details

Build ID: 185

Break Point Label:

Port: 41412

Agent Name: support-lin3-2

Agent IP Address: 192.168.26.235

Create Time: 2009-01-10 10:08:41

Command: pstree -alp

Agent CommandShell Command

Type	Request	Response	Requested By	Created Time
S	ps-tree -alp	init,1 -acpid,2214 -agent,2324 /opt/eccloud/686_Linux/bin/runagent -cm support-lin1.8030 -efsid 0 -numagents 4 -secureconsole n-d -execserver,1231 -sleep,1236 60 -agent,2341 /opt/eccloud/686_Linux/bin/runagent -cm support-lin1.8030 -efsid 1 -numagents 4 -secureconsole n-d -execserver,1225 -ecbreakpoint,1229 -pstree,1249 -alp -agent,2358 /opt/eccloud/686_Linux/bin/runagent -cm support-lin1.8030 -efsid 2 -numagents 4 -secureconsole n-d -execserver,1237 -sleep,1241 60 -agent,2375 /opt/eccloud/686_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.

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 `eccloud bin` directory. Figure 3 shows the IP address and port for `ecbreakpoint`. Use those values in `curl` and make sure to use the appropriate escape characters for spaces, slashes, and so on.  
Example for running the `ps` command on the agent:  
`http://192.168.26.235:41184/exec.xml?command=%2Fbi2Fps%20-ef`

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

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

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

**Table 2:**ecbreakpoint access URLs

## Labeling Breakpoints

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

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

a:
      @ ecbreakpoint --label $@

b:
      @ ecbreakpoint --label $@

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

**Figure 6:** Makefile with a label for each target

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

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

**Figure 7:** Labeled breakpoints on the Build Details page

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

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

**Figure 8:** Breakpoint labels with the same name

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

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

Figure 9: ecbreakpoint label option augmented with the working directory

Break Points									
Label	Agent Name	IP Address	Port	Created	Actions				
a.c_/u01 /ak /work/breakpoints /pattern/proj_a	support-lin3-4	192.168.26.235	44348	2009-01-11T09:08:53.000	Retry	Continue	Agent Command	Shell 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 `cmttool runAgentCmd` program to configure agent-side breakpoints (Figure 1).

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

**Figure 1:** Session commands to manipulate agent-side breakpoints

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

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

**Table 1:** Breakpoint associative array

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

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

**Figure 2:** Agent breakpoint triggered by regex on stdout

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

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

**Figure 3:** Makefile to generate error

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

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

**Figure 4:** Cluster Manager UI for agent-side breakpoints

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

**Figure 5:** /var/log/ecagent4.log

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



```

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

```

**Figure 6:** Agent breakpoint that initiates an arbitrary program

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

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

**Figure 7:** Breakpoint list on the Build Details page

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

## Specifying a Breakpoint Through a File

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

Example breakpoint contained in a file named test.txt

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

Example of passing in the breakpoint contained in test.txt

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

# Troubleshooting Breakpoints

---

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

## Breakpoint Does Not Trigger

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

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

## Breakpoint regex Is Wrong

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

## Output Is Not as Expected

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

## Use Your Editor to Run Breakpoints

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

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

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

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

## Using Cluster Manager Administration Tools

---

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

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

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

Specific command-line tool functions are:

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

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

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

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

## Using `clusterexec`

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

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

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

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

Each command may be the following type:

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

Some platform-specific rules to observe:

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

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

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

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

`clusterexec` **Command-line Options**

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

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

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

#### eRunner Built-in Commands

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

Commands	Description
<code>restartEranner</code>	Restart eRunner service on the hosts.
<code>printVersion</code>	Print the eRunner service version on the hosts.
<code>logLevel &lt;level&gt;</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=&lt;value&gt;</code>	Set of hosts within the cluster to update. <i>value</i> should be in the form "host1 host2[:port] ..."

Switch Options	Description
<code>--cm=&lt;host[:port]&gt;</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=&lt;value&gt;</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=&lt;value&gt;</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 =&gt; 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=&lt;value&gt;</code>	Aborts execution after <i>value</i> seconds.

## Using clusterdownload

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

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

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

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

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

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

-- or use --

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

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

Switch option values are described in the following table:

`clusterdownload` **Command-line Options**



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

Switch Options	Description
<code>--include=&lt;pattern&gt;</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=&lt;file&gt;</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=&lt;info debug&gt;</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 &lt;host:port&gt;</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 &lt;number&gt;</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 "&lt;user1&gt;/&lt;password1&gt; &lt;user2&gt;/&lt;password2&gt;...&lt;userN&gt;/&lt;passwordN&gt;"</code>	(Windows platforms) Specifies multiple agent user names and passwords. The quotes are required. <b>Note:</b> You must specify the same number of users as the actual number of agents.
<code>-agentuser "&lt;agentID&gt; &lt;user&gt;/&lt;password&gt;"</code>	(Windows platforms) Specifies the user name and password for a specific agent. <code>&lt;agentID&gt;</code> is the numerical ID of the agent to configure (between 1 and the actual number of agents). The quotes are required.
<code>-lsfhost &lt;value&gt;</code>	Specifies whether this machine is part of an LSF grid. Enter <code>y</code> (yes) or <code>n</code> (no).
<code>-maxopenfds &lt;value&gt;</code>	Set the maximum number of open file descriptors for agents. To clear the number and revert to the OS default, enter <code>ecconfig -maxopenfds ""</code> . The value is recorded via the <code>MAX_OPEN_FDS</code> field in the <code>ecagent.conf</code> file. The recommended value is 8192.
<code>-secureconsole &lt;value&gt;</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 &lt;path&gt;</code>	Location where agents store temporary files. To revert to the system default, use double quotes as the argument ( <code>-tempdir ""</code> ). The system default is <code>/tmp</code> on Linux and Solaris and <code>C:\WINDOWS\Temp</code> on Windows.
<code>-help, -?</code>	Displays the <code>ecconfig</code> command usage message.

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

## Using ecagent Tools

---

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

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

## Using ecinstconf

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

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

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



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

## Using eclock

---

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

An eclock example:

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

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

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

**Note:** Mutex names are case sensitive.

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

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

The following help text is the eclock Windows help.

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

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

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

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

## Using the runagent.local File

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

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

### Preventing "runaway" Jobs

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

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

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

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

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

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

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

### Configuring Stalled Job Detection Settings

This is a Windows-only setting.

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

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

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

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

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

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

## Configuring Agent Log Rotation

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

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

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

Use this format: `session logger <option>`

For example: `session logger retaincount 10`

Available log rotation options

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

---

# System Maintenance Tasks

---

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

## Cleaning Up Data

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

## Deleting from the Builds List

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

## Deleting Extraneous or Obsolete Builds

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

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

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

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

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

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

## Managing the Message Log

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

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

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

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

## Securing the Cluster Manager Database

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

```
ACCELERATOR_JDBC_USER_NAME  
ACCELERATOR_JDBC_PASSWORD
```

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

# Database Configuration

## Configuring Accelerator to Use an External Database

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

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

1. Fill in the fields as follows:

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

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

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

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

## Backing Up and Restoring a Database

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

### Backing Up Your Database

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

```
cmtool exportData <file_name>
```

where `<file_name>` is the file name or path to export to. If you use a file name, the destination is the current working directory of the Java process; for example, `/opt/eccloud/i686_Linux` or `C:\ECloud\i686_win32`. If you use a path, the `cmtool` login user must have write access to that path.

**Note:** This command initiates a full database dump, so a large database might require an extended period of time to export.

### Restoring Your Database

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

```
cmtool importData <file_name>
```

where `<file_name>` is the name of the file to import. The file's path is relative to the current working directory of the Java process; for example, `/opt/eccloud/i686_Linux` or `C:\ECloud\i686_win32`.

**Note:** This command initiates a full database dump, so a large database might require an extended period of time to import.

Make sure that you

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



# Enabling LDAP

---

To enable LDAP, follow this procedure:

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

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

## Import License

---

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

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

### Importing a License

Do one of the following:

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

## Using Comments

---

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

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

### Adding a New Comment

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

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

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

### Using HTML for Comments

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

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

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

### Editing a Comment

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

### Deleting a Comment

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

## Stopping Builds

---

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

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

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

### Stopping a Single Build from the ElectricAccelerator Web Interface

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

### Stopping Multiple Builds from the Web Interface

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

### Stopping a Build Using `cmtool`

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

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

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

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

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

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

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

To get failed builds:

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

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

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

For example, to end build 4458, type:

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

## Deleting Builds

---

### Deleting a Single Build

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

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

### Deleting a Group of Builds from the Build List

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

## Creating Build Classes

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

This takes you to an empty build class details page.

2. Name the build class.

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

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

3. Enter the build class tag definition.

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

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

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

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

### Build Class Examples

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

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

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

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

```
Supersoftware_Win_v.2.1_12345_WED_AUG_22_14_37_12
```

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

```
Supersoftware_Sol_v.1.7_12356_WED_AUG_23_11_14_39
```

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

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

```
MegaSoftware_Win_v.1.3_12356_JSMITH_20050411100838
```

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

## Editing Build Classes

---

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

or

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

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

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

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

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



## Deleting Build Classes

---

### Deleting a Single Build Class

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

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

### Deleting Multiple Build Classes

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

# Agent Policies

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

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

## Agent Allocation Policy

Before selecting the agent allocation policy, consider these choices:

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

## Preemption Policy

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

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

### Preemption Policy Field Descriptions

Setting	Description
Always	If reassigning an agent would drop a build below its minimum number of agents, agents <b>cannot be taken</b> . If reassigning an agent would leave a build with at least its minimum number of agents, <b>unlocked agents can always be taken</b> .
Priority	If reassigning an agent drops a build below its minimum number of agents, agents <b>can be taken</b> 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, <b>unlocked agents can be taken</b> from a lower priority build or an equivalent priority build.
Never	Agents <b>can never be taken</b> .

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

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

### Preemption Policy Scenario 1:

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

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

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

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

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

### Preemption Policy Scenario 2:

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

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

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

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

### Agent Lock Interval (seconds)

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

### Deep/Wide Agent Allocation Policy

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

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

Click **OK** to save all selections.

## Understanding a Cluster Sharing Policy

---

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

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

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

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

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

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

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

## Modifying the Number of Agents per Host

---

On Linux and Solaris platforms, follow these steps:

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

For example, `AGENT_NUMBER=2`

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

For example, `ecconfig -numagents 2`

## Enabling Agents

---

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

### Enabling a Single Agent

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

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

### Enabling a Group of Agents

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

## Disabling Agents

---

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

### Disabling a Single Agent

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

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

### Disabling a Group of Agents

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



## Testing Agents

---

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

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

### Checking the Status of a Single Agent

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

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

### Checking the Status of a Group of Agents

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

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

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

## Deleting Agents

---

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

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

### Deleting a Single Agent

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

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

### Deleting a Group of Agents from the Agent List

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

This removes the agents from the database.

### Removing an Agent from the Cluster Permanently

You must do one of the following:

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

## Creating Resources

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

This takes you to an empty resource details page.

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

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

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

3. Add a description if you wish.

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

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

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

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

### Host Name and Glob-Style Pattern Example 1

Resource Definition:

Name - "Xcard"

Hosts - "earth;wind;fire"

eMake execution:

```
--emake-resource=Xcard
```

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

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

### Host Name and Glob-Style Pattern Example 2

Resource Definition:

Name - "Xsoft"

Hosts - "earth;asteroid\*"

Build Class Definition:

Name - "Xbuilds"

Resource Request - "Xsoft"

eMake execution:

```
--emake-class=Xbuilds
```

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

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

## Editing Resources

---

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

or

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

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

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

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

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

## Deleting Resources

---

### Deleting a Single Resource

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

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

### Deleting Multiple Resources

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

## Priority Pools

---

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

### Using Priority Pools

Follow this procedure:

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

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

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

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

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

### Use Case 1 - High Performance Builds

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

Pool makeup:

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

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

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

## Use Case 2 - Multiple Departments

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

Pool makeup:

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

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

Launching eMake with <code>--emake-resource=</code>	means the build uses
<code>"Deptb:"</code>	Pool <code>"Deptb"</code> 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 <code>"Depta"</code> 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 <code>"64bit"</code> (six machines). If Department B is running a build, only <code>"a_17_64"</code> through <code>"a_20_64"</code> (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

---

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

or

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

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

The query value is a SQL WHERE clause.

7. Click **OK** when finished.

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

## Editing Filters

---

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

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

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

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

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

# Deleting Filters

---

## Deleting a Single Filter

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

## Deleting Multiple Filters

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

## Generating Reports

---

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

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

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

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

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

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

## Adding Custom Reports

---

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

The arguments are (in this order):

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

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

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

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

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

## Making Custom Reports Available to the Cluster Manager

To make the report available:

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

## Known Issues

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

## The Home Page

---

The Home page, and all product web pages, provide one-click access to Builds, Agents, Reports, Messages, and Administration information and functions—the tabs across the top of the page are always there for quick navigation.

### On the Top Bar

The top bar also provides the following links:

- **Logout** - Logs out the current user.
- **About** - Details which ElectricAccelerator version is running on your server.
- **Help** - Opens page-specific help throughout the Accelerator web interface.

Accelerator also supplies overview/concept help topics that are not linked to a particular web page. To see these help topics, click **Help** from any web page to open the help system. The left-pane (table of contents) contains the list of help topics.

### Server Information

The Home page provides the following at-a-glance information:

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

### Comments

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

## Builds

The Builds page allows you to see:

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










### Functionality

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

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

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

### Column Descriptions

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

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



## Build Details

---

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

### Functionality

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

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

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

### Details

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

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

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

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

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

### Comments

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

# Build Classes

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

## About Build Classes

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

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

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

## Functionality

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

## Column Descriptions

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

Column	Description
Priority	Class' build priority level. Classes have default priorities and boost values. Boost values range from -10 to +10 (default 0) where a higher boost value means that builds in that class can use available agents ahead of builds with the same priority and lower boost. Read about this setting's role in <a href="#">agent policies</a> .
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 <a href="#">agent policies</a> .
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	<b>Edit Build Class</b> —Takes you to the corresponding Edit Build Class page. Your existing build class specifications populate the appropriate fields. <b>Delete Build Class</b> —Deletes the corresponding build class.

# Build Class Details

---

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

## Functionality

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

## Details

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

## Comments

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

## Build Classes - Create or Edit a Build Class

---

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

### Field Descriptions

Field	Description
Name	Name of the build class.

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

Field	Description
Priority	Priority for builds in this build class. Select Low, Normal, or High, and if using a boost value, use the adjacent pull-down to select a boost value. For more information, see <a href="#">agent policies</a> .
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 <b>New Resource</b> link to define resources.)
Allow jobcache	<p>Checkbox that determines whether to allow the JobCache feature to be used for this build class. For the default build class and for build classes that exist when you upgrade to Accelerator 8.0, this checkbox is checked (meaning that JobCache is allowed) by default. For a build class that you create, this checkbox is unchecked by default.</p> <p>JobCache is an add-on option. It is available only on Linux and Solaris platforms and requires separately-purchased licensing. For more information about JobCache, see the <i>ElectricAccelerator Electric Make User Guide</i>.</p>
Annotation Upload	Checkbox that determines whether to upload the annotation file to the Cluster Manager.
Annotation Levels	<p>Checkboxes that indicate which levels of information to include in the annotation file. Available annotation levels are as follows:</p> <ul style="list-style-type: none"> <li>• Basic Annotation (if the JobCache add-on option is enabled, basic annotation includes JobCache annotation)</li> <li>• Env (extended environment information)</li> <li>• History (serialization details)</li> <li>• File (files read or written)</li> <li>• Lookup (all file names accessed)</li> <li>• Waiting (complete dependency graph)</li> <li>• Registry (registry updates for Windows)</li> </ul>

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

# Agents

The Agents page allows you to view all configured agents.

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




## Functionality

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

## Column Descriptions

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



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

# Agent Details

---

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

## Functionality

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

## Status

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

## Configuration

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

## Agent Console

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

## Comments

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

## Agent Policies

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

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

### Agent Allocation Policy

Before selecting the agent allocation policy, consider these choices:

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

### Preemption Policy

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

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

#### Preemption Policy Field Descriptions

Setting	Description
Always	If reassigning an agent would drop a build below its minimum number of agents, agents <b>cannot be taken</b> . If reassigning an agent would leave a build with at least its minimum number of agents, <b>unlocked agents can always be taken</b> .
Priority	If reassigning an agent drops a build below its minimum number of agents, agents <b>can be taken</b> 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, <b>unlocked agents can be taken</b> from a lower priority build or an equivalent priority build.
Never	Agents <b>can never be taken</b> .

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

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

#### Preemption Policy Scenario 1:

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

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

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

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

### Preemption Policy Scenario 2:

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

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

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

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

## Agent Lock Interval (seconds)

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

## Deep/Wide Agent Allocation Policy

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

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

Click **OK** to save all selections.

## Resources

---

This page displays all defined resources.

### Functionality

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

### Column Descriptions

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

# Resource Details

---

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

## Functionality

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

## Details

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

## Comments

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

## Resources - Create or Edit a Resource

---

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

### Field Descriptions

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

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

# Reports

---

Report results are displayed on this page. Your choices are:

- Agent Usage
- Build Agent Allocation
- Build Agent Allocation Per Day
- Build Conflicts
- Build Durations
- Build Results
- Build Results Per Day
- Build Usage
- Build Users
- Build Wait Time
- Cluster Usage
- Cluster Utilization
- Sea Level
- Resource Usage Over Time
- Realtime Server Metrics

## Report Descriptions

### Agent Usage

This report is a chart showing agents' per day availability for the specified duration. The relevant values are agent shortage (minimum, maximum, and average), which is the number of agents that builds could have used. A sustained level of agent shortage indicates that the cluster is probably too small. The chart also shows the number of agents available, which should remain fairly constant at the number of agents in the cluster. Discrepancies could result from one of the following: network issues may be causing agents to become unavailable, there may be maintenance performed on agents, or there may be an issue with the agents themselves.

### Build Agent Allocation

This report is a pie chart showing agent allocation for builds over a specified time period. Agent allocation is a value computed by eMake that describes how often eMake was able to get the agents it wanted or could have. The number 100 indicates the build had the possible agents 100% of the time, while 0 indicates the build did not get any agents. 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 is most likely too small.

The default report displays the following allocation ranges: 0-20%, 20-40%, 40-60%, 60-80%, and 80-100%.

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]

### Build Agent Allocation Per Day

This report is the same as Build Agent Allocation, except it is a bar chart showing agent allocation per day.



## Build Conflicts

This report is a bar chart showing the number of builds that had a specific range (increments of 100) of conflicts. Conflicts affect performance and indicate the use of an out-of-date history. If a large number of builds have high conflict counts (> 20-30) check the history files used for those builds. One method to discover the cause of the conflicts is to run builds with annotation (if updating the history does not resolve this issue).

## Build Durations

This report is a chart showing the build durations of a specified time period. Durations are grouped in 10-minute increments by default. You can change the default value (in minutes) in the Granularity field. This report is most useful to identify extremes, especially when build classes can be used to group similar builds together.

## Build Results

This report is a pie chart illustrating the distribution between results of different builds. Some possible groups are:

- Stopped - The build was stopped by the user.
- Timed Out - The build was timed-out by the Cluster Manager because it lost connection to the eMake machine.
- Error - eMake returned an error code.  
For errors, the legend contains one entry per error code.
- OK - The eMake invocation finished without any errors.

The default report displays the following groups: Stopped, Timed Out, Error: 1-255, and OK

By default, the Ranges field contains: Stopped,[257,257],Timed Out,[256,256],Error: 1-255,[1,255],OK,[0,0]

## Build Results Per Day

This report is the same as Build Results, except it is a bar chart showing build results per day.

## Build Usage

This report is a chart showing how many builds were running on the cluster at the same time during a specified period. It also shows the agent shortage during the time period, which allows the correlation between over-subscription of the cluster and cluster size.

## Build Users

This report is a list showing build statistics by user name/host name/IP. Statistics include how many builds were run, their duration, wait time, and workload.

## Build Wait Time

This report is a bar chart showing the wait time for the builds of a specified period. Wait time is grouped in 10 minute increments. Wait time is the time when eMake does not have any agents available to run the build. A high number of builds with a non-zero wait time indicates that the cluster may be too small.

## Cluster Usage

This report presents a view into statistical data relating to cluster utilization, using data collected by the Cluster Manager. The report lists all available data points, and these may be too numerous to show in the chart at one time, so it is recommended to specify a short time period. For a specific date range, data is presented in a line chart, illustrating the change of certain key properties in the server as follows:

- Avg. Agent Shortage - If this value is more than 0, at least one build could have used more agents, but the cluster did not have enough agents. Generally, this means more agents need to be purchased.

- Avg. Agents Available - This is a view of how many agents were available at one time.
- Avg. Concurrent Builds - This view shows the number of builds running at a particular time.

## Cluster Utilization

This report represents the number of requested agents at any given moment, averaged over the set of days represented in the Cluster Manager data. Averaging provides mapping of a "virtual" 24-hour period. Because the information is averaged, utilization may appear to be lower than the actual cluster size.

To view this report, you must have Read permission for Resources and Reports. If a user does not have Read permission for both, reports are still available but the following message displays: "Error: AccessDenied: User 'XXX' does not have the required ResourcesRead privileges to perform this operation."

## Sea Level

This report plots 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. If a user does not have Read permission for both, reports are still available but the following message displays: "Error: AccessDenied: User 'XXX' does not have the required ResourcesRead privileges to perform this operation."

## Resource Usage over Time

This report shows the number of simultaneous builds that ran on each resource over a given period of time.

Each color represents the number of simultaneous builds on a resource. This report includes information for all builds that you have Read permission for.

## Realtime Server Metrics

To view these reports, 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.

# Server Load

---

## About Server Load

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

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

## Functionality

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

## Column Descriptions

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

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

## Realtime Server Metrics

---




This page can display a number of server metrics:

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

# Messages

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

## Column Descriptions

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

## Using the Custom SQL Filter

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

The following fields are available for custom SQL filters:

- id
- text
- severity
- agent\_name
- create\_time
- build\_id

### Examples

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

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

For examples of filters that use the following fields see:

- severity: Warnings
- agent\_name: Messages By Agent
- create\_time: Messages By Date
- build\_id: Messages By Build ID

## Message Policies

---

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

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



## Administration

---

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

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

# Filters

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

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

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

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

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

## Filter Panel

The Filter panel always has the following functions:

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



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

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

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

## Functionality

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

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

## Column Descriptions

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

## Filters - Create or Edit a Filter

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

### Field Descriptions

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

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

where

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

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

## Permissions

---

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

### Functionality

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

### Column Descriptions

The following column heading descriptions apply to either table.

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

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

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

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

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

## Edit User Permissions

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

Permissions control these respective features:

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

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



## Enable User

---

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

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

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

## Edit Group Permissions

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

Permissions control these respective features:

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

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

## Enable Group

---

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

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

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

## Users

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

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

### Functionality

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

### Column Descriptions

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

### User Details

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

## Users - Create or Edit Local Users

---

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

### Creating a User

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

### Field Descriptions

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

### Editing a User

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

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

## User Password

---

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

Click **OK** to save the change.

## Groups

---

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

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

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

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

### Functionality

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

### Column Descriptions

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

### Group Details

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

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

### Group Members

This table lists all members who belong to this group.

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

## Groups - Create or Edit a Local Group

---

### Group Members

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

### Creating a Group

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

### Editing a Group

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

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

### Removing Users From a Group

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



## Add New User to Group

---

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

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

### Adding a User to a Group

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

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

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

This displays your filtered list.

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

Now the selected users are added to the group.

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

# User Settings

---

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

## Editing User Settings

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

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

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

2. Make any necessary changes.

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

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

## Field Descriptions

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

## Server Settings

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

### Field Descriptions

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

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

Click **OK** to save your settings.

## About Resource Managers

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

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

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

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

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

# Database Configuration

---

## Configuring Accelerator to Use an External Database

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

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

1. Fill in the fields as follows:

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

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

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

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

## Backing Up and Restoring a Database

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

### Backing Up Your Database

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

```
cmtool exportData <file_name>
```

where `<file_name>` is the file name or path to export to. If you use a file name, the destination is the current working directory of the Java process; for example, `/opt/eccloud/i686_Linux` or `C:\ECloud\i686_win32`. If you use a path, the `cmtool` login user must have write access to that path.

**Note:** This command initiates a full database dump, so a large database might require an extended period of time to export.

### Restoring Your Database

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

```
cmtool importData <file_name>
```

where `<file_name>` is the name of the file to import. The file's path is relative to the current working directory of the Java process; for example, `/opt/eccloud/i686_Linux` or `C:\ECloud\i686_win32`.

**Note:** This command initiates a full database dump, so a large database might require an extended period of time to import.

Make sure that you

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

## Licenses

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

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

### Column Descriptions

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

# Import License

---

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

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

## Importing a License

Do one of the following:

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



## Cloud

---

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

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

### CloudHostManager Information

This section contains messages about actions performed on the grid.

### Current Jobs

Column heading descriptions:

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

## Error Codes

---

<b>1000 (eMake Error Messages)</b>	<b>2000 (Agent Error Messages)</b>	<b>3000 (CM Error Messages)</b>
------------------------------------	------------------------------------	---------------------------------

<p>EC1001 The 'exclusive' priority is deprecated. Using 'high' instead.</p> <p>EC1002 The command line option <code>--make-annotate</code> is deprecated.</p> <p>EC1003 Ignoring redundant <code>EMAKE_ROOT</code> value.</p> <p>EC1004 Could not create temp dir.</p> <p>EC1005 No writable temporary directories are available.</p> <p>EC1006 Bad value for <code>--make-cygwin</code>.</p> <p>EC1007 Unknown annotation detail token.</p> <p>EC1008 Invalid logcapture setting. (OBSOLETE)</p> <p>EC1009 Unknown pragma.</p> <p>EC1010 Unknown history mode.</p> <p>EC1011 Invalid compression thresholds.</p> <p>EC1012 Invalid write limit: must be 1 or greater.</p> <p>EC1013 Unknown make emulation mode option.</p> <p>EC1014 <code>--make-debugger</code> not supported for cluster builds.</p> <p>EC1015 Invalid emake root directory.</p> <p>EC1016 Bad root (e.g. cannot handle UNC path names).</p> <p>EC1017 Bad root <code>" / "</code>: the root directory is not a supported emake root.</p> <p>EC1018 Unable to load <code>cygwin1.dll</code>.</p> <p>EC1019 Unable to open annotation output file.</p> <p>EC1020 Local mode only supports 'basic' annotation detail.</p> <p>EC1021 Local mode does not support setting an annotation file name.</p> <p>EC1022 Illegal registry path:</p>	<p>EC2002 Unknown command code <code>0x%x</code> (last command was <code>%s</code>).</p> <p>EC2003 Session::doCommand received name record whose <code>EfsFileName</code> would not fit in a single page.</p> <p>EC2004 Bad root <code>"[root]"</code>: UNC paths aren't allowed (root must be in a drive such as <code>C: /</code>).</p> <p>EC2005 Bad root <code>"[root]"</code>: must start with a drive name such as <code>C:/</code>.</p> <p>EC2006 Bad root <code>"[root]"</code>: this would interfere with the Electric Agent's state files stored in <code>"[dir]"</code>.</p> <p>EC2007 Bad root <code>"[root]"</code>: this would interfere with the system root <code>"[dir]"</code>.</p> <p>EC2008 Bad root <code>"[root]"</code>: this would interfere with the temporary directory <code>"[dir]"</code>.</p> <p>EC2009 Bad root <code>"[root]"</code>: there is a CD-ROM occupying this drive letter on the node.</p> <p>EC2010 Bad root <code>"[root]"</code>: there is a removable disk occupying this drive letter on the node.</p> <p>EC2011 pipeline error.</p> <p>EC2012 Socket already open.</p> <p>EC2013 Couldn't create client socket: [error].</p> <p>EC2014 Couldn't find host <code>"[host]"</code>: [error].</p> <p>EC2015 Couldn't connect to server: [error].</p> <p>EC2016 Couldn't set close-on-exec for socket: [error].</p> <p>EC2017 No connection open.</p> <p>EC2018 No connection open.</p> <p>EC2019 Error reading from socket: [error].</p> <p>EC2020 I/O error sending to socket: [error].</p> <p>EC2021 Error polling for socket to become writable: [error].</p> <p>EC2022 Couldn't create server socket: [error].</p> <p>EC2023 Couldn't set close-on-</p>	<p>EC3001 Failed to initialize the curl handle.</p> <p>EC3002 Request failed.</p> <p>EC3003 Response does not have <code>&lt;ECloud&gt;</code> tag.</p> <p>EC3004 XML parser threw exception: [err] (XML is <code>'[xml]'</code>).</p> <p>EC3005 <code>doSendHTTP</code>: You must call <code>connect()</code> first.</p> <p>EC3006 Couldn't resolve CM host.</p> <p>EC3007 Invalid password for user 'emake'.</p> <p>EC3008 HTTP code [string].</p> <p>EC3009 Unable to connect to CM.</p> <p>EC3010 Sending HTTP command failed with curl error ([code]): [error].</p> <p>EC3011 <code>doSendHTTP</code>: You must call <code>connect()</code> first.</p> <p>EC3012 Unable to open file <code>"[file]"</code>.</p> <p>EC3013 CM refused <code>notifyBuild</code> request.</p> <p>EC3014 Error code is not a valid int.</p> <p>EC3015 Potentially unsafe mount directory.</p> <p>EC3016 Unrecognized emake-debug option <code>"[option]"</code>.</p> <p>EC3017 <code>ClusterManagerConnection::downloadFile</code>: You must call <code>connect()</code> first!</p> <p>EC3018 unable to open file <code>"[file]"</code></p> <p>EC3019 unable to close file <code>"[file]"</code></p> <p>EC3020 Couldn't receive CM response</p> <p>EC3101 <code>BuildStopped</code>: Build X was stopped from the cluster manager.</p> <p>EC3102 <code>BuildTimedOut</code>: Build X was stopped because it timed out.</p> <p>EC3148 Couldn't start cluster build</p>
--	---	--

<p>root must be one of: HKEY_CLASSES_ROOT or HKEY_LOCAL_MACHINE.</p> <p>EC1023 Illegal build mode, switching to 'local'.</p> <p>EC1024 Invalid EMAKE_ROOT directory.</p> <p>EC1025 Unknown make emulation mode in emulation table.</p> <p>EC1026 Cannot open equivalence map: [file] (OBSOLETE)</p> <p>EC1027 Invalid file in equivalent map: [error] (OBSOLETE)</p> <p>EC1028 Build is stopping.</p> <p>EC1029 Unable to determine name of command file.</p> <p><a href="#">EC1030 Unable to open command file.</a></p> <p><a href="#">EC1031 Unable to write command file.</a></p> <p>EC1032 An exception occurred.</p> <p>EC1033 Unable to create temporary file.</p> <p>EC1034 The last 60 agent requests to the cluster manager have failed.</p> <p>EC1035 Couldn't remove backing store.</p> <p>EC1036 Chain::getVersion detected a malformed chain, treating as a stale version.</p> <p>EC1037 Cannot open history file.</p> <p><a href="#">EC1038 Version mismatch, ignoring history file.</a></p> <p><a href="#">EC1039 Ignoring malformed history file: no Section header.</a></p> <p><a href="#">EC1040 Ignoring malformed history file: a bad Job section header.</a></p> <p>EC1041 Ignoring malformed history file: no Job section.</p> <p><a href="#">EC1042 Ignoring malformed history file: no Serial Deps section.</a></p>	<p>exec for server socket: [error].</p> <p>EC2024 Couldn't bind server socket to port: [error].</p> <p>EC2025 Couldn't query port number from socket: [error].</p> <p>EC2026 Couldn't listen on server socket: [error].</p> <p>EC2027 Server got error during accept: [error].</p> <p>EC2028 Couldn't set close-on-exec for server socket: [error].</p> <p>EC2029 Couldn't read disk cache table: improper format.</p> <p>EC2030 Couldn't open "[comm file]" for writing: [error].</p> <p>EC2031 EfsCommander couldn't set close-on-exec: [error].</p> <p>EC2032 EfsCommander couldn't close mHandle: [error].</p> <p>EC2033 EfsCommander::flush couldn't write to EFS: [error].</p> <p>EC2034 EfsCommander was passed malformed registry key, must be absolute object path.</p> <p>EC2035 FileUsageWithData::emitFileData found NULL data locator, unable to send file data.</p> <p>EC2036 FileUsageWithData::emitFileData found record with incorrect type, unable to send file data.</p> <p>EC2037 pipeline error.</p> <p>EC2038 Mount::set couldn't open repository "[repo]": [error].</p> <p>EC2039 Mount::set couldn't create directory "[dir]": [error].</p> <p>EC2040 Couldn't read mount file: improper format.</p> <p>EC2041 Unable to create agent HTTP listener: [error].</p> <p>EC2042 Repository unable to create agent file "[file]": [error].</p> <p>EC2043 Repository couldn't write agent file "[file]": [error].</p> <p>EC2044 Couldn't read repository: [error].</p>	
---	--	--

<p>EC1043 Ignoring malformed history file: a bad Serial Deps section.</p> <p>EC1044 Ignoring malformed history file: no Dependencies.</p> <p>EC1045 Ignoring malformed history file: no Prereq Dependencies.</p> <p>EC1047 more output fragments than expected.</p> <p>EC1048 No agents available; aborting build.</p> <p>EC1049 unexpected exception halting build.</p> <p>EC1050 agent error; aborting build.</p> <p>EC1051 Unable to create link.</p> <p>EC1052 Unable to rename directory.</p> <p>EC1053 Unable to read input stream.</p> <p>EC1054 Unable to determine current working directory.</p> <p>EC1055 unexpected exception halting build.</p> <p>EC1056 Unable to read directory.</p> <p>EC1057 Unable to read ACL for directory.</p> <p>EC1058 Unable to read backing store file.</p> <p>EC1059 Unable to read backing store file.</p> <p>EC1060 Unable to open file.</p> <p>EC1061 Unable to modify the file times.</p> <p>EC1062 Unable to modify attributes of file.</p> <p>EC1063 Unable to create directory.</p> <p>EC1064 Unable to modify the file times of file.</p> <p>EC1065 Unable to modify attributes of file.</p> <p>EC1066 RegQueryInfoKey failed.</p> <p>EC1067 RegEnumValue failed.</p>	<p>EC2045 Couldn't write repository: [error].</p> <p>EC2046 Session::doCommand received name record before directory attributes.</p> <p>EC2047 Session::doCommand received name record before file attributes.</p> <p>EC2048 Unable to get size of emake stub command file [file]: [error].</p> <p>EC2049 Unable to read contents of emake stub command file [file]: [error].</p> <p>EC2050 Session::doCommand received E2A_SET_GOT_ALL_DIRECTORY_ENTRIES command for unknown directory.</p> <p>EC2051 E2A_LOAD_FILE_FROM_NODE received file data before attributes.</p> <p>EC2052 Error waiting for disk cache data in E2A_LOAD_FILE_FROM_NODE request.</p> <p>EC2053 Session::readFileData received file data before attributes.</p> <p>EC2054 Session::readFileData couldn't open cache file for writing.</p> <p>EC2055 pipeline error.</p> <p>EC2056 AgentExec::runCommand () was unable to contact the execserver: [error].</p> <p>EC2057 AgentExec::runCommand () was unable to get command results from the execserver: [error].</p> <p>EC2058 AgentExecImpl::startServer couldn't fork for execserver: [error].</p> <p>EC2059 AgentExecImpl::startServer couldn't ping execserver.</p> <p>EC2060 UnixBufferPool::allocateBuffer couldn't find free buffer or WinBufferPool::allocateBuffer couldn't find free buffer.</p> <p>EC2061 UnixBufferPool::flush couldn't write to {buffer [x]:[y]}+[]: [error].</p>	
--	---	--

<p>EC1068 Unable to open file.</p> <p>EC1069 Unable to write file.</p> <p>EC1070 Unable to create symlink.</p> <p>EC1071 Unable to modify file times of file.</p> <p>EC1072 Unable to modify attributes of file.</p> <p>EC1073 Job caused multiple agents to fail.</p> <p>EC1074 Mismatched object sizes.</p> <p>EC1075 No agents enabled on this cluster. You may want to contact your administrator.</p> <p>EC1076 No agents available, waiting...</p> <p>EC1077 [n] agent(s) allocated, proceeding...</p> <p>EC1078 ios_base::failure.</p> <p>EC1080 ios_base::failure.</p> <p>EC1082 Agent sent invalid content id.</p> <p>EC1083 Agent asked for file data on a non-file.</p> <p>EC1084 Agent sent invalid directory id.</p> <p>EC1085 Agent attempted to read a stale directory.</p> <p>EC1086 Agent attempted to read a stale directory.</p> <p>EC1087 Recursive make scheduled by wrong type of job.</p> <p>EC1088 Agent sent invalid name id.</p> <p>EC1089 Invalid new name id.</p> <p>EC1090 Attempted to create a file over an existing file.</p> <p>EC1091 Non-negative new name id.</p> <p>EC1092 Unable to determine path for [name] (OBSOLETE)</p> <p>EC1093 Unable to determine path for [name] (OBSOLETE)</p> <p>EC1094 Unable to determine path for [name] (OBSOLETE)</p>	<p>EC2062 Error mounting filesystem at "[mount point]": [error].</p> <p>EC2063 *** bad root "[root]".</p> <p>EC2064 Unable to attach pipe to file descriptor.</p> <p>EC2065 Error reading from pipe: [error].</p> <p>EC2066 Unable to create pipe.</p> <p>EC2067 Couldn't read cygwinMounts table: improper format.</p> <p>EC2068 GetVolumePathNameW ([path]): [error].</p> <p>EC2069 [drive]: cannot filter a volume mount point..</p> <p>EC2070 QueryDosDeviceW ([mountpoint]): [error].</p> <p>EC2071 Can't handle UNC paths.</p> <p>EC2072 Unhandled device mapping: [device].</p> <p>EC2073 Couldn't create path [drive] to hold root [mountpoint]: [err].</p> <p>EC2074 Can't create DOS drive for root "[mountpoint]": drive is already in use.</p> <p>EC2075 Error preparing mount point: [error].</p> <p>EC2076 setRootPaths([root], [mountpoint], [kernelpath], [volumedev]): [error].</p> <p>EC2077 Registry::setRoots was passed an illegal root handle: [handle].</p> <p>EC2078 Registry::setRoots couldn't create shadow registry key: [error].</p> <p>EC2079 Registry::remapKey unable to find root for [key] "[name]".</p> <p>EC2080 Registry::setKey unable to create key "[key]": [error].</p> <p>EC2081 Registry::setKey unable to set registry value "[name]": [error].</p> <p>EC2082 Agent has shut down or switched to a different CM.</p>	
--	--	--

<p>EC1095 Unable to find current working directory on agent.</p> <p>EC1096 Detected invalid job type while reloading parse data.</p> <p>EC1097 Detected invalid target type while reloading parse data.</p> <p>EC1099 Could not determine temp path on the node.</p> <p><a href="#">EC1100 Command was not A2E_AC, got [hex].</a></p> <p>EC1101 Bad agent request code: [hex].</p> <p>EC1102 Node not responding.</p> <p>EC1103 Command was not A2E_PUT_VERSION, got [hex].</p> <p>EC1104 Received A2E_PARSE_RESULT on wrong type of job.</p> <p>EC1105 Command was not A2E_PUT_TEMPDIR, got [hex].</p> <p>EC1106 Bad temp dir.</p> <p>EC1107 Command was not A2E_ACK, got [hex].</p> <p>EC1108 The protocol version of Electric Make ([ver]) is incompatible with the software installed on the Electric Cloud cluster [machine] ([version]).</p> <p>EC1109 The protocol version of Electric Make ([ver]) is a more recent version than the software installed on the Electric Cloud cluster [machine] ([version]).</p> <p>EC1110 The protocol version of Electric Make ([ver]) is incompatible with the software installed on the Electric Cloud cluster [machine] ([version]).</p> <p>EC1111 The protocol version of Electric Make ([ver]) is a more recent version than the software installed on the Electric Cloud cluster [machine] ([version]).</p> <p>EC1112 The protocol version of Electric Make ([ver]) is incompatible with the software installed on the Electric Cloud cluster [machine] ([version]).</p> <p>EC1113 The protocol version of</p>	<p>EC2083 session abort.</p> <p>EC2084 session abort wait.</p> <p>EC2085 session abort.</p> <p>EC2086 session abort wait.</p> <p>EC2087 any message.</p> <p>EC2088 ios_base::failure message talking to agent.</p> <p>EC2089 error from runCommand.</p> <p>EC2090 error from EFS.</p> <p>EC2091 Mount point "[mount point]" was not created by Mount::set.</p> <p>EC2092 E2A command sent to A2A server.</p> <p><a href="#">EC2093 Clock skew between emake and agent is %d ms.</a></p> <p>EC2094 Problem resolving cygwin mounts.</p> <p>EC2095 Error cleaning up cygwin mounts.</p> <p><a href="#">EC2096 Cygwin mount conflict.</a></p> <p>EC2097 Bad root "...": there is an unformatted disk occupying this drive letter on the node.</p> <p>EC2098 Couldn't read symlink cache table: improper format.</p> <p>EC2099 Couldn't create symlink root.</p> <p>EC2100 Couldn't set non-blocking mode on socket.</p> <p>EC2101 Requested nameId is already in use as contentId.</p> <p>EC2102 Requested contentId is already in use as a nameId.</p> <p>EC2103 Requested directoryId is a nameId.</p> <p>EC2104 Requested directoryId is a file contentId.</p> <p>EC2105 Requested contentId is a directoryId.</p> <p>EC2106 No visible content version for contentId.</p> <p>EC2107 Bad root "path": unable to stat "component/of/path".</p> <p>EC2108 Emake is 64-bit but agent is 32-bit.</p> <p>EC2109 Agent can't access path</p>	
--	--	--

<p>Electric Make ([ver]) is a more recent version than the software installed on the Electric Cloud cluster [machine] ([version]).</p> <p>EC1114 Emake-agent handshake failed, possibly from timeout.</p> <p><a href="#">EC1115 Emake-agent handshake failed (part 2), possibly from timeout.</a></p> <p>EC1116 User terminated build via debugger.</p> <p>EC1117 Directory::findContentForDot couldn't find parent for directory.</p> <p><a href="#">EC1118 pipeline error.</a></p> <p><a href="#">EC1119 pipeline error.</a></p> <p>EC1120 The command line option --emake-multi-remake is deprecated.</p> <p>EC1121 The command line option --emake-logcapture is deprecated.</p> <p>EC1122 Unable to determine path for [name] (OBSOLETE)</p> <p>EC1123 You did not preload cygwin1.dll in your top-level make. Use \--emake-cygwin=P there.</p> <p><a href="#">EC1124 Unable to rename file [path] to [path].</a></p> <p>EC1125 Unable to convert [option] "[path]": [error].</p> <p>EC1126 error from TCPsocket.</p> <p>EC1127 Error growing backing store: [error].</p> <p>EC1128 History file path references non-existent or.</p> <p>EC1129 History file is not writable.</p> <p><a href="#">EC1130 Build has fewer roots than history file: [history filename].</a></p> <p>EC1131 Cygwin path translation of "%s" failed.</p> <p>EC1132 Remote parse failed.</p> <p>EC1133 Invalid ledger option '...' in option string '...'.</p>	<p>due to insufficient permissions.</p> <p>EC2110 Agent expected directory but found file.</p> <p>EC2111 Error creating sandbox mountpoint "path".</p> <p>EC2112 Session::readFileData found incomplete file in diskcache.</p> <p>EC2113 Error in waitComplete in CacheWritePipelineStage::finalize.</p> <p>EC2114 Error in Session::doSetGotAllSubkeys.</p> <p>EC2115 Error in Session::doSetGotAllValues.</p> <p>EC2116 Don't have valid RegKeyInfo entry in Session::mRegKeyMap.</p> <p>EC2117 SessionEfsService::needKey didn't get key information from emake.</p> <p>EC2118 Registry::insertValues couldn't open registry key.</p> <p>EC2119 Session::doPutAllKeyVersions got invalid parent key id NNN.</p> <p>EC2120 Requested keyId is in use.</p> <p>EC2121 Requested parent keyId is in use.</p> <p>EC2122 Requested validId is in use.</p> <p>EC2123 Session::doPutAllValueVersions received valueId before keyId.</p> <p><a href="#">EC2128 Security Error: authentication rejected</a></p> <p><a href="#">EC2129 Security Error: authentication required</a></p> <p><a href="#">EC2130 Security Error: authentication not offered</a></p> <p><a href="#">EC2131 Security Error: already authenticated</a></p> <p><a href="#">EC2132 Security Warning: unavailable user XXX; instead setting user AAA and groups BBB,CCC</a></p> <p><a href="#">EC2133 Security Warning: unavailable primary group YYY; instead setting user AAA and</a></p>	
--	---	--



<p>EC1134 Couldn't open ledger file '...'.  EC1135 Couldn't commit ledger file '...'.  EC1136 Error accessing Ledger database in parent make.  EC1137 Unrecognized option "[option]" in EMAKEFLAGS.  EC1138 Couldn't load ClearCase.  EC1139 Error accessing ClearCase object.  EC1140 Invalid cross make OS "[os]".  EC1141 The priority 'high' is not supported.  EC1142 Can't create Ant context directory.  EC1143 Can't remove Ant context directory.  EC1144 Job XXX modified registry, now reverted.  EC1145 Empty value specified for --emake-reg-roots.  EC1146 Agent sent invalid registry key id.  EC1147 Agent sent invalid registry value id.  EC1148 Error retrieving subkey information for key NNN.  EC1149 Error retrieving value information for key NNN.  EC1150 Unable to invoke emake from stub.  EC1151 ParseJob XXX failed after N attempts.  EC1152 Unable to read 'path' for transfer to agent.  EC1153 Job recorded USAGE_EXISTS on a non-existent file.  EC1166 [signal]  EC1170 Fatal Security Error: ...  EC1171 [agent]: [agentPrincipal]: Security Error: ...  EC1172 Unknown security mode for agent</p>	<p>groups BBB,CCC  EC2134 Security Warning: unavailable supplementary groups YYY,ZZZ; instead setting user AAA and groups BBB,CCC  EC2135 Empty group list in E2A_SET_USER  EC2136 Unable to create file: "[pathname]": [system error]  EC2137 Unable to delete temporary file: "[pathname]": [system error]</p>	
--	--	--

<p>communications: "[option]"</p> <p>EC1173 ...</p> <p>EC1174 Invalid ClearCase options [invalid value]</p> <p>EC1175 Bad warning number: "[warning]"</p> <p>EC1176 Unable to delete temporary file: "[pathname]": [system error]</p> <p>EC1178 Couldn't stat local agent file for copy into backing store: "[pathname]": [system error]</p> <p>EC1179 Couldn't open local agent file for copy into backing store: "[pathname]": [system error]</p> <p>EC1182 Failed to raise signal [sig]; instead exiting with code [code]</p> <p>EC1183 #pragma multi is redundant on patterns.</p> <p>EC1184 #pragma multi not supported on patterns that immediately follow non-pattern targets. Stop.</p> <p>EC1185 #pragma multi not supported on static patterns. Stop.</p> <p>EC1186 #pragma multi not supported on double-colon rules. Stop.</p> <p>EC1187 target file `XYZ' has both #pragma multi rules and rules not affected by #pragma multi. Other rule at makefile:line. Stop.</p> <p>EC1188 target file `XYZ' cannot appear in #pragma multi rules with differing target sets. Other rule at makefile:line. Target `ABC' appears only in ... rule. Stop.</p> <p>EC1189 target file `XYZ' involves both #pragma multi and overridden commands. Other rule at makefile:line. Stop.</p> <p>EC1190 target file `XYZ' is #pragma multi but has no</p>		
--	--	--

explicit commands. Stop. EC1289 No valid eMake licenses...		
--	--	--

**Note:** Agent errors regarding the establishment of the virtual filesystem for a particular build will be displayed if there are at least three errors. These errors would occur during the initial setup of the agent's build-specific environment but before any particular build step is run on that agent. The most common type of error involves eMake roots or Cygwin mounts, where virtual filesystem setup is specific to the build but not to any particular build step.

## EC1001

### *Summary*

The 'exclusive' priority is deprecated. Using 'high' instead.

### *Description*

See above.

### *Reasons*

See above.

### *Fixes*

'High' is used in place of 'exclusive'.

## EC1002

### *Summary*

The command line option `--emake-annotate` is deprecated. Use the `--emake-annofile` and `--emake-annodetail` options instead.

### *Description*

See above.

### *Reasons*

See above.

### *Fixes*

Use the `--emake-annofile` and `--emake-annodetail` options instead.

## EC1003

### *Summary*

The directory `dir` specified in the error message was specified as an eMake root, but that location is covered by the directory `other dir`, so it is redundant.

### *Description*

During startup eMake performs a series of validation checks on the directories specified as eMake roots. As part of the validation process eMake checks whether any of the directories specified are subdirectories of any other directory specified as an eMake root. If any such redundant entries are found, eMake will display this warning and proceed as if the redundant directory had not been specified as an eMake root.

### Reasons

This message is displayed if the user specifies two directories as eMake roots such that one directory is contained within the other directory. For example, the user might specify both `/foo/bar/baz` and `/foo` as eMake roots, but `/foo/bar/baz` is contained within `/foo`.

### Fixes

This message is displayed as a warning only: the build will proceed as expected with no loss of performance or correctness. The warning can be avoided by removing the redundant path from the eMake root specification.

## EC1004

### Summary

eMake failed to create a temporary directory inside the directory specified in the error message. The text in the error detail provides more information about why the error occurred.

Example:

```
WARNING EC1004: Could not create temp dir in /vobs/cm_tools: Error in mkdir(/vobs/cm_tools/eccloud_tmp_8902_4): Permission denied
```

### Description

eMake creates several temporary directories at startup, used for storing build output files during the execution of the build. By default eMake will create one temporary directory per eMake root, so the files can later be moved to their final location using an efficient rename operation.

This warning indicates that eMake was unable to create one of these temporary directories. The error detail provides additional information about the cause of the failure, include the system error message if available. As long as eMake is able to successfully create at least one temporary directory it will proceed to run the build, although performance may be negatively impacted. If eMake is unable to create any temporary directories it will exit with error code EC1005.

### Reasons

This warning is produced as a result of an interacting with an external entity (the filesystem on the computer where eMake was invoked), so the cause of the failure could be just about anything. The error detail in the message should provide guidance as to the cause of the failure, such as `Permission denied`, indicating that the user the invoked eMake does not have write access to the directory.

### Fixes

The resolution for this issue depends on the specific cause of the warning. Possible remedial actions include explicitly changing the temporary directory location using the `--emake-tmpdir` command-line option, or if that option is already in use, specifying a different directory; or implicitly changing the temporary directory location by changing the eMake root setting.

## EC1005

### Summary

ERROR EC1005: No writable temporary directories are available.

Please specify at least one writable directory with the `--emake-tmpdir` option or in the `EMAKE_TMPDIR` environment variable.

### Description

As Electric Make (eMake) runs, files generated during the build are written to a temporary location and only moved to their correct final location when eMake determines that the job that produced a file did not have any conflicts. This enables the transactional nature of eMake. Therefore, in order to function correctly, eMake must have a writable location in which to create files. Error EC1005 indicates that no suitable location could be found.

### Reasons

Electric Make attempts to create the temporary directory in several locations, including any location explicitly specified by the user with the `--emake-tmpdir` command-line option or the `EMAKE_TMPDIR` environment variable; the current working directory; and the system temp directory. This error appears if all of these locations are not writable by the user running eMake.

### Fixes

Ensure that at least one of the locations enumerated above is writable by the user running eMake, or specify an alternate location using `--emake-tmpdir` or `EMAKE_TMPDIR`.

## EC1006

### Summary

eMake was invoked with an invalid value for the `--emake-cygwin` option.

### Description

The `--emake-cygwin` command-line option is a multistate option. When specified, the value must begin with Y, N, or A; or y, n, or a. Any other value produces this error.

<code>--emake-cygwin</code> option value	meaning
Y	require cygwin1.dll
N	ignore cygwin1.dll
A	load cygwin1.dll if available

### Reasons

An invalid value was specified for the `--emake-cygwin` command-line option.

### Fixes

Correct the value used for the `--emake-cygwin` command-line option, or remove `--emake-cygwin` from the command-line.

## EC1007

### Summary

EC1007: Unknown annotation detail token: "[token]"

### Description

You have defined an unknown annotation detail token.

### Reasons

See above.

### Fixes

Valid tokens:

basic  
env  
history

file  
lookup  
registry  
waiting

## EC1010

### Summary

EC1010: Unknown history mode: "[bad]"

Valid modes are: 'read', 'create', 'merge'

### Description

Electric Make (eMake) allows you to control the manner in which the history file will be used by the build.

- Read mode means that eMake reads the history file if present but not update it when the build finishes.
- Create mode means that eMake reads the history file if present and replaces it with a new history file containing only the discovered dependencies from the current build when the build finishes.
- Merge mode means that eMake reads the history file if present and add discovered dependencies from the current build when the build finishes, if those dependencies are not already present in the history file.

You may explicitly choose which of these modes to use with the `--emake-history` command-line option.

### Reasons

An invalid value is specified for the `--emake-history` command-line option.

### Fixes

Correct the value for the `--emake-history` command-line option. Valid values are read, create and merge.

## EC1015

### Summary

The directory specified in the error message was given as an eMake root, but the path could not be found or accessed; or it could be accessed but the path identifies a non-directory file.

### Description

At startup eMake attempts to validate each path specified as an eMake root to ensure that it exists, is accessible to the user invoking the build, and that it is a directory. Error code EC1015 indicates that one of these tests failed. Note that eMake will resolve symlinks and relative paths before attempting to locate the directory. For example, if the root is specified as `"/foo"`, where `"foo"` is a symlink to `"/another/location"`, eMake will verify the existence, accessibility and nature of `"/another/location"` rather than of `"/foo"`.

### Reasons

This error will be shown if the directory specified in the error message meets any of the following criteria:

- The path or any component of the path does not exist.
- The path or any component of the path is not accessible to the user invoking eMake.
- The path specifies a non-directory object, such as an ordinary file

### Fixes

- Verify that the path specified in the error message is actually needed for your build. If not, remove it from your eMake root specification.
- If the path is required for your build, verify that the path exists and is accessible by the user invoking the build. On Unix platforms, try "ls -ld path"; on Windows, try "dir path". If the path does not exist or is not accessible, create it or change the permissions to grant the user access.
- If the path exists and is accessible by the user invoking the build, verify that it is a directory. If it is not a directory but you expect it to be, consider deleting the path and creating a directory in its place; if it is a not a directory and you expect it not to be a directory, consider using the parent directory in your eMake root specification (eg, "/foo/bar" instead of "/foo/bar/file.txt").

### Notes for Isilon filers

DISCLAIMER: Consult Isilon support before making any changes to your Isilon filer configuration. Information related to Isilon filers is presented with no express or implied warranty. In no event shall Electric Cloud, Inc., be held liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this information.

Some customers have reported seeing ElectricAccelerator error EC1015 when a directory specified as an eMake root is hosted on an Isilon filer, if the filer is not configured to support 32-bit applications on 64-bit hosts, and if using the 32-bit eMake client on a 64-bit host. In this case, adding

```
vfs.nfsrv.do_32bit_fileid=1
```

to /etc/mcp/override/sysctl.conf on the Isilon server may resolve this issue.

## EC1016

### Summary

eMake displays this message when it has identified an eMake root specification as a UNC path.

### Description

eMake does not support UNC pathnames in eMake root values. Therefore, if the user specifies a UNC path as an eMake root (either with the `--emake-root` command-line option or the `EMAKE_ROOT` environment variable), eMake will display this error and exit. The offending path is given in the error message.

### Reasons

The user has specified a UNC path as an eMake root.

### Fixes

Remove the UNC path from the eMake root specification. If your build requires files from that UNC path, you can use the `subst` command to map the UNC path to a Windows drive letter, then specify that drive in the eMake root.

Note that some versions of eMake (5.2.x) incorrectly identify paths like "X:" as UNC paths (that is, a two-letter path consisting of a drive letter followed by a colon). Subsequent releases have corrected this error.

## EC1020

### Summary

eMake displays this error message if it is invoked in local mode, but the `--emake-annodetail` command-line option has been specified with an annotation detail level that is not supported in local mode.

### Description

When running a build locally, without the use of Accelerator agents, eMake does not have access to the data needed to enable various annotation detail levels, such as file or history details. In local mode, only basic, env and waiting annotation detail levels are permitted.

### Reasons

The user has specified `--emake-annodetail` with an unsupported annotation detail level when invoking eMake in local mode (that is, eMake is not using Accelerator agents). This error most commonly occurs when the user has forgotten to specify a cluster manager when invoking eMake, with either the `EMAKE_CM` environment variable or the `--emake-cm` command-line option.

### Fixes

If you intended to invoke eMake in local mode, you must remove the invalid detail level from the `--emake-annodetail` option. The error message indicates which annotation detail level caused the error.

If you did not intend to invoke eMake in local mode, ensure that you have specified a Cluster Manager for eMake to use, either by setting the `EMAKE_CM` environment variable, or by adding the `--emake-cm` option to your eMake command-line.

## EC1021

### Summary

eMake displays this error message if it is invoked in local mode, but the `--emake-annofile` command-line option is set.

### Description

When running a build locally, without the use of Accelerator agents, eMake is not able to write annotation to a named file. Instead, annotation is written to the standard output stream of the eMake process.

### Reasons

The user has specified `--emake-annofile` when invoking eMake in local mode (that is, eMake is not using Accelerator agents). This error most commonly occurs when the user has forgotten to specify a cluster manager when invoking eMake, with either the `EMAKE_CM` environment variable or the `--emake-cm` command-line option.

### Fixes

If you intended to invoke eMake in local mode, you must remove the `--emake-annofile` option from the command-line. If you wish to capture the annotation to a file, you can use the standard I/O redirection facilities on your platform to do so. For example, on Linux you can use

```
emake --emake-annodetail=basic > emake.xml
```

If you did not intend to invoke eMake in local mode, ensure that you have specified a cluster manager for eMake to use, either by setting the `EMAKE_CM` environment variable, or by adding the `--emake-cm` option to your eMake command-line.

## EC1030

### Summary

EC1030: Unable to open command file "filename": [OS error message]

Unable to open remote parse result file "filename": [OS error message]

### Description

During a cluster build, Electric Make (eMake) is invoked on the cluster hosts when a submake is invoked by the build, and when makefiles are parsed to determine the work that needs to be done in the build. In either case, the eMake process on the cluster hosts communicates with the Electric Agent process by means of a file on disk



containing the results of the cluster-side invocation. These error messages indicate that the cluster-side eMake process was unable to open that file. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

One common cause of this error on Unix systems is that the directory containing the file is not writable by the user running the build.

Remember that the path referenced in the error message is on a cluster host, not on the build host. You can reference the annotation file to determine the host where error occurred.

### **Fixes**

The fix depends on the cause of the error.

## **EC1031**

### **Summary**

EC1031: Unable to write command file "filename": [OS error message] during [operation]

Unable to write remote parse result file "filename": [OS error message]

### **Description**

During a cluster build, Electric Make (eMake) is invoked on the cluster hosts when a submake is invoked by the build, and when makefiles are parsed to determine the work that needs to be done in the build. In either case, the eMake process on the cluster hosts communicates with the Electric Agent process by means of a file on disk containing the results of the cluster-side invocation. These error messages indicate that the cluster-side eMake process was unable to write data to that file. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

Remember that the path referenced in the error message is on a cluster host, not on the build host. You can reference the annotation file to determine the host where error occurred.

### **Fixes**

The fix depends on the cause of the error.

## **EC1038**

### **Summary**

EC1038: Version mismatch, ignoring history file: [file]

### **Description**

As Electric Make (eMake) has evolved, so has the eMake history file format. Though Electric Cloud attempts to maintain backwards compatibility such that history files produced by older versions of eMake can be used with newer versions of eMake, occasionally incompatible changes are required. To avoid attempts to use incompatible history files, the history file contains a version number, which eMake validates when it first reads the file. If the history version is not one that eMake can support, the history is considered invalid.

### **Reasons**

This message is displayed if you attempt to run a build with an incompatible history file. This is most likely to occur when you have recently upgraded to a new version of eMake.

### **Fixes**

Verify that you are using the correct history file. Elaborate history file management systems introduce the possibility of accidentally using the wrong history file for the build.

If you verified you are using the correct history file, generate a new history file for the build using the new version of eMake. This is the surest way to get a good history file again.

If generating a new history file is not practical, it may be possible to convert the old history file to a newer format. Contact Electric Cloud technical support for assistance if this type of conversion is required.

## **EC1039**

### **Summary**

WARNING EC1039: Ignoring malformed history file: filepath

### **Description**

This warning indicates that the named history file is invalid. The build proceeds as if no history file is available, so the build may run longer than expected due to conflicts, but correctness is unaffected.

### **Reasons**

The Electric Make (eMake) history file consists of several plain-text sections. Each section begins with a header naming the section and ends with a single blank line. The sections must occur in the correct order: makes, jobs, serial-deps, prereqs, and prereq-deps. Message EC1039 specifically indicates that the makes section header was missing or did not match the literal text "makes:".

### **Fixes**

The best way to eliminate this warning is to generate a new history file by running the build to completion. eMake automatically generates a new, properly formatted history file when the build finishes as long as the `--emake-history` mode is either merge (the default if nothing is specified) or create.

If the `--emake-history` mode is read, then eMake does not replace the history file automatically. In that case you must rerun the build with `--emake-history` set to merge or create.

You may also eliminate this warning by deleting the invalid history file, but you must still generate a new history file for the build in this case.

## **EC1040**

### **Summary**

WARNING EC1040: Ignoring malformed history file: filepath

### **Description**

This warning indicates that the named history file is invalid. The build proceeds as if no history file is available, so the build may run longer than expected due to conflicts, but correctness is unaffected.

### **Reasons**

The Electric Make (eMake) history file consists of several plain-text sections. Each section begins with a header naming the section and ends with a single blank line. The sections must occur in the correct order: makes, jobs, serial-deps, prereqs, and prereq-deps. Message EC1040 specifically indicates that the jobs section header was missing or did not match the literal text "jobs:".

### **Fixes**

The best way to eliminate this warning is to generate a new history file by running the build to completion. eMake automatically generates a new, properly formatted history file when the build finishes as long as the `--emake-history` mode is either merge (the default if nothing is specified) or create.

If the `--emake-history` mode is read, then eMake does not replace the history file automatically. In that case you must rerun the build with `--emake-history` set to merge or create.

You may also eliminate this warning by deleting the invalid history file, but you must still generate a new history file for the build in this case.

## EC1042

### Summary

WARNING EC1042: Ignoring malformed history file: filepath

### Description

This warning indicates that the named history file is invalid. The build proceeds as if no history file is available, so the build may run longer than expected due to conflicts, but correctness is unaffected.

### Reasons

The Electric Make (eMake) history file consists of several plain-text sections. Each section begins with a header naming the section and ends with a single blank line. The sections must occur in the correct order: makes, jobs, serial-deps, prereqs, and prereq-deps. Message EC1042 specifically indicates that the serial-deps section header was missing or did not match the literal text "serial-deps:".

### Fixes

The best way to eliminate this warning is to generate a new history file by running the build to completion. eMake automatically generates a new, properly formatted history file when the build finishes as long as the `--emake-history` mode is either merge (the default if nothing is specified) or create.

If the `--emake-history` mode is read, then eMake does not replace the history file automatically. In that case you must rerun the build with `--emake-history` set to merge or create.

You may also eliminate this warning by deleting the invalid history file, but you must still generate a new history file for the build in this case.

## EC1043

### Summary

WARNING EC1043: Ignoring malformed history file: filepath

### Description

This warning indicates that the named history file is invalid. The build proceeds as if no history file is available, so the build may run longer than expected due to conflicts, but correctness is unaffected.

### Reasons

The Electric Make (eMake) history file consists of several plain-text sections. One of these is the serial-deps section, which enumerates the discovered dependencies in the build using a simple encoding consisting of lists of indexes into the makes and jobs sections. Message EC1043 indicates that there was an error processing the serial-deps section. Generally this means that the history file is corrupt or truncated.

### Fixes

The best way to eliminate this warning is to generate a new history file by running the build to completion. eMake automatically generates a new, properly formatted history file when the build finishes as long as the `--emake-history` mode is either merge (the default if nothing is specified) or create.

If the `--emake-history` mode is read, then eMake does not replace the history file automatically. In that case you must rerun the build with `--emake-history` set to merge or create.

You may also eliminate this warning by deleting the invalid history file, but you will still have to generate a new history file for the build in this case.

## EC1045

### Summary

WARNING EC1045: Ignoring malformed history file: filepath

### Description

This warning indicates that the named history file is invalid. The build proceeds as if no history file is available, so the build may run longer than expected due to conflicts, but correctness is unaffected.

### Reasons

The Electric Make (eMake) history file consists of several plain-text sections. One of these is the prereq-deps section, which enumerates any implicit prerequisites for jobs in the build that were discovered by eDepend. Message EC1045 indicates that there was an error processing the prereq-deps section. Generally this means that the history file is corrupt or truncated.

### Fixes

The best way to eliminate this warning is to generate a new history file by running the build to completion. eMake automatically generates a new, properly formatted history file when the build finishes as long as the `--emake-history` mode is either `merge` (the default if nothing is specified) or `create`.

If the `--emake-history` mode is read, then eMake does not replace the history file automatically. In that case you must rerun the build with `--emake-history` set to `merge` or `create`.

You may also eliminate this warning by deleting the invalid history file, but you must still generate a new history file for the build in this case.

## EC1049

### Summary

[unexpected exception halting build]

Example:

ERROR EC1049: Unable to start new thread: Cannot allocate memory

### Description

An unexpected exception occurred. The result of receiving this error message is a failed build.

### Reasons

There are many possible causes for this error. Refer to the specific OS message to help determine the cause of the error.

### Fixes

Refer to the specific OS message to help determine the cause of the error.

Using this example: ERROR EC1049: Unable to start new thread: Cannot allocate memory, the fix is to reduce the number of maximum agents used for the build. This was done by reconfiguring the build class's maximum number of agents per build or by adding `--emake-maxagents=[max.number]` to your command line.

## EC1050

### Summary

[agent]: [error]; aborting build

Examples:

ERROR EC1050: nnnagent001:

agent reports: [bad root "W:/": the volume information for the disk occupying this drive letter on the host was unavailable: The system cannot find the file specified (error code 2)]; aborting build

ERROR EC1050: nnnagent088:

The agent protocol version of Electric Make (20) is a more recent version than the software installed on the Electric Cloud agents in cluster cls999 (18).

Please contact your Electric Cloud administrator for assistance; aborting build

### **Description**

An error occurred on the specified agent, which aborted the build.

### **Reasons**

There are many possible causes for this error. Refer to the specific OS message to help determine the cause of the error.

### **Fixes**

Refer to the specific OS message to help determine the cause of the error.

### **Custom error text for protocol mismatches**

You can insert custom error text that is displayed when there is a protocol mismatch. In `<ECloud install>/<arch>/conf/accelerator.properties`, add your custom text to the `ACCELERATOR_CLIENT_MISMATCH_DETAIL` property.

## **EC1051**

### **Summary**

EC1051: Unable to link newpath to oldpath

### **Description**

In the process of committing build outputs to disk (here, creating a hard link to an existing file), Electric Make (eMake) was unable to create a hard link.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on.

### **Fixes**

The fix depends on the cause of the error.

## **EC1052**

### **Summary**

EC1052: Unable to rename directory oldpath to newpath: [OS error message]

### **Description**

In the process of committing build outputs to disk (here, renaming a directory from its original name to a new name), Electric Make (eMake) was unable to rename a directory. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### **Fixes**

The fix depends on the cause of the error.

## **EC1053**

### **Summary**

EC1053: Unable to read input stream

### **Description**

In the process of committing build outputs to disk (here, copying data from a temporary backing store location to its final location on disk), Electric Make (eMake) was unable to read from a backing store file.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. However, this error is very unlikely to occur in practice because the file being read was created by eMake itself earlier in the build.

### **Fixes**

The fix depends on the cause of the error.

## **EC1054**

### **Summary**

EC1054: Unable to determine current working directory: [OS error message]

### **Description**

At startup, Electric Make was not able to determine the path to the directory where it was invoked. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### **Fixes**

The fix depends on the specific cause of the error.

## **EC1056**

### **Summary**

Can't read directory [path]: [OS error message]

### **Description**

Electric Make encountered an error while trying to get the contents of the directory identified in the error message. The specific cause of the failure is given by the OS error message.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### **Fixes**

The fix depends on the cause of the error.

## **EC1057**

### **Summary**

EC1057: Unable to read ACL for directory path (ACL error): filesystem error.

### **Description**

Access Control List (ACL) is how Windows handles file permissions.

A failure to read the ACL is not fatal, but may lead to issues with the permissions of the files created.

If a failed Windows API call caused the attempt to read ACL to fail, the error message gives the API call and failure code.

### **Reasons**

When eMake receives a file from a cluster host, it places that file in a temporary directory (when possible, on the same device as the file belongs), and then moves the file into place at commit time. This can lead to problems on Windows because the file, when created, inherits the ACL specified by its parent directory, and simply moving it into place leaves it with the ACL from the temporary directory instead of the ACL it would have had if it were created in place.

To avoid this problem, eMake caches the ACL of each directory it works with, calculates the appropriate ACL for anything created in that directory, and applies the appropriate ACL to files in the temporary directory.

### **Fixes**

The `cacls` command will display and edit ACLs from the Windows command line.

Example

```
WARNING EC1057: Unable to read ACL for directory
m:/dist/fsf/PROJ/boost/1.33.1/common/include/boost/variant
(m:/dist/fsf/PROJ/boost/1.33.1/common/include/boost/variant: LookupAccountSid: The trust relationship between
this workstation and the primary domain failed. (1789)): The operation completed successfully
Looking up 1789 in WinError.h, that is ERROR_TRUSTED_RELATIONSHIP_FAILURE.

The first thing to do in this case is to run cacls m:/dist/fsf/PROJ/boost/1.33.1/common/include/boost/variant to see
what Windows sees. Normally, if LookupAccountSid() returns ERROR_NONE_MAPPED ("No mapping
between account names and security IDs was done. (1332)"), we just give the name (not found) rather than
erroring out.
```

## **EC1058**

### **Summary**

ERROR EC1058: Unable to read backing store file [path]: The system cannot find the file specified

### **Description**

EC1058 means that the eMake backing store file changed from underneath it. This means that a file that eMake believed should have been in its temporary directory was either moved, renamed, or its permissions modified during the build such that it became inaccessible.

### **Reasons**

This can happen if you have another process running in the background on the same system. From previous cases, common culprits are antivirus, build clean up scripts, and so on.

Suggestions for resolution:

- If you are aware of other processes that may be attempting to move or modify files in the eMake backing store, turn them off (or configure them to avoid changing tmpdir files during a build)
- If not, and the problem persists, keep a Process Monitor trace running on the eMake machine at the same time as the build - it will show us definitively what is happening to the files in the backing store. You can get Process Monitor from <http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>

## EC1060

### *Summary*

eMake displays this message if an error occurs trying to write the contents of a file to disk.

### *Description*

In the process of writing the contents of a file created during the build to disk, eMake was unable to open the file identified by path in the message. The OS error message provides the specific cause of the failure. Note that the path will be a location inside a temporary directory created by eMake to store files created during the build between the time of creation and the time that the files are committed to their final location.

### *Reasons*

Because the error involves an external system in this case (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### *Fixes*

The fix will depend on the specific cause of the error. For example, if the error states no space left on device, then you need to ensure that the device hosting the eMake temporary directory has sufficient space to store the files created during the build. You may also be able to use the `--emake-tmpdir` command-line option to explicitly set the eMake temporary directory location to another filesystem.

## EC1061

### *Summary*

eMake was not able to set the timestamps on the file specified in the warning message.

### *Description*

In the process of committing build results to disk (here, explicitly setting file times because of a touch command or similar during the build), Electric Make (eMake) was unable to change the timestamps on the named file. The OS error message provides the specific cause of the failure.

### *Reasons*

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### *Fixes*

The fix depends on the cause of the error.

## EC1063

### *Summary*

ERROR EC1063: Unable to create directory dirpath: [OS error message]



### **Description**

In the process of committing build outputs to disk (here, creating a directory that was created during the build), Electric Make (eMake) was unable to create the specified directory. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

One possible cause for this error is that another process has modified the build output area while eMake is running. This can lead to unpredictable results and should be avoided if possible.

### **Fixes**

The fix depends on the cause of the error.

## **EC1064**

### **Summary**

WARNING EC1064: Unable to modify the file times of dirpath: [OS error message]

### **Description**

In the process of committing build results to disk (here, explicitly setting the timestamps of a directory created during the build), Electric Make (eMake) was unable to modify the file times. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

One possible cause for this error is that another process has modified the build output area while eMake is running. This can lead to unpredictable results and should be avoided if possible.

### **Fixes**

The fix depends on the cause of the error.

## **EC1065**

### **Summary**

WARNING EC1065: Unable to modify attributes of dirpath: [OS error message]

### **Description**

In the process of committing build results to disk (here, updating attributes such as permissions of a directory created during the build), Electric Make (eMake) was unable to modify the attributes of the named directory. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

One possible cause for this error is that another process has modified the build output area while eMake is running. This can lead to unpredictable results and should be avoided if possible.

### **Fixes**

The fix depends on the cause of the error.

## **EC1068**

### **Summary**

EC1068: Unable to open file [path]: [OS error message]

### **Description**

In the process of committing build results to disk, Electric Make (eMake) was unable to open the specified file. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### **Fixes**

The fix depends on the cause of the error.

## **EC1069**

### **Summary**

EC1069: Unable to write file [path]: [OS error message]

### **Description**

In the process of committing build outputs to disk, Electric Make (eMake) was unable to write to the named file. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### **Fixes**

The fix depends on the cause of the error.

## **EC1070**

### **Summary**

EC1070: Unable to create symlink [path]: [OS error message]

### **Description**

In the process of committing build outputs to disk, Electric Make (eMake) was unable to create the named symlink. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

This error can also occur due to ClearCase ROFS. See KBEA-00068 - Overcoming problems with MVFS read-only file systems.

### **Fixes**

The fix depends on the cause of the error.

## **EC1071**

### **Summary**

EC1071: Unable to modify file times of [path]: [OS error message]

### **Description**

In the process of committing build results to disk (here, explicitly setting file times because of a touch command or similar during the build), Electric Make (eMake) was unable to change the timestamps on the named file. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### **Fixes**

The fix depends on the cause of the error.

## **EC1072**

### **Summary**

EC1072: Unable to modify attributes of [path]: [OS error message].

### **Description**

In the process of committing build results to disk (here, updating file permissions or ownership as a result of a chmod or chown operation or similar during the build), Electric Make (eMake) was unable to make the necessary changes to the named file. The OS error message provides the specific cause of the failure.

### **Reasons**

Because the error involves an external system in this context (the filesystem on the eMake host), the list of possible causes for this error is virtually limitless: a corrupted filesystem; hardware failure; insufficient permissions; and so on. Refer to the specific OS error message to determine the cause of the failure.

### **Fixes**

The fix depends on the cause of the error.

## **EC1073**

### **Summary**

Job caused multiple agents to fail.

### **Description**

This error means that a job failed on 3 separate agents and the job caused the agents to become non-responsive. When this occurs, Electric Make fails the job and aborts the build. The build proceeds only if you have set -k, -i, or other options that allow the build to continue after an error.

This ElectricAccelerator behavior is designed to prevent a single command from bringing down an entire cluster.

### **Reasons**

This message could be displayed if any of the following occurs:

- someone restarts agents without disabling them first
- an agent/host loses network connectivity
- any other non-agent connectivity issue

For possible hints about why you received this message, examine the Messages tab on the Cluster Manager UI for the affected Agents when running that build.

### **Fixes**

Always disable agents before restarting them. Resolve any connectivity issues.

## **EC1075**

### **Summary**

eMake displays this message when there are no agents enabled on the cluster.

### **Description**

This message indicates that eMake failed to acquire agents from the Cluster Manager, because there are no agents that are both enabled and good (shown with a green checkmark in the "Status" column of the Agents tab in the Cluster Manager GUI).

### **Reasons**

This message means that there are not any enabled and good agents on the Cluster Manager.

Note that if you have enabled a Resource Manager in the Cluster Manager, and you have specified a resource for this build, then this message means that there are not any enabled, good agents matching the resource specification. There may be enabled, good agents on the Cluster Manager, but not part of the resource specified for this build.

### **Fixes**

Check that the Cluster Manager has enabled, good agents. If there are agents enabled but they are not good, check the Cluster Manager Messages log for information about why the agents have been marked bad.

If you are using resources, you should also make sure that you have specified the resource correctly. If the specification is correct, check the resource definition in the Cluster Manager (if you are using the built-in Resource Manager) to ensure it includes all the agents you intend it to include. If you are using an external Resource Manager such as LSF, check that the external resource manager is configured correctly for the resource you have specified.

## **EC1076**

### **Summary**

eMake displays this message when it cannot acquire any agents from the Cluster Manager.

### **Description**

This message indicates that eMake failed to acquire agents from the Cluster Manager, so eMake is unable to continue executing the build. eMake will wait in this state until agents are made available to it. This warning does not indicate a build failure, but it does mean that the build will take longer to complete than normally expected.

### **Reasons**

Fundamentally this message means that there are not enough enabled agents to service all currently running builds. Other builds are using all the available agents, leaving none for this build.

Note that if you have enabled a Resource Manager in the Cluster Manager, and you have specified a resource for this build, then this message means that there are not enough enabled agents matching the resource

specification. There may be unused, enabled agents on the Cluster Manager, but not part of the resource specified for this build.

### **Fixes**

Check that the Cluster Manager has enough agents available to service the currently running builds. If not, you can add agents to the cluster, or use max agent and build priority settings to limit the number of agents any single build can use, in order to make room for other concurrent builds.

If you are using resources, you should also make sure that you have specified the resource correctly. If the specification is correct, check the resource definition in the Cluster Manager (if you are using the built-in Resource Manager) to ensure it includes all the agents you intend it to include. If you are using an external Resource Manager such as LSF, check that the external resource manager is configured correctly for the resource you have specified.

## **EC1077**

### **Summary**

eMake displays this message when it is at last allocated agents and able to continue executing the build, after being stalled due to a shortage of agents.

### **Description**

This message that eMake previously failed to acquire agents from the Cluster Manager, but now has been assigned [count] agents. This message will have been preceded by warning EC1076.

### **Reasons**

Fundamentally this message means that there are not enough enabled agents to service all currently running builds. Other builds were using all the available agents, leaving none for this build.

Note that if you have enabled a Resource Manager in the Cluster Manager, and you have specified a resource for this build, then this message means that there were not enough enabled agents matching the resource specification. There may have been unused, enabled agents on the Cluster Manager, but not part of the resource specified for this build.

### **Fixes**

Check that the Cluster Manager has enough agents available to service the currently running builds. If not, you can add agents to the cluster, or use max agent and build priority settings to limit the number of agents any single build can use, in order to make room for other concurrent builds.

If you are using resources, you should also make sure that you have specified the resource correctly. If the specification is correct, check the resource definition in the Cluster Manager (if you are using the built-in Resource Manager) to ensure it includes all the agents you intend it to include. If you are using an external Resource Manager such as LSF, check that the external resource manager is configured correctly for the resource you have specified.

## **EC1100**

### **Summary**

Electric Make (eMake) generates this error when it fails to receive an A2E\_ACK message from an agent during the initial connection handshake with that agent.

### **Description**

The handshake consists of several messages, with a few E2A\_SYNC/A2E\_ACK pairs at specific points during the sequence. One such point is following the E2A\_SET\_TIME message. eMake sends the sequence E2A\_SET\_TIME, E2A\_SYNC; and it expects the agent to respond with A2E\_ACK. This error code indicates that the agent replied with some other command code.

### **Reasons**

There is no known reason this error code would appear.

### **Fixes**

If you see this error occur reproducibly, an eMake jn debug log and an agent "session trace all" log will be instrumental in diagnosing the problem.

## **EC1115**

### **Summary**

eMake-agent handshake failed (part 2), possibly from timeout

### **Description**

When an agent is assigned to a build, the local eMake keeps in contact with the agent to make certain that the job is still running. If the agent (or the network) is very busy, it is not unusual for the handshake exchange (which is basically a heartbeat) to fail. eMake will resend the handshake message. If the agent stops responding for an extended period of time, eMake will abort the job. If the jobs are not aborting, then you can ignore such messages.

### **Reasons**

See above.

## **EC1118**

### **Summary**

An I/O error occurred when reading file data from an agent and saving it to disk. Examples:

ERROR EC1118: error writing [file]: [OS error message]

ERROR EC1118: unexpected end of file reading file data from TCP connection from agent

ERROR EC1118: error reading TCP connection from agent: [exception detail], [OS error message]

### **Description**

An error of some kind has occurred when trying to read and record file data from an agent. The error message contains additional details to help diagnose the problem. For example, if the error occurred during the write-to-disk portion of the operation, the message states "error writing ...", specifying the name of the backing store file eMake selected to receive the data and the OS error message for the failed write. If the error occurred during the read-from-network portion of the operation, the message states "error reading ...", specifying the type of exception that occurred and the OS error message for the failed read.

### **Reasons**

The list of possible causes for this error is virtually limitless: out of disk space on the eMake client; a failed network connection to the agent; hardware failure of some sort; and so on. Read the OS error message to determine the specific cause of the failure.

### **Fixes**

This is truly an example of an "exceptional" error - one that in practice, should not be expected to occur at all. Most likely if it happens once it will not happen again, so a retry may be all that is in order. If the messages are occurring persistently, refer to the specific error message from the OS and reason about what could cause that type of failure. Searching for the OS error message using Google may turn up some clues.

## **EC1119**

### **Summary**

An I/O error occurred when trying to read data from disk and send it to an agent. Examples:

ERROR EC1119: error reading [file]: [OS error message]

ERROR EC1119: error writing TCP connection to agent: [exception detail], [OS error message]

### **Description**

An error of some kind has occurred when trying to send file data to an agent. The error message contains additional details to help diagnose the problem. For example, if the error occurred during the read-from-disk portion of the operation, the message states "error reading ...", specifying the name of the file eMake was reading from and the OS error message for the failed read. If the error occurred during the write-to-network portion of the operation, the message states "error writing ...", specifying the type of exception that occurred and the OS error message for the failed write.

### **Reasons**

The list of possible causes for this error is virtually limitless: hardware failure; network connectivity problems; and so on. Read the OS error message to determine the specific cause of the failure.

### **Fixes**

This is truly an example of an "exceptional" error - one that in practice, should not be expected to occur at all. Most likely if it happens once, it will not happen again, so a retry may be all that is in order. If the messages are occurring persistently, refer to the specific error message from the OS and reason about what could cause that type of failure. Searching for the OS error message using Google may turn up some clues.

## **EC1124**

### **Summary**

eMake displays this error when there is a problem processing a rename operation encountered during the build.

### **Description**

When eMake executes a build, it executes jobs in a virtualized filesystem context, rather than directly against the real filesystem. This gives eMake the flexibility to revert jobs if necessary due to conflicts, and also eliminates the risk that simultaneously running jobs will interfere with each other. Only after a job has been verified to be free of conflicts will eMake apply the filesystem operations recorded in the virtualized filesystem to the real filesystem. Error EC1124 indicates that an error was encountered when apply a rename operation to the real filesystem. The source and target of the rename operation are given in the error message.

### **Reasons**

Because eMake is interacting with an external entity—here, the filesystem on the host from which eMake was invoked—the list of possible causes for this error is virtually limitless: out of disk space on the eMake client; a failed network connection to the agent; hardware failure of some sort; and so on. Read the OS error message to determine the specific cause of the failure.

One possible cause for this error is that another process has modified the build output area while eMake is running. This can lead to unpredictable results and should be avoided if possible.

### **Fixes**

The fix depends on the specific cause of the error.

## **EC1130**

### **Summary**

eMake was invoked with an eMake root specification containing fewer entries than were used when the history file was generated.

### **Description**

The eMake history file records information about missing dependencies discovered while executing a build, for the purpose of improving performance in subsequent builds. The information is recorded as a series of eMake root relative paths, which enables the user to reuse the history file as long as the structure of the filesystem within the eMake roots is consistent from one build to the next, even if the absolute location of the build tree changes. Because the build may use multiple roots, the paths in the history file are specified as relative to a particular root identified by number; naturally if the number of roots used for the build changes, then the numeric identifier for each root may change, which will render eMake unable to match the information in the history file to the currently running build. Therefore, if the history file was generated by a build using a root specification with more entries than the current build has specified, eMake will display this warning.

Note that this message is a warning only, not an error. eMake will proceed to execute the build as expected. Performance may be impacted, but correctness will not be affected.

### **Reasons**

The user invoked eMake with an eMake root specification that contains fewer entries than the specification that was used when the history file was generated.

### **Fixes**

Ensure that eMake is invoked with an eMake root specification that matches the one used when the history file was generated.

## **EC1137**

### **Summary**

This warning indicates the specified EMAKEFLAGS option is unrecognized.

### **Description**

See above.

### **Reasons**

This warning may be displayed if EMAKEFLAGS is defined with options that contain spaces in the value (for example, `--emake-option="value with spaces"`). Depending on how you use EMAKEFLAGS, this may result in builds not operating correctly. Specifically, if you specified `--emake-root` or `--emake-resource` through EMAKEFLAGS, and those options have spaces, your build may not use the correct root or the correct resource.

### **Fixes**

For the issue mentioned above, use the EMAKE\_ROOT environment variable to specify the eMake root, and eliminate spaces in your resource specification.

## **EC1138**

### **Summary**

The user invoked eMake with the `--emake-clearcase` option specified, but eMake was unable to load the ClearCase integration module.

### **Description**

eMake will display error EC1138 if the user specifies a value for the `--emake-clearcase` command-line option, but eMake fails to load the ClearCase integration module for any reason. The error detail in the error message will provide additional information about the cause of the error.



### Reasons

Because eMake is interacting with an external system (ClearCase), the cause of the failure could be just about anything. Look to the error detail in the error message for the specific cause of the failure.

For example, one cause of this error is that ClearCase is not installed or is improperly installed on the host used to invoke eMake:

ERROR EC1138: Unable to load ClearCase: Couldn't open ecclearcase6.so: libatriabuild.so: cannot open shared object file: No such file or directory

Couldn't open ecclearcase7.so: libatriabuild.so: cannot open shared object file: No such file or directory

### Fixes

The resolution to this issue will depend on the specific cause of the error.

## EC1140

### Summary

Invalid cross make OS "[os]": must be "linux" or "solaris"

### Description

ElectricAccelerator 4.2.0 added support for a limited form of "cross make": that is, running the Electric Make client on a host running a different operating system than that used on the cluster hosts. This feature is supported only from Linux to Solaris or from Solaris to Linux.

### Reasons

The user specified an invalid value for the `--emake-crossmake` command-line option.

### Fixes

Correct the value specified for the `--emake-crossmake` command-line option. The value must be either "linux", indicating that the user wishes to use Linux-based agents for the build; or "solaris", indicating that the user wishes to use Solaris-based agents for the build.

## EC1144

### Summary

Electric Make (eMake) emits this warning in v4.2.x when it reverts a job that made registry modifications (if `--emake-pedantic` mode is enabled).

### Description

Prior to ElectricAccelerator v4.3.0, eMake did not version registry operations the way it does filesystem operations. One consequence of this shortcoming is that when jobs that made registry modifications are reverted (for example, due to a conflict or an upstream error), the modifications made to the registry are not reverted. This can lead to inconsistencies between the registry and the filesystem, for example, if the job that was reverted was registering a TLB location; in turn, this can cause future builds to fail unexpectedly.

### Reasons

See above.

### Fixes

Prior to ElectricAccelerator v4.3.0, the best way to avoid this problem is to make sure you always have a good history when running multi-agent builds. By avoiding job reverts, you should avoid this problem entirely.

## EC1145

### Summary

Electric Make emits this warning anytime the user specifies an empty value for the `--emake-reg-roots` option.

To disable registry mirroring, specify `--emake-reg-roots=-HKEY_CLASSES_ROOT`.

### Description

Prior to Accelerator 4.3.0, the interpretation of the `--emake-reg-roots` command-line option was that it would replace the built-in default registry root setting. A side-effect of this implementation was that the user could completely disable registry mirroring by specifying `--emake-reg-roots=""` (that is, an empty string value).

However, the override behavior made it easy for users to unintentionally disable registry mirroring for the default locations when all they wanted was to also mirror some set of keys. This in turn made it possible for users to inadvertently "corrupt" the registry on the cluster by allowing modifications to bleed through to the real registry there.

Therefore, in 4.3.0 the interpretation of the `--emake-reg-roots` command-line option was changed such that values specified that way are used to augment the built-in default setting, rather than replace it. Therefore, using an empty string to disable registry mirroring no longer works as it did prior to 4.3.0.

### Reasons

An empty value was specified for `--emake-reg-roots`.

### Fixes

Do not use an empty value for the `--emake-reg-roots` option. If you want to disable registry mirroring, you must explicitly "uninclude" the portion of the registry that is mirrored by default (HKEY\_CLASSES\_ROOT). You can do so by specifying that path as an exclude value for the registry root:

```
--emake-reg-roots=-HKEY_CLASSES_ROOT
```

## EC1151

### Summary

Despite multiple retries, eMake was unable to successfully execute the parse job indicated in the error message.

### Description

This error indicates that eMake has crashed with an access violation while trying to parse a makefile during a build. Because this can be caused by defects in system libraries, eMake will retry any parse jobs that fail in this way, up to three times. If none of these attempts is successful, eMake will display error EC1151.

### Reasons

This error may be caused by an improperly configured cluster host; or by a defect in eMake.

### Fixes

Take note of the cluster host on which the crash occurred. If the same host consistently produces this failure, consider reimaging that host. If that does not resolve the issue, contact Electric Cloud support at [support@electric-cloud.com](mailto:support@electric-cloud.com). Be prepared to provide the following information:

- An eMake debug log generated with `--emake-debug=jn`.
- An eMake remote parse debug log generated with `--emake-rdebug=jp`.
- The crash dump or core file generated by eMake at the time of the failure. Note that this may be found on

the cluster host in the bin directory of the ElectricAccelerator installation.

- The eMake version number, from `--version`.

## EC1152

### *Summary*

eMake was not able to open the file specified in the error message for reading.

### *Description*

While executing a build, eMake was asked to send the contents of the specified file to a cluster agent, but eMake was not able to read the file. The system error detail will provide more information about the specific cause of the failure.

### *Reasons*

Because eMake is interfacing with an external entity (the filesystem on the computer where eMake was invoked), the cause of the error could be just about anything. The system error detail will provide more information about the specific cause of the failure. However, this error generally means either the filesystem has been modified by a third party while eMake was running, or that there is a defect in ElectricAccelerator.

### *Fixes*

First verify that no process besides eMake is modifying the files used during the build, which can create the inconsistent state that could provoke this error message. If you are sure that no other process has modified the files while eMake is running, contact [support@electric-cloud.com](mailto:support@electric-cloud.com) for assistance.

## EC1166

### *Summary*

WARNING EC1166: [signal]

### *Description*

eMake received the terminal signal described by the text of the message. For example, the user may have typed Ctrl-C. Of course, this message is NOT printed for signals that cannot be intercepted, such as Unix/Linux SIGKILL.

After printing this message, eMake will attempt to perform a controlled shutdown.

**Note:** This message will be printed only the first time a terminal signal is received. Subsequent terminal signals do not hasten the controlled shutdown, nor do they trigger this message.

Also note, this message is issued in cluster mode only. In local mode it would race with other messages, causing interleaving with other build output.

### *Reasons*

The user or some other agency has sent a terminal signal to the eMake process.

### *Fixes*

If the user did not actually desire to kill their build, then they need to track down who or what sent the signal. The signal description may provide a clue.

## EC1170

### *Summary*

An unrecoverable security error; after issuing this message, eMake will fail an assertion.

### **Description**

This internal failure should not happen, and eMake will fail an assertion in order to avoid possible exploitation of the unexpected state by an attacker, and also to aid in debugging. Please report the full text of this error to ElectricAccelerator engineering.

### **Reasons**

The reasons will have to be researched if and when this error occurs.

### **Fixes**

Please report the full text of this error to ElectricAccelerator engineering, along with any other relevant data concerning how it arose. Any fixes or workarounds will need to be researched.

## **EC1171**

### **Summary**

eMake failed in its preparations to authenticate itself to the agent, or the agent failed mutual authentication. (This error is detected on the eMake side.)

### **Description**

eMake tried to authenticate itself to a particular agent, but failed to do so, or the agent failed to authenticate itself back to the eMake client. The error message text gives further details, including the agent principal and Web UI name.

### **Reasons**

There are any number of reasons that this sort of failure might occur. The most common example is that, under Unix/Linux, the user might have neglected to run Kerberos kinit.

The one reason that is unlikely is that the agent is unhappy with the eMake client--that type of failure is the subject of EC2128.

### **Fixes**

Check the error text for clues as to which stage of authentication failed. For example, under Unix/Linux, be sure the user has run Kerberos kinit (or the equivalent thereof).

## **EC1172**

### **Summary**

An unrecognized agent security mode was specified.

### **Description**

In a cluster build, eMake will attempt to secure its connections with agents according to a mode set by the first of the following that exists:

- "--emake-agent-security=..." command line option

- "EMAKE\_AGENT\_SECURITY" environment variable

If none of the above specifies the mode, then eMake defaults to unauthenticated communications with agents.

But if one of the above specifies an unrecognized mode, then eMake fails with EC1171.

### **Reasons**

The user specified an unrecognized agent security mode.

### **Fixes**

Check the text of the error message for a list of available modes, and choose one of them.

## EC1173

### Summary

eMake failed to initialize needed security facilities.

### Description

eMake was asked to authenticate itself to other parts of the cluster, but it failed to initialize the needed security facilities. Please examine the error text for details concerning this issue.

### Reasons

For example, on Linux we might have failed to locate the GSSAPI shared object library.

### Fixes

On Linux, if Kerberos has not been installed on the eMake client host, then install it there. If it is already installed, then be sure that its GSSAPI shared object library is included in the environment variable LD\_LIBRARY\_PATH. If it is, then set the environment variable ECLLOUD\_GSSAPI\_LIBRARY to the filename (typically without directory information) of the Kerberos GSSAPI library that is to be used.

## EC1174

### Summary

The user has invoked eMake with an invalid value for the `--emake-clearcase` command-line option.

### Description

eMake has built-in support for a variety of ClearCase features, which can be enabled on a feature-by-feature basis by specifying the `--emake-clearcase` option when eMake is invoked. The `--emake-clearcase` option value is specified as a comma separated list of option values. The following values are supported:

Value	Meaning
vobs	Enable per-VOB temporary directories to improve performance
rofs	Enable awareness of ClearCase read-only filesystems
symlink	Enable awareness of ClearCase symlinks (Windows only)
*	Equivalent to vobs,rofs,symlink

eMake will display error EC1174 if the user specifies an invalid option value for the `--emake-clearcase` command-line option.

### Reasons

The user has specified an invalid value for the `--emake-clearcase` command-line option.

### *Fixes*

The invalid option value will be reported in the error message; remove that value from the `--emake-clearcase` option to resolve this error.

## **EC1175**

### *Summary*

Bad warning number in argument to "`--emake-hide-warning`" or in environment variable "EMAKE\_HIDE\_WARNING".

### *Description*

The argument to "`--emake-hide-warning`" should be a comma-separated list of zero or more warning numbers; the same is true for any value assigned to "EMAKE\_HIDE\_WARNING". If an element of such a list is not the decimal representation of a positive integer, or is implausibly large, then eMake will issue this error. (Currently the application does not check if the given numbers are actually assigned, and perhaps that is best for forward compatibility.)

### *Reasons*

See above.

### *Fixes*

The error message text will mention the first offending list element. Replace it with an actual warning number, and check the remainder of the list for similar errors.

## **EC1176**

### *Summary*

eMake created a local temporary file but was unable to delete it.

### *Description*

eMake sometimes creates local temporary files, often as response files containing command line arguments. This warning triggers when eMake fails to delete such a file.

### *Reasons*

Probably something changed between the time when eMake created the temporary file and the time eMake tried to delete it. For example, a build step may have altered the permissions of the file or its containing directory.

### *Fixes*

If filesystem permissions changed, either eliminate that change or find a different place to locate the temporary file. If the temporary file must be relocated but its location is not under user control, then contact product support.

## **EC1182**

### *Summary*

Emake, Electrify, or electrifymon failed to terminate itself by a signal, and had to exit instead.

### *Description*

Emake, Electrify, or electrifymon tried to terminate itself by a particular UNIX/Linux signal, usually after trapping such a signal and cleaning up, or in order to emulate the termination status of another process. However, that attempt failed, and so it exited with the specified exit code instead.

### Reasons

This error should not occur, though it is conceivable that it could be caused by very high load on the same machine.

### Fixes

Contact Electric Cloud.

## EC1183

### Summary

This warning is issued when `#pragma multi` is applied to a pattern rule in GNU Make emulation mode.

### Description

GNU Make pattern rules already support the notion of multiple output files produced simultaneously, and therefore `#pragma multi` is redundant on them.

### Fixes

Remove `#pragma multi` from the pattern rule; it does nothing, and so removing it should have no effect other than to eliminate the warning.

## EC1184

### Summary

`#pragma multi` on rules that contain a mix of pattern and non-pattern targets is not supported.

### Description

GNU Make allows pattern targets to follow non-pattern targets on the same line, even though it forbids the reverse. `#pragma multi` on such constructs is not supported.

### Reasons

For example, `#pragma multi` is not supported in this case:

```
#pragma multi
nonpat %.aa %.bb %.cc: ; @echo building ...
```

### Fixes

Split the rule into a non-pattern rule and a pattern (which is essentially what is meant without the `#pragma multi`). If desired, then apply `#pragma multi` to the non-pattern part only.

## EC1185

### Summary

`#pragma multi` on static pattern rules is not supported.

### Description

A static pattern rule is like a cookie cutter for building several specific targets individually, each time using the same pattern to compose the rule. It makes little sense to say that the static pattern should treat the targets as built simultaneously because a single invocation of the commands would only build just one of the specified targets.

### Reasons

You attempt to apply `#pragma multi` to a static pattern rule.

### Fixes

If what you mean is that the commands really do build all of the targets together, then restructure the rule as an ordinary `#pragma multi` rule, not a static pattern. Otherwise, drop `#pragma multi`.

## EC1186

### Summary

`#pragma multi` double-colon rules is not supported.

### Description

Currently, a meaning is not assigned to `#pragma multi` for double-colon rules; that combination is reserved for a possible future version of ElectricAccelerator.

Multiple double-colon rules may share the same target; all of the commands run. Typically they are used for targets such as "clean", and imply an extensible collection of actions to be taken when their target is requested. By contrast, the point of `#pragma multi` is to run the commands in question only once, even if more than one of its targets is needed. So `#pragma multi` tends to reduce the number of commands run, whereas double-colon rules are used to run more and more commands, often with the collection growing as more makefile inclusions are performed. Thus, the goals of the two features are at least superficially at odds.

Furthermore, `#pragma multi` requires agreement on target sets and prerequisite lists, and that could be quite difficult to coordinate in an extensible collection of rules such as a family of double-colon rules with overlapping targets.

### Reasons

You applied `#pragma multi` to a double-colon rule.

### Fixes

Drop `#pragma multi` from the double-colon rule. If, however, you have an idea of what it should mean in this case, and have considered the problems mentioned above, then contact Electric Cloud to make a feature request.

## EC1187

### Summary

You must use or avoid `#pragma multi` consistently in any collection of rules and dependency lines sharing a given target.

### Description

In order to keep the meaning clear, if any rule or dependency line with a given target is `#pragma multi`, then all rules and dependency lines including that same target must also be `#pragma multi`. Otherwise it is not clear whether the targets should be treated as simultaneously updated, and if so, with what other targets.

### Reasons

There are two rules or dependency lines with overlapping target sets, one with `#pragma multi` and one without.

### Fixes

Either make all of the relevant rules and dependency lines `#pragma multi`, or remove that pragma from all of them.



## EC1188

### Summary

When one target of a `#pragma multi` rule has a separately declared dependency, then that dependency must specify the same target set as the rule. Otherwise there could easily be different prerequisite lists triggering the same rule, depending on which target is considered.

### Description

If the prerequisite lists of different targets of a `#pragma multi` rule were allowed to differ, then it would be unclear which prerequisites trigger the commands in that rule. By requiring the same target set in a separate dependency specification, a pattern of usage is enforced that makes it clear which prerequisites trigger the commands in the `#pragma multi` rule.

For uniformity this rule is enforced even if no commands are mentioned explicitly.

### Reasons

This error triggers when `#pragma multi` is applied to overlapping but differing target sets.

### Fixes

Reconcile the target sets.

## EC1189

### Summary

Ordinary rules may override the commands of other ordinary rules, though there is a warning. For `#pragma multi`, command overriding is prohibited.

### Description

Overriding of the commands for a single target is tricky enough, but where multiple targets are to be updated simultaneously, the practice of overriding commands is even more confusing, and so it is prohibited.

### Reasons

This error triggers when two `#pragma multi` rules with the same target set both specify commands.

### Fixes

Pick the commands you want, and drop the other ones.

## EC1190

### Summary

A `#pragma multi` dependency specification was not supplemented by a `#pragma multi` rule with commands and the same target set.

### Description

Emake will not search implicit rules for commands to build `#pragma multi` targets. Suffix rules build targets individually, which does not make sense in combination with `#pragma multi`. GNU-Make-style pattern rules can build multiple targets simultaneously, but `#pragma multi` is unnecessary in that context (and there are subtle issues involving prerequisites that work differently between the two mechanisms).

Therefore without an explicit set of commands for the target set in question, eMake will not have a way to build those targets.

### Reasons

A `#pragma multi` dependency specification was not supplemented by a `#pragma multi` rule with commands and the same target set.

### **Fixes**

Either add an explicit `#pragma multi` rule with commands and the same target set to indicate how emake should simultaneously build those targets, or in the case of commands specified within a pattern, drop `"#pragma multi"` from the dependency specification and instead rely upon the established semantics of GNU-Make-style pattern rules.

## **EC1289**

### **Summary**

There was a problem with a local agent license.

### **Description**

The cluster manager handles agent licensing for cluster agents. However, the eMake client restricts the use of local agents based on a local license. This error message appears if the license is invalid, or if you are using local agents without a cluster and no license exists.

### **Reasons**

Licenses can be unsatisfactory in various ways. See the details of the error text for more information.

### **Fixes**

Provide a valid license for local agents. In particular, it must allow the local agent feature.

## **EC2002**

### **Summary**

An unrecognized code was seen by the agent on an eMake client connection.

### **Description**

When the agent receives commands from the eMake client, they start with code numbers. One such code was not recognized.

### **Security Exposure**

Issuing this error on a connection that requires authentication but has not yet been authenticated should be harmless, because all it tells the attacker is that the command code was not recognized, and what the last command was on that communications channel. The last command is quite predictable before authentication, and so no real information is exposed thereby. Knowledge that a command code is unrecognized would at most provide the agent protocol version, which could be fetched directly by `E2A_GET_VERSION`--which does not currently require authentication (see Accelerator Kerberos Authentication for a discussion of that exposure).

### **Reasons**

Usually eMake and the agent check the version of the protocol that they will use before doing real work, and so only a bug, network glitch, and so on, would cause an unrecognized code to be sent to the agent from the eMake client.

### **Fixes**

Contact Electric Cloud technical support for further investigation.

## **EC2093**

### **Summary**

Clock skew between eMake and agent is %d ms.

### **Description**

This indicates that the system time on the eMake machine is different than the time on the reported agent.

### **Reasons**

Clocks on computers drift over time, so unless the customer takes precaution to avoid that times will go out of sync over time.

From a correctness perspective, the Accelerator should not be affected by the clock skew. The up to date checks will only use file time stamps. The main reason this is being reported is that customers that use the system clock directly or indirectly (through some third party tool) may experience unexplainable issues.

### **Fixes**

The customer should consider setting up NTP on all machines. This network service will guarantee consistent time on all machines.

In exclusive mode (only one active build per agent host) the warning will not occur, as the time on the agent is set to the eMake machines time at the beginning of the session.

In shared mode (multiple active builds per agent host) the clock skew threshold can be configured in the Cluster Manager. If there is a need to suppress the warning, a large value for the threshold may help.

## **EC2096**

### **Summary**

This warning indicates a Cygwin mount conflict.

### **Description**

See above.

### **Reasons**

A specified directory may not be accessible to an agent and cannot be mounted.

### **Fixes**

Verify that the agents have permission to access the locations. For Windows, agents are not run using the user ID of the user that invoked eMake; agents run as ECloudInternalUser##.

You can also map that location to the eMake system and include that drive as part of the eMake root.

## **EC2128**

### **Summary**

The authentication offered by the eMake client is hereby rejected.

### **Description**

The agent has rejected an attempt by the eMake client to authenticate itself.

### **Security Exposure**

If our policy was instead to simply drop the agent connection, then attackers could reasonably infer that a dropped connection means authentication failure, because other explanations (such as network connectivity interruptions) are much less likely.

Thus in order to convincingly hide authentication rejection, we would have to provide a convincing simulation of authentication success. The more convincing a simulation, the more computational resources it would consume, and the more confusing it would be to legitimate users and administrators when authentication fails by accident. At least for now, it seems better to focus on detecting actual vulnerabilities, rather than confusing attackers into wasting effort on secure systems.

### **Reasons**

Various causes are possible. Check the agent logs for details--for security reasons the agent intentionally withholds the details of the failure.

In particular, if on Linux you see "Security Error: gss\_accept\_sec\_context: GSSAPI Minor Status 0x2: No such file or directory" in the agent log, then you may need to install a keytab or tell the agent where to find it; see Kerberos Usage in ElectricAccelerator for details.

### **Fixes**

The fix will depend upon the cause, but be suspicious because most innocent causes of failure do not trigger this error--they are much more likely to trigger EC1171.

## **EC2129**

### **Summary**

The agent requires authentication for this eMake client connection.

### **Description**

If an eMake client connects to an agent that requires authentication, but does not attempt authentication, then it receives this error.

### **Security Exposure**

An attacker could attempt to use the agent without authentication. If that attempt succeeded, then the attacker could predict the absence of this message. If it failed, then the attacker could reasonably predict that this message would issue. Thus, this message provides no additional information to an attacker.

### **Reasons**

See above.

### **Fixes**

Enable authentication on the eMake client.

## **EC2130**

### **Summary**

Authentication is not offered for this eMake client connection.

### **Description**

eMake to agent communications require agreement on security settings. If authentication is disabled on the agent, then it will reject requests to authenticate.

### **Security Exposure**

This message only issues from agents that do not require authentication. Instead of attempting to trigger this message, an attacker could simply exploit the insecure agent, and the success of that attempt would provide the information that this message would have provided. Therefore this message does not provide any information that an attacker could not get another way.

### **Reasons**

See above.

### **Fixes**

Enable authentication on the agent.

## EC2131

### *Summary*

Authentication was already performed for this eMake client connection.

### *Description*

eMake should not perform redundant authentication on a connection.

### *Security Exposure*

This message issues only on an authenticated communications channel. We do not currently defend against TCP injection attacks, and any information provided by this message would be the least of the exploits possible in such a scenario. If we later add defenses against such attacks, then an attacker would not have the ability to trigger this message (without having already gained authenticated access), and so the information it provides would be harmless.

### *Reasons*

A bug in the code.

## EC2132

### *Summary*

On Unix or Linux, authentication is required and the user ID available to the eMake client is not allowed on one of the agent hosts involved in the build.

### *Description*

When authentication is required, a Unix or Linux agent allows only the user and group IDs that are available to the authenticated user on its host (the machine on which the agent is running). If the eMake client specifies other user or group IDs than those available on the agent host, then the agent eliminates the unauthorized user and group IDs. If the actual user ID is not the one requested, then the agent sends this warning back to the eMake client.

### *Reasons*

The Kerberos principal used in authentication, namely the principal passed to kinit or similar, specifies a particular user on the agent host. That principal may differ from the user who is running eMake. Alternatively, there is a mismatch between the eMake client host and the agent host regarding which user ID is assigned to that user.

### *Fixes*

First check that the principal used in authentication (typically the argument to kinit) matches the Unix/Linux user account that is running eMake. If not, then rerun kinit (or similar) with a principal that does match the Unix/Linux user account (or switch to a different Unix/Linux user account and rerun kinit).

If the principal and user match, then log in as that user on both the eMake client host and the agent host, run `id` in both accounts, and compare the results. Do they match? If not, then change the user ID on one of the machines to match the other, and reconcile the groups as well in order to avoid EC2133 and EC2134. Be sure that all agent hosts match each other, too, or you will just get the same error with a different agent host.

You can also check to see if there is some type of `setuid/sudo` wrapper around eMake.

## EC2133

### *Summary*

Like EC2132, except that in this case the user ID matches, but the primary group ID requested by eMake is not available to that user on an agent host.

**Description**

See above.

**Reasons**

The primary group currently in use on the eMake client is not available to the same user on an agent host.

**Fixes**

Either grant membership in the group on all agent hosts, or change to a different primary group on the eMake client. But in the latter case, if you keep the original primary group in the supplementary group list, then you will probably encounter EC2134.

**EC2134****Summary**

Like EC2132, except that in this case the user ID matches, and the primary group ID requested by eMake is available to that user, but one or more supplementary group IDs requested by eMake is not available to that user on an agent host.

See also EC2133, which is for an unavailable primary group ID.

**Description**

See above.

**Reasons**

A supplementary group available to the eMake client is not available to the same user on an agent host.

**Fixes**

Either grant membership in the group on all agent hosts, or deny membership on the eMake client.

**EC2135****Summary**

eMmake sent an empty group list.

**Description**

eMake sent an empty group list in E2A\_SET\_USER, which should never happen, except possibly on Windows, where E2A\_SET\_USER is ignored.

**Reasons**

There is a bug in eMake, or some type of fake eMake client that is sending an empty group list, perhaps as an attack.

**Fixes**

Contact product support.

**EC2136****Summary**

An agent failed to create a file.

**Description**

Agents sometimes creates files, often as response files containing command line arguments. This warning triggers when an agent fails to create such a file.

### **Reasons**

File permissions are a likely culprit. Or perhaps a directory that should exist does not.

### **Fixes**

Check the details of the failure message and correct the problems mentioned there.

## **EC2137**

### **Summary**

An agent created a temporary file but was unable to delete it.

### **Description**

Agents sometimes creates temporary files, often as response files containing command line arguments. This warning triggers when an agent fails to delete such a file.

### **Reasons**

Probably something changed between the time when the agent created the temporary file and the time that same agent tried to delete it. For example, a build step may have altered the permissions of the file or its containing directory.

### **Fixes**

If filesystem permissions changed, either eliminate that change or find a different place to locate the temporary file. If the temporary file must be relocated but its location is not under user control, then contact product support.

## **EC3015**

### **Summary**

This error indicates a potentially unsafe mount directory.

### **Description**

See above.

### **Reasons**

One of the following is defined as emake-root:

```
/
/proc
/dev
/var/tmp
/etc
/bin
```

### **Fixes**

Ensure none of the above are defined as emake-root.

## **EC3017**

### **Summary**

An attempt was made to download a file from the Cluster Manager without first connecting to it.

### **Description**

An ElectricAccelerator software component has attempted to download a file from the Cluster Manager without first connecting to the Cluster Manager.

***Fixes***

Contact Electric Cloud.

**EC3018*****Summary***

Failed to open a local file.

***Description***

An ElectricAccelerator component attempted to download a file from the Cluster Manager, but the local copy of the file could not be opened.

***Reasons***

Any of the usual reasons a program might fail to open a file.

***Fixes***

Check the details of the error message to figure out why the file could not be opened: containing directory does not exist, permissions, and so on.

**EC3019*****Summary***

Failed to close a local file.

***Description***

An ElectricAccelerator component attempted to download a file from the Cluster Manager, but the local copy of the file could not be closed.

***Reasons***

Any of the usual reasons a program might fail to close a file.

***Fixes***

Check the details of the error message to figure out why the file could not be closed: I/O error, disk full, and so on.

**EC3020*****Summary***

The Cluster Manager sent a response, but it was not received.

***Description***

An ElectricAccelerator component was in the process of receiving a response from the Cluster Manager when a local error occurred.

***Reasons***

Out of memory, failure to write a downloaded file to disk, and so on.

***Fixes***

Check for disk-full conditions and free up space. Contact Electric Cloud.



## EC3101

### Summary

emake displays this error when the build is stopped using the Cluster Manager.

### Description

Error EC3101 indicates that the build identified in the error message was deliberately terminated using the Cluster Manager. This is considered a fatal error: emake will immediately release all cluster resources, cease executing the build, and exit with exit code 2.

### Reasons

EC3101 will be displayed only if a user explicitly terminates the build using the Cluster Manager. There are two ways a user can terminate the build through the Cluster Manager:

- Click the "Stop Build" link for the build, either on the "Builds" tab or on the "Edit Build" page for a running build:
- Invoke the stopBuild command in cmtool: `cmtool --cm=mycm stopBuild 12345`

### Fixes

There is no fix, per se, because this error is only shown in response to an explicit user action.

## EC3102

### Summary

This error means the eMake process failed to notify the Cluster Manager before it timed out. eMake stops at that point.

### Description

See above.

### Reasons

The question is, why were there no heartbeats from Electric Make (eMake) to the Cluster Manager? Possible reasons:

- Network problems, where either the Cluster Manager or the eMake machine was unable to see the other machine.
- The eMake heartbeat thread (which is the AgentManager thread) is excessively slow. One particular reason can be excessive logging that causes the interval between heartbeats to exceed the default duration of 1 minute.
- At times the Cluster Manager can run out of file descriptors due to a very small number of system level file descriptors (the default was increased). Another reason for running out of file descriptors could be the LSF request queue being stalled by the LSF server, causing the Cluster Manager to stack up more and more requests (fixed in an earlier release).

### Fixes

- Increase the number of open file descriptors (it should be in the many thousands). The procedure varies for each operating system.
- Increase the Cluster Manager's timeout duration (default is 1 minute). Be careful not to make the timeout too long because potentially dead builds may continue to hold on to agents.  
Follow these steps:

1. Go to the "<ECloud Install>\<arch>\conf" directory on the Cluster Manager and edit "accelerator.properties".
2. Increase the number for "EMAKE\_HEARTBEAT\_TIMEOUT=".
3. Save the changes for "accelerator.properties".
4. Restart the Cluster Manager.

## EC3148

### *Summary*

Usually the brief description from the Messages.h file.

### *Description*

NoResourceManager: The build requested resource 'accelerator', but the server is not configured to support resource management.

### *Reasons*

Scenario: Resources were created and configured on the Cluster Manager. This feature is only active and dependent on a switch in the Resource Manager Type. At some point, someone or some process modified the Resource Manager Type. Subsequently, if a build specifies a resource, errors result because the resource is not there for use.

### *Fixes*

Go to the Administration tab > Server Settings. For Resource Manger Type, select built in.