



ElectricAccelerator Installation and Configuration Guide

version 6.2

Electric Cloud, Inc.
676 W. Maude Avenue
Sunnyvale, CA 94085
www.electric-cloud.com

Copyright © 2002 - 2012 Electric Cloud, Inc. All rights reserved.

Published September 2012

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, screen display appearance, and so on.

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

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

Other product names mentioned in this guide may be trademarks or registered trademarks of their respective owners and are hereby acknowledged.

Contents

Chapter 1: ElectricAccelerator Introduction	1-1
About ElectricAccelerator	1-1
Cluster Manager	1-1
Electric File System (EFS)	1-1
ElectricAccelerator Agent (Agent)	1-2
Electric Make (eMake)	1-2
Electrify	1-2
Understanding Component Interactions	1-2
Electric Make and EFS	1-3
Electric Make and Cluster Manager	1-3
Cluster Manager and Other Components	1-3
When to Use the ElectricAccelerator Cluster	1-4
Grid Integration	1-4
Chapter 2: System Requirements and Supported Platforms	2-1
System Requirements and Supported Platforms	2-1
Linux	2-1
Solaris	2-3
Windows	2-3
Notes to ensure a consistent build environment	2-4
Notes regarding 64-bit vs. 32-bit usage	2-4
Known Limitations	2-5
Hardware Requirements	2-5
Software Requirements	2-6
Third-party Supported Build Tools	2-7
Database Support	2-10
The "checksum" utility	2-10
Default Installation Directories	2-10
Installation Log Locations	2-10
Agent Log Locations	2-10
Disk Cache Directory and Agent Temporary Storage Location	2-11
Known Linux Kernel Issue and ElectricAccelerator Performance	2-11
Planning for Grid Use	2-12
Chapter 3: Installing ElectricAccelerator	3-1
General Information	3-1
Installation Notes	3-1
User Interface Installation Method	3-2
Cluster Manager	3-2

Electric Agent/EFS	3-9
Electric Make	3-13
Additional Install Options	3-15
Interactive Command-line Installation Method	3-16
Cluster Manager	3-16
Electric Agent/EFS	3-17
Electric Make	3-18
Additional Install Options	3-19
Path Settings	3-19
Silent Installation Method	3-20
Installer Command Line Options	3-20
Creating an Installer Properties File	3-26
Automating a Linux/Solaris Silent Install	3-26
Automating a Windows Silent Install	3-26
Migrating an eMake only installation to a different machine	3-27
Installed Tool Files	3-27
ElectricAccelerator Developer Edition Installation	3-27
User Interface Installation Method	3-28
Interactive Command-line Installation Method	3-30
Post-Installation Activities	3-31
Microsoft Visual Studio	3-31
Installing an Apache Server Certificate	3-32
Testing the Cluster Manager Installation	3-34
Uninstalling ElectricAccelerator	3-34
Chapter 4: Upgrading ElectricAccelerator Components	4-1
Upgrade Notes	4-1
Upgrading Multiple Agents on Cluster Hosts	4-2
User Interface Upgrade Method	4-4
Electric Agent/EFS	4-4
Cluster Manager	4-7
Electric Make	4-8
Interactive Command-line Upgrade Method	4-8
Electric Agent/EFS	4-8
Cluster Manager	4-8
Electric Make	4-9
ElectricAccelerator Version Compatibility Matrix	4-9
Additional Explanation	4-10
Installing the backward-compatible eMake package on agents	4-10
Chapter 5: Initial Configuration Tasks	5-1
Logging In and Licensing	5-1
ElectricAccelerator Home Page	5-4
Server Information	5-4
Using Comments	5-5
Enabling Resource Management	5-5
Agent Log Rotation	5-6
Configuring log rotation	5-6

Using the accelerator.properties File	5-8
Changing Build and Accelerator Log Locations	5-8
Disabling Agents by Default	5-8
User Authentication	5-8
Setting Up Email Notification	5-13
Adding Custom Protocol Mismatch Error Text	5-14
Using the Database Connection Monitor	5-14
Platform-specific Information	5-15
RHEL 6 and Accelerator services started from a shell	5-15
Solaris 10 Configuration/debugging	5-15
Using the Web Interface	5-15
Accessing Other ElectricAccelerator Information	5-15
Chapter 6: Integrating ElectricAccelerator with the Grid	6-1
Grid Host Manager	6-1
Examples	6-2
LSF-specific Information	6-5
Setting Up LSF	6-6
Additional Information	6-7
Chapter 7: Windows-Specific Information	7-1
Agent-Side Application Setup	7-1
Visual Studio	7-1
Microsoft Office	7-2
WinZip	7-2
MSBuild	7-2
Antivirus Software	7-2
Cygwin	7-2
Additional Information	7-2
Registry-Specific Information	7-3
Registry Use Under Windows	7-3
Registry Underlay	7-4
ExcludeProcessList Registry Entry	7-4
Windows Kernel Non-paged Pool Memory	7-5
Support for Managed Code/.NET	7-5
Glossary of Terms	7-5

Chapter 1: ElectricAccelerator Introduction

ElectricAccelerator® is a collection of software components that manage and accelerate the build process. This guide walks you through the ElectricAccelerator installation process for UNIX or Windows systems and is intended for administrators, build engineers, and software engineers.

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.

During ElectricAccelerator installation, the following components will be installed:

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

For a complete list of “runnable” programs installed on your cluster, see the “*Installed Files/Tools*” help topic in ElectricAccelerator online help.

Cluster Manager

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.

Electric File System (EFS)

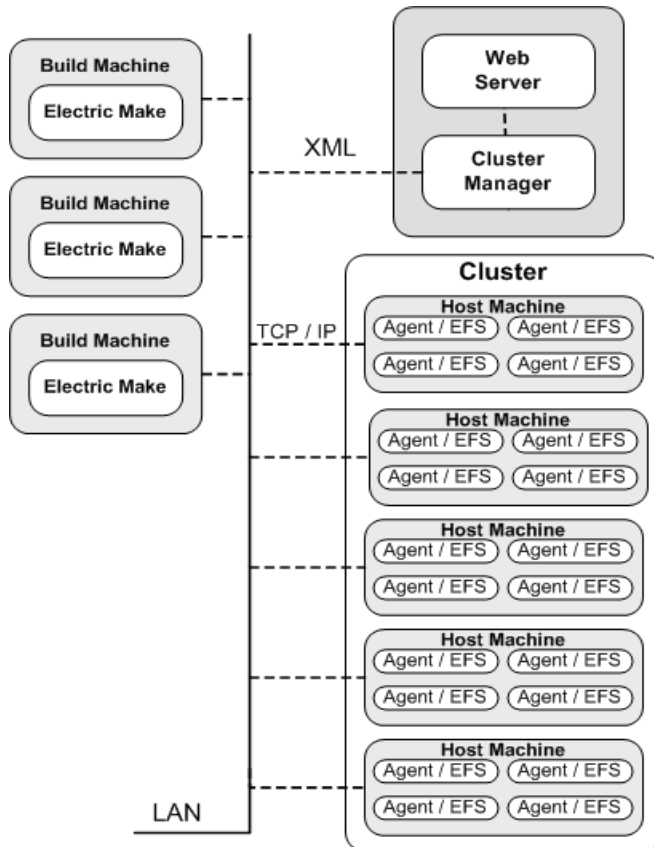
Electric File System (EFS) is a special-purpose file system driver, monitoring every file access and providing Electric Make with complete usage information. This driver collects dependency information, which allows Electric Make to automatically detect and correct out-of-order build steps. Each EFS 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.

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.



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.

Understanding Component Interactions

To a user, ElectricAccelerator may appear identical to other Make versions—reading makefiles in several different formats and producing identical results. Using a cluster for builds is transparent to the ElectricAccelerator user.

Important differences in ElectricAccelerator build processing versus other distributed systems:

- ElectricAccelerator components work together to achieve faster, more efficient builds. Instead of running a sequential build on a single processor, ElectricAccelerator executes build steps in parallel on a cluster of hosts.
- For fault tolerance, job results are isolated until the job completes. If an Agent fails during a job, ElectricAccelerator discards any partial results it might have produced and reruns the job on a different Agent.
- Missing dependencies discovered at runtime are collected in a history file that updates each time a build is invoked. ElectricAccelerator uses this collected data to improve performance of subsequent builds.

Electric Make and EFS

High concurrency levels in ElectricAccelerator are enabled by the Electric File System (EFS). When a job such as a compilation runs on a host, it accesses files such as source files and headers through EFS. EFS records detailed file access data for the build and returns that data to Electric Make.

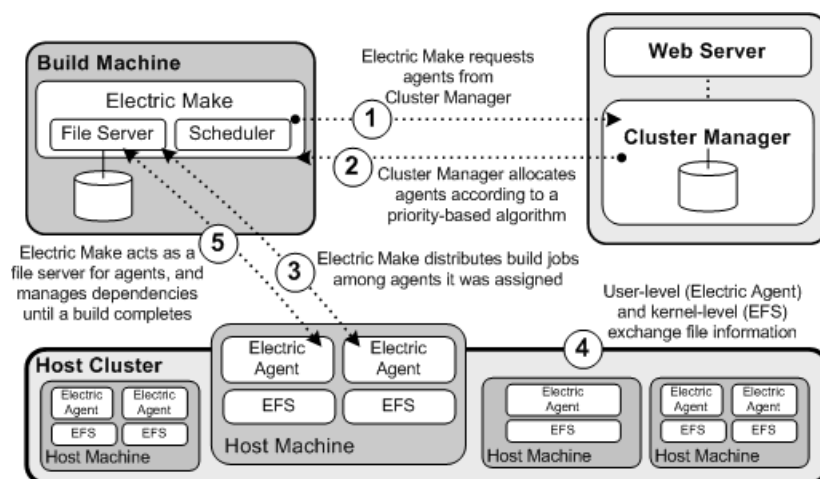
Electric Make acts as a file server for Agents, reading the correct file version from file systems on its machine and passing that information back to the Agents. Agents retain different file version information and do not rely on Electric Make's file sequencing ability to provide the correct version for a job. The Agent receives file data, downloads it into the kernel, notifying EFS, which then completes the original request. At the end of a job, Electric Agent returns any file modifications to Electric Make so it can apply changes to its local file systems.

Electric Make and Cluster Manager

When Electric Make is invoked on the build machine, it communicates with Cluster Manager to acquire a set of Agents it can use for the build. When Electric Make finishes, it sends Cluster Manager the build results, and tells Cluster Manager that Agents are free now to work on other builds. If more than one build is invoked, Cluster Manager allocates agents using a priority-based algorithm. Builds with the same priority share Agents evenly, while higher priority builds are allocated more Agents than lower priority builds. By default, agents running on the same host machine are allocated to the same build. In real time, Cluster Manager dynamically adjusts the number of agents assigned to running builds as each build's needs change, which allows ElectricAccelerator to make the best use of cluster resources.

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 the work to 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. In local mode, Electric Make behaves exactly like your existing Make, running all jobs in series on the local system.

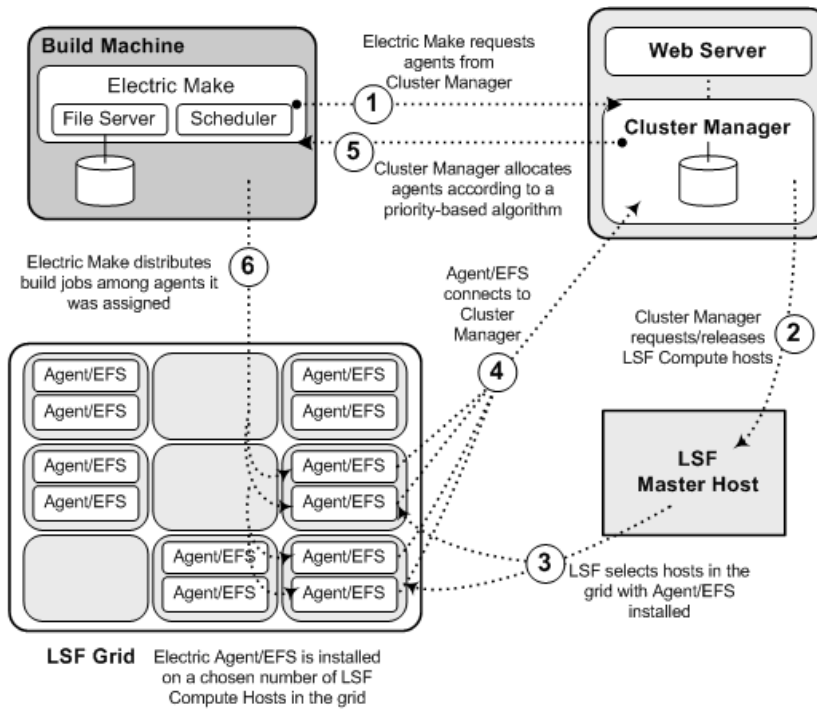
Grid Integration

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 does not interact directly with the 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 the 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.



Chapter 2: System Requirements and Supported Platforms

This chapter describes hardware and software requirements for installing and running ElectricAccelerator on Windows or UNIX systems.

System Requirements and Supported Platforms

Refer to the ElectricAccelerator Release Notes for the latest updates to this information.

ElectricAccelerator Cluster Manager, Electric Make, and Electric Agent/EFS currently run on:

Linux

Red Hat Enterprise Linux version	Notes
RHEL 6.0, 6.1, 6.2 kernel 2.6.32-x	For 64-bit systems, you must install 32-bit libraries before invoking the installer. Run <code>yum install glibc.i686</code> . If yum server is not present, the required RPMs are <code>glibc-2.11.1.1.10.el6.i686</code> and <code>nss-softokn-freebl-3.12.4-11.el6.i686</code> . Be advised of a known Linux kernel issue .
RHEL 5.4 and later kernel 2.6.18-x	For agent hosts, you must install the corresponding kernel-devel version package before invoking the installer. Be advised of a known Linux kernel issue .
RHEL 4.8 and later kernel 2.6.9-x	For agent hosts, you must install these three packages: <code>kernel-devel</code> , <code>gcc</code> , and <code>gcc-c++</code> , before invoking the installer.

SUSE Linux Enterprise Server version	Notes
SLES 11, 11 SP1, 11 SP2 kernel 2.6.27, 2.6.32, 3.0.10	For agent hosts, you must install the gcc and kernel-source packages before invoking the installer. To install gcc, run <code>sudo zypper install gcc</code> (This installs the following: gcc gcc43 glibc-devel linux-kernel-headers.) To install kernel-source, run <code>sudo zypper install kernel-source</code>
SLES 10 SP4 kernel 2.6.16	

Ubuntu version	Notes
Ubuntu 12.04 kernel 3.2.0	For 64-bit systems, you must install 32-bit libraries before invoking the installer. Run <code>sudo apt-get install ia32-libs</code> to install the 32-bit libraries.
Ubuntu 11.04, 11.10 kernel 2.6.38, 3.04	
Ubuntu 10.04.1, 10.10 kernel 2.6.32, 2.6.35	
Ubuntu 9.10 kernel 2.6.31	

Ubuntu Notes:

- MySQL 5.5.16 requires additional libraries on the Cluster Manager machine. If Ubuntu 64-bit is missing the libraries, you may need to install i386 deb.

Linux Notes (all versions):

- Linux platforms require additional libraries to run MySQL 5.5.16. To run MySQL 5.5.16, the **Cluster Manager must have the 32-bit version of the following libraries:**
 - libaio.so.1
 - libncurses.so.5

You may need to add those libraries to the Cluster Manager machine.

RHEL example using a package manager:

```
yum install libaio.i686
yum install ncurses-libs.i686
```

RHEL example without a package manager:

```
up2date --nox --arch=i386 libaio
up2date --nox --arch=i386 ncurses
```

Ubuntu example using a package manager:

```
apt-get install libaio1
```

- Ensure that SELinux is disabled because it can interfere with ElectricAccelerator. To disable SELinux, modify /etc/selinux/config by changing SELINUX=enforcing to SELINUX=disabled.
- ElectricAccelerator does not support extended attributes (xattr). Attempting to query or set extended attributes for a file returns an ENOTSUPP ("Operation not supported") error.
- For all supported Linux platforms, both 32-bit and 64-bit are supported.
- For 32-bit, only x86 is supported. For 64-bit, only x86-64 is supported. IA-64 (Itanium) is not supported.

Solaris

Solaris version	Notes
Solaris 10	SPARC and x86
Solaris 9	SPARC only Agent and eMake only, no Cluster Manager support

Solaris Note (all versions):

- Application is 32-bit only for Cluster Manager and eMake, and 64-bit only for agents.
- Solaris 9 support for Cluster Manager machines ended as of Accelerator v6.1. Because Accelerator now uses MySQL 5.5, and MySQL 5.5 does not support Solaris 9, Accelerator no longer supports Solaris 9 as a Cluster Manager platform.

Solaris x86 Note:

- For Solaris x86, eMake relies on several third-party open source libraries. The version of libxml2 that Electric Cloud builds in-house is not the same as the version that is available on SunFreeware.com. If you run eMake and get an error such as:

```
ld.so.1: emake: fatal: libxml2.so.2: version `LIBXML2_2.7.6' not found
(required by file /opt/eccloud/i686_SunOS.5.10/bin/emake)
ld.so.1: emake: fatal: libxml2.so.2: open failed: No such file or direc-
tory
```

then it is likely that your LD_LIBRARY_PATH environment variable contains a directory that includes a version of libxml2.so.2 that is not compatible with eMake. If you unset LD_LIBRARY_PATH entirely, eMake's rpath is configured to find all of its necessary libraries.

Windows

Windows version	Notes
Windows 7 (32- or 64-bit)	
Windows Server 2008 R2 (64-bit only)	

Windows version	Notes
Windows Vista (32- or 64-bit)	
Windows Server 2003 R2 (32- or 64-bit)	SP1 and SP2 required
Windows XP SP3 (32-bit only)	

Windows Notes (all versions):

- NTFS is required for all Windows machines.
- For 32-bit, only x86 is supported. For 64-bit, only x86-64 is supported. IA-64 (Itanium) is not supported.
- 64-bit registry mirroring is supported only if you use a 64-bit Agent/EFS (running on 64-bit Windows Vista/Server 2008 R2) with 64-bit eMake (running on any 64-bit Windows platform). Registry mirroring does not work under 64-bit Windows Server 2003 because of an operating system limitation.
- To run Electric Make with Visual Studio on Windows, one of the following is required:
 - Visual Studio 2005 SP1
 - Visual Studio 2005 SP1 Redistributable Package (if you use Visual Studio .NET 2002 or 2003). The following URLs are provided for your convenience:

<http://www.microsoft.com/downloads/details.aspx?familyid=200B2FD9-AE1A-4A14-984D-389C36F85647&displaylang=en> (32-bit)

<http://www.microsoft.com/downloads/details.aspx?familyid=EB4EBE2D-33C0-4A47-9DD4-B9A6D7BD44DA&displaylang=en> (64-bit)

Notes to ensure a consistent build environment

Electric Cloud recommends all hosts in a cluster have synchronized clocks, which is important particularly if you are running in the “shared” agent allocation policy.

- For UNIX machines, you can use `ntpdate` to sync clocks.
- For Windows machines, participation in a domain generally ensures synchronized clocks.

Notes regarding 64-bit vs. 32-bit usage

- The installer for Windows and Linux provides both 32-bit and 64-bit versions of Electric Make. (For Solaris, only 32-bit is available.)
- The ElectricAccelerator installer automatically sets the path to 32-bit. To use 64-bit, you must edit the environment variable path to include the 64-bit bin location before the 32-bit location.
- 64-bit executables are installed in a new subdirectory of the install location, `<installDir>/64/bin`.

For example:

If you install into the Windows `C:\ECLoud\i686_win32` directory, the 64-bit executables are in the `C:\ECLoud\i686_win32\64\bin` directory.

For Linux, the directory is `/opt/eccloud/i686_Linux/64/bin`.

- Though 32-bit Electric Make can be used with 32- or 64-bit cluster hosts, Electric Cloud recommends using 64-bit Windows Vista/Server 2008 R2 cluster hosts with 64-bit eMake only. This is because on Windows, eMake uses `--emake-reg-roots` by default, even if it is not set. Using 32-bit eMake with a 64-bit Windows Vista/Server 2008 R2 Agent/EFS causes registry mirroring issues.

- The agent installer automatically determines whether to install the 32-bit or 64-bit agent based on the machine architecture. No user action is required.

Known Limitations

Cluster Manager and Agent install locations

Electric Cloud does not support installation of Cluster Manager or Agents on the following:

- NFS
- CIFS
- Samba shares

Database location

Electric Cloud does not support storing your database on Network Appliance systems or NFS.

Issues that you may encounter include the following:

- ElectricAccelerator is unable to start MySQL because MySQL attempts to lock files that are already locked.
- If MySQL starts, you may encounter issues related to MySQL UTF-8 support.

Database names

The database name cannot contain a . (period). If the name contains a period, the Cluster Manager will not start.

Hardware Requirements

Electric Cloud recommends installing Cluster Manager, Electric Make, and Electric Agent components on independent machines so each component can run efficiently without competing for system resources.

The following are *minimum* hardware requirements for each machine where ElectricAccelerator components are installed (Agent, Cluster Manager, Electric Make):

- **Processor** - Pentium 4 (Linux and Windows) or SPARC or x86 (Solaris)
- **Installer disk space** - 300 MB, additional component disk space usage varies and depends on the size of your builds.

Agent machines

Agents use system memory to cache small or unmodified files. For optimal performance, each host in the cluster must have enough memory for your link step, which is typically the largest single build step, plus another 200 MB.

If your builds have increased in size substantially since the system was originally delivered, and you have noticed a slowdown in build times, consider upgrading the host hardware.

	Memory	Disk Space
Recommended	2 - 3 GB per agent (if your build contains very large build steps)	Free disk space should be at least 3 - 4 times the size of a complete build (input and output).
Minimum	1 GB per agent (machine minimum of 2 GB)	

Cluster Manager machine

	Memory	Disk Space
Minimum	2 GB	5 GB (to allow space for build log storage)

Electric Make machine

	Memory	Disk Space
Minimum	2 GB	Free disk space should be at least 200 MB or equal to the size of your largest build.

Software Requirements

To connect to Cluster Manager, your web browser must be Microsoft Internet Explorer 7.0 or later, or Mozilla Firefox 3.5 or later.

If you have a web server (for example, Apache, IIS) or other application using port 80 on the Cluster Manager host, you have the following options:

- Uninstall the web server.
- Disable the web server.
- Reconfigure the web server to use another port.
- Choose a non-default port (not 80) for Apache during Cluster Manager installation. The default Apache SSL port is 443.

The Cluster Manager server must not contain a previous MySQL installation. For MySQL, the default port is 3306. If you plan to use an Oracle or MS SQL database, you must create it before installing ElectricAccelerator.

Note: By default, the Cluster Manager server uses port 8030 and secure port 8031.

Your tool chain should be installed on each agent host. You can virtualize the tool chain in some cases, but this can be tricky if the Windows registry is involved. Also, for performance reasons it is better to install the tool chain on each agent host. Many companies use Norton Ghost™ software to mirror all agent hosts.

Cygwin

If you run builds on Windows in Cygwin environments, ensure you have cygwin1.dll version 1.5.25, 1.7.7, or 1.7.9 installed.

Install the same version of Cygwin on all agent hosts and eMake machines. Mixing different Cygwin versions (for example, running v1.5 on an eMake machine and v1.7 on agents) is not supported. (In particular, Cygwin versions 1.5 and 1.7 default to incompatible representations for symbolic links.)

Note: Cygwin version 1.7.x is supported for x = 7 or 9 only. There are known problems for other versions of 1.7.x.

By default, Cygwin 1.7.7 applies overly restrictive permissions to most directories. The permissions prevent the Administrators group from creating new subdirectories and may prevent the agent software from creating new directories to serve as mount points in order to reflect eMake client mount points.

On all agent hosts modify the permissions for the Cygwin installation directory and any other directories under which you want the agent software to dynamically create Cygwin mount points. For agent installations that use standard ECloudInternalUser* accounts, grant the "Administrators" group permission to "create folders / append data." For custom agent users, grant permission for subdirectory creation to those agent users.

Notes to ensure a consistent build environment

Configure Electric Agent hosts identically across a cluster. Ideally, all Agent host machines should be the same, using the same operating system and hardware configuration. You may experience difficulties if you do not follow this recommendation. **In addition**, if any of your tool applications (compilers and so on) are installed locally, ensure they are all the same version and kept in sync.

Note: Ghosting with the Electric Agent already installed is not recommended.

You may experience issues on Windows agent hosts depending on your firewall settings: the ports used to communicate with agents are dynamic.

If you need to lock down communication with an agent host, exclude the following programs from the block list:

- C:\Ecloud\i686_win32\bin\ecagent.exe
- C:\Ecloud\i686_win32\bin\erunnerd.exe

In addition, exclude the following programs during agent host installation:

- C:\ecinst\ecinstconf.exe
- C:\ecinst\eclouduninst.exe
- C:\ecinst\fsize.exe
- C:\ecinst\Setup.exe
- C:\ecinst\SetupErannerTmp.exe
- C:\ERunnerTmp\i686_win32\bin\erunnerdtmp.exe

These programs exist during ElectricAccelerator installation only. You might find it easier to disable communication restrictions with the agent host during agent installation. If you do not want to do this:

- Create folders C:\ecinst and C:\ERunnerTmp\i686_win32\bin
- Move files with those specified names into these directories.
- Exclude these programs from the “blocked programs” list.
- Delete the directories C:\ecinst and C:\ERunnerTmp

Agents can be installed now.

Third-party Supported Build Tools

- GNU Make 3.80 and 3.81
- Microsoft NMAKE 7.x and 8.x
- Symbian Make (Windows)
- Visual Studio .NET 2002, 2003, 2005, 2008, 2010, 2012

Running Visual Studio on Windows Vista

Due to an operating system limitation, ElectricAccelerator does not support Visual Studio 2002 or Visual Studio 2003 on Windows Vista.

Microsoft released Visual Studio 2005 Service Pack 1 and Visual Studio 2005 Service Pack 1 Update for Windows Vista that together address areas of Visual Studio impacted by Windows Vista enhancements. ElectricAccelerator supports Visual Studio 2005 on Windows Vista only with the recommended patch applied.

The full text regarding running Visual Studio on Windows Vista is available from Microsoft at <http://msdn.microsoft.com/en-us/vstudio/aa948853.aspx>.

Note: ElectricAccelerator can support Visual Studio 6 workspaces if they are exported to NMAKE files. However, every time something changes in the workspace, an export must be done again.

- Cygwin 1.5.25, 1.7.7, and 1.7.9
- Apache Ant 1.6.5, 1.7.1, and 1.8.1

Note: For more information on Ant builds, see the ElectricAccelerator Electric Make Users Guide.

- Rational ClearCase 6, 7, 7.1, and 8.0

ElectricAccelerator supports building within ClearCase dynamic views and provides the Ledger feature, which can track files that change as a result of a change to the view's configuration specification. (See "ElectricAccelerator Ledger File" in the *ElectricAccelerator Electric Make Users Guide*.) ClearCase 7 works inside a 64-bit eMake for Linux and Windows.

ClearCase was tested on the following ClearCase versions and patches:

ClearCase servers

ClearCase	ClearCase version 7.0.1 (Wed May 30 17:04:58 EDT 2007) @(#) MVFS version 7.0.1.0 (Wed Apr 11 21:19:21 2007) built at \$Date: 2008-10-24.19:30:48 (UTC) \$
cleartool	7.0.1.0 (Mon Apr 23 21:11:26 EDT 2007)
db_server	7.0.1.0 (Fri Apr 27 21:11:26 EDT 2007)
ClearCase	ClearCase version 2003.06.00 (Tue Dec 16 21:15:58 EST 2003) clearcase_p2003.06.00-55 (Tue May 13 11:49:56 EDT 2008) clearcase_p2003.06.00-56 (Tue May 13 22:14:31 EDT 2008) @(#) MVFS version 2003.06.10+ (Wed Mar 5 21:25:36 2008) VNODE built \$Date: 2008-10-28.18:29:21 (UTC) \$
cleartool	2003.06.10+ (Fri May 2 00:28:18 EDT 2008)
db_server	2003.06.10+ (Fri May 2 00:22:09 EDT 2008)

Linux ClearCase clients

ClearCase	ClearCase version 8.0.0.00 (Mon Sep 19 15:31:35 EDT 2011) (8.0.D110918) @(#) MVFS version 8.0.0.0 (Sat Aug 13 02:28:08 2011) built at \$Date: 2011-08-13.06:34:21 (UTC) \$
cleartool	8.0.0.0 (Tue Sep 6 16:33:44 2011)
db_server	8.0.0.0 (Tue Sep 13 15:37:38 2011)
ClearCase	ClearCase version 7.1.1.1 (Tue Apr 13 15:53:39 EDT 2010) (7.1.1.01.00_2010A.FCS) ClearCase version 7.1.1.2 (Tue May 25 20:46:08 EDT 2010) (7.1.1.02.00_2010B.D100525) ClearCase version 7.1.1.3 (Wed Aug 04 03:03:51 EDT 2010) (7.1.1.03.00_2010C.D100803) @(#) MVFS version 7.1.1.3 (Thu Jun 24 03:49:46 2010) built at \$Date: 2010-10-20.16:49:29 (UTC) \$
cleartool	7.1.1.3 (Tue Jul 6 03:53:32 2010)
db_server	7.1.1.3 (Wed Jun 23 04:05:51 2010)

ClearCase ClearCase version 2003.06.00 (Tue Dec 16 21:15:58 EST 2003)
 clearcase_p2003.06.00-55 (Tue May 13 11:49:56 EDT 2008)
 clearcase_p2003.06.00-56 (Tue May 13 22:14:31 EDT 2008)
 @(#) MVFS version 2003.06.10+ (Wed Mar 5 21:25:36 2008) VNODE built
 \$Date: 2008-04-11.18:51:42 (UTC) \$

cleartool 2003.06.10+ (Fri May 2 00:28:18 EDT 2008)

db_server 2003.06.10+ (Fri May 2 00:22:30 EDT 2008)

Windows ClearCase clients

ClearCase ClearCase version 8.0.0.00 (Mon Sep 19 15:31:35 EDT 2011) (8.0.D110918)
 @(#) MVFS version 8.0.0.0 (Tue Sep 6 17:52:12 2011)

cleartool 8.0.0.0 (Tue Sep 6 19:31:30 2011)

db_server 8.0.0.0 (Tue Sep 13 15:51:55 2011)

ClearCase ClearCase version 7.1.1.1 (Tue Apr 13 15:53:39 EDT 2010) (7.1.1.01.00_
 2010A.FCS)
 ClearCase version 7.1.1.2 (Tue May 25 20:46:08 EDT 2010) (7.1.1.02.00_
 2010B.D100525)
 ClearCase version 7.1.1.3 (Wed Aug 04 03:03:51 EDT 2010) (7.1.1.03.00_
 2010C.D100803)
 @(#) MVFS version 7.1.1.3 (Tue Jun 29 04:03:33 2010)

cleartool 7.1.1.3 (Tue Jul 6 05:54:49 2010)

db_server 7.1.1.3 (Wed Jun 23 11:45:39 2010)

ClearCase ClearCase version 2003.06.00 (Fri Apr 18 13:06:18 2003)
 clearcase patch p2003.06.01 (Fri Sep 5 11:59:28 2003)
 clearcase patch p2003.06.12 (Fri Nov 7 11:59:28 2003)
 clearcase patch p2003.06.14 (Fri Oct 8 11:59:28 2004)
 clearcase patch p2003.06.15 (Fri Jun 9 11:59:28 2005)
 clearcase patch p2003.06.15 (Fri Sep 9 11:59:28 2005 PATCH 2005C)
 clearcase patch p2003.06.15 (Fri Nov 9 11:59:28 2005 PATCH 2005D)
 clearcase patch p2003.06.15 (Fri Jan 6 11:59:28 2006 PATCH 2006A)
 clearcase patch p2003.06.15 (Fri Apr 7 11:59:28 2006 PATCH 2006B)
 @(#) MVFS version 2003.06.10+ (Thu Jan 27 04:26:46 2005)

cleartool 2003.06.10+ (Mon Feb 27 08:26:52 2006)

db_server 2003.06.10+ (Fri May 6 22:12:48 2005)

Solaris ClearCase clients

ClearCase ClearCase version 7.0.1 (Wed May 30 16:29:44 EDT 2007)
 @(#) MVFS version 7.0.1.0 (Wed Apr 11 21:42:52 2007)

cleartool 7.0.1.0 (Tue May 15 10:10:24 EDT 2007)

db_server 7.0.1.0 (Tue May 15 10:08:56 EDT 2007)

ClearCase ClearCase version 2003.06.00 (Fri Apr 25 09:05:22 EDT 2003)
 clearcase_p2003.06.00-55 (Tue May 13 11:49:56 EDT 2008)
 @(#) MVFS version 2003.06.00 (Thu Jan 30 20:47:56 EST 2003)

cleartool 2003.06.10+ (Fri May 2 00:01:35 EDT 2008)

db_server 2003.06.10+ (Thu May 1 23:56:35 EDT 2008)

Note: ElectricAccelerator does not currently support ClearCase integration on Solaris x86. If you have need of this support, please contact your Electric Cloud sales representative. You can make use of `ecclearcase_fake.so` to provide information about your ClearCase setup through an ini file. See “eMake’s “fake” interface for ClearCase” in Chapter 6 of the *ElectricAccelerator Electric Make Users Guide*.

Database Support

ElectricAccelerator supports the following databases:

- MySQL 5.5.16 (ElectricAccelerator default)
- Oracle 11g Release 2
- MS SQL Server 2008

The "checksum" utility

An MD5 checksum file is available on the Electric Cloud FTP site. If you choose to verify that ElectricAccelerator files are intact and unaltered from their original form and content after you download them, download the corresponding MD5 checksum file also.

MD5 utilities are available for supported operating systems.

- On Linux, verify with `md5sum --check md5.txt`
- Most Linux installations provide an `md5sum` command for calculating MD5 message digests.
- An MD5 utility for Windows can be downloaded at <http://www.fourmilab.ch/md5/>.

Default Installation Directories

- Windows
`C:\ECloud\i686_win32`
- Linux
`/opt/ecloud/i686_Linux`
- Solaris SPARC
`/opt/ecloud/sun4u_SunOS`
- Solaris x86
`/opt/ecloud/i686_SunOS5.10`

Installation Log Locations

The installation log file is created in the install directory’s root. The name is `install_$timestamp.log`.

The default location:

- Windows
`C:\ECloud`
- Linux and Solaris
`/opt/ecloud`

Agent Log Locations

Agent log files’ default location:

- Windows
`<ECloud Install>\ecagent?.log`

- Linux and Solaris
/var/log/ecagent?.log

Disk Cache Directory and Agent Temporary Storage Location

The same location is used for the disk cache directory and agent temporary storage. By default, the location is:

- Windows
C:\WINDOWS\Temp
- Linux and Solaris
/tmp

The most common reason to change this location is due to insufficient disk cache space.

To change the disk cache directory or agent temporary storage location:

Use the agent's `ecconfig` command:

```
ecconfig -tempdir <newtempdir>
```

Use the `clusterexec` command (run on either the Cluster Manager or the build server) to change all agents at once:

```
clusterexec --cm=<YOURCM> ecconfig -tempdir <newtempdir>
```

Notes:

1. You must specify a full `PATH` to the directory you want to use. Each agent on the host creates a unique subdirectory within the disk cache/temporary storage location, so they do not conflict with each other.
2. The path specified for `-tempdir` must already exist. On Linux/UNIX, the path must have permissions that allow build users to create temporary files. On Windows, agent users must have permission to create temporary files because the eMake client user is not impersonated.

You must restart the agents for your changes to take effect:

```
clusterexec --cm=<YOURCM> restartAgent
```

Note: After specifying a different disk cache/temporary storage location, you may switch back to the default location listed at the top of this section. To do this, use `ecconfig -tempdir` with an empty string [""] as the location.

Known Linux Kernel Issue and ElectricAccelerator Performance

Affected kernel versions

- RHEL versions later than 2.6.18-194.32 and earlier than 2.6.32-131.
- Ubuntu versions 2.6.31, 32, 33, and 34

Symptoms

Affected systems may encounter reduced performance on both ext3 and ext4 filesystems. Symptoms may include:

- `hung_task_timeout_secs` messages in system `dmesg` logs
- widely variable agent availability (entering and exiting agent “penalty” status frequently)
- contention over the `ecagent.state` file
- slower builds (with unexplained variances)

To help determine if your environment has this kernel issue, run

```
dmesg | grep hung_task_timeout
```

If `hung_task_timeout` errors are present, this indicates that a known Linux kernel issue is present. Contact your kernel provider to obtain a different version of the precompiled kernel.

Corrective actions

For systems running RHEL 5.6, 5.7, 5.8, and 6.0

Consider upgrading to 2.6.32-131 (RHEL 6.1), or downgrading to 2.6.18-194.32 (RHEL 5.5).

For systems running Ubuntu 10.04

Consider upgrading to kernel version 2.6.35 or later.

To install the upstream kernel, do the following, for example:

```
sudo add-apt-repository ppa:kernel-ppa/ppa
sudo aptitude update
sudo apt-get install linux-image-2.6.35-15-generic
sudo apt-get install linux-headers-2.6.35-15-generic
sudo apt-get install linux-maverick-source-2.6.35 linux-maverick-headers-2.6 (optional)
```

Upgrading the kernel may require you to reinstall various kernel modules such as video drivers, `efs`, and `vm-tools`. You can do so by running: `apt-get dist-upgrade` and then rebooting.

Planning for Grid Use

If you plan to use a host grid in conjunction with ElectricAccelerator, here is a planning checklist:

- Choose a machine for Cluster Manager.

This machine will use the grid as any other client-only host. If you are upgrading an existing Cluster Manager, you must install the grid client-only product—and ensure it is 32-bit only.

IMPORTANT: The Solaris x86 Cluster Manager cannot integrate with LSF.

- Choose hosts within the grid to run your builds.

In general, these are machines you already use for software builds. ElectricAccelerator will use these machines in parallel to run builds, but machine availability is controlled by the grid management software. Electric Make allows you to select a subset of these machines using grid resource query strings. (See the *ElectricAccelerator Electric Make Users Guide*, “Electric Make Command-line Options and Environment Variables” chapter for information on the `emake --emake-resource= parameter`.)

- Choose machines for Electric Make.

If you will be running Electric Make within the grid, you must install Electric Make on those hosts.

- Select or create a user ID to run Cluster Manager.

IMPORTANT: This user ID must be set up with appropriate permissions and able to execute grid commands.

Note: If this is a UNIX user ID, you may need to run, for example, “`profile.lsf`” in your shell startup script.

If this is a Windows user ID, you may need to go to Control Panel > Administrative Tools > Services. Right-click the ElectricAccelerator Cluster Manager service and go to Properties. Click the Log On tab. Change Local System Account to This account. Type-in the user ID and password for the user account.

- Select a queue for ElectricAccelerator.

This queue must be allowed to submit jobs that request exclusive host use (`bsub -x` option) or a queue that specifies `HJOB_LIMIT=1`. Set up the queue so jobs can be submitted on any platform where you intend to use Electric Make. For example, if your software builds are on Linux and Windows, the queue must allow jobs on either platform.

Chapter 3: Installing ElectricAccelerator

General Information

This chapter describes all versions of the ElectricAccelerator installation process.

If you are upgrading a previously installed Accelerator version, [Upgrading ElectricAccelerator Components](#)

You can install the software using a graphical user interface or an interactive command-line interface (for Linux and Solaris), or by performing a non-interactive command-line installation. All methods are documented in this chapter.

The three methods for installing Accelerator are:

- **User Interface** - This is a set of installation screens that provides click-through automation and prompts for information you need to supply.
- **Command-line** - Use this method if you prefer using an interactive command-line for the installation process. This method is available on UNIX machines only.
- **Silent** - This is a non-interactive command-line installation. You may find this installation method preferable for installing multiple Electric Agents.

Read the procedures thoroughly *before* attempting to install, configure, or uninstall any component.

Notes:

1. Installing any component completely replaces the current installation on that machine. Only one ElectricAccelerator installation can exist on a machine and it is whatever combination of components were chosen from the most recent install. For example, installing the Cluster Manager on a machine with the Electric Agent, results in a machine with Cluster Manager only, not a machine with the Electric Agent and Cluster Manager.
2. If you plan to install Electric Agent on a machine that has ElectricAccelerator Developer Edition installed, Electric Cloud recommends doing a full uninstall of ElectricAccelerator Developer Edition first.

Installation Notes

Installing from the Cygwin shell

If you choose to run the installer from the Cygwin shell, be advised of the following:

- Before running the installer, disable UAC, or start the Cygwin shell using the Run as Administrator menu item (right-click the Cygwin desktop icon). Running the installer with UAC enabled may result in a "permission denied" error. This is applicable for all Windows versions that use UAC (Windows 7,

Windows Server 2008 R2, and Windows Vista).

- You may encounter issues when running the installer from the Cygwin `/tmp` directory. Electric Cloud recommends running the installer from a different directory. This is applicable for all Windows versions.

Antivirus software

Some antivirus software may affect the installer. Turn antivirus software off during installation. If antivirus software is running when you start the installer, you may receive an error dialog. The antivirus software may have reported ElectricAccelerator files as a virus and removed them from the temp location. As a workaround, turn off the antivirus software and rerun the installer.

umask

Electric Cloud recommends umask 0022. Do not set a different umask during installation.

Installing from a network drive

On Windows, if you invoke the installer from a network drive, you may receive an Unknown Publisher security warning. You can disregard this warning and proceed with installation.

Absolute path for Solaris

On Solaris systems, you must use an absolute path for the installation directory.

mysqlcheck

By default, mysqlcheck looks for my.cnf in `/etc` and `~/.my.cnf` for configuration information. The installer does not currently copy my.cnf to `/etc`, causing mysqlcheck to fail. You must add the following to the mysqlcheck command line parameter: `--defaults-file=<path to my.cnf>`

For example: `mysqlcheck --defaults-file=/opt/eccloud/i686_Linux/mysql/my.cnf -o eccloud`

User Interface Installation Method

Cluster Manager

Cluster Manager is a web server and can be installed on any networked machine with access to Electric Make and the cluster hosts, or on the main build machine where Electric Make is installed.

Note: Install only one copy of Cluster Manager, regardless of the number of hosts in the cluster or the number of machines with Electric Make installed. Electric Make is installed by default when Cluster Manager is installed.

If you are overwriting a previously installed ElectricAccelerator version, follow upgrading instructions in [Upgrading ElectricAccelerator Components](#) before attempting to install the new version.

IMPORTANT: On Windows Server 2008 R2, you may encounter a “Not Responding” error during install. Electric Cloud recommends waiting for up to 10 minutes for the install to continue normally.

Installing Cluster Manager

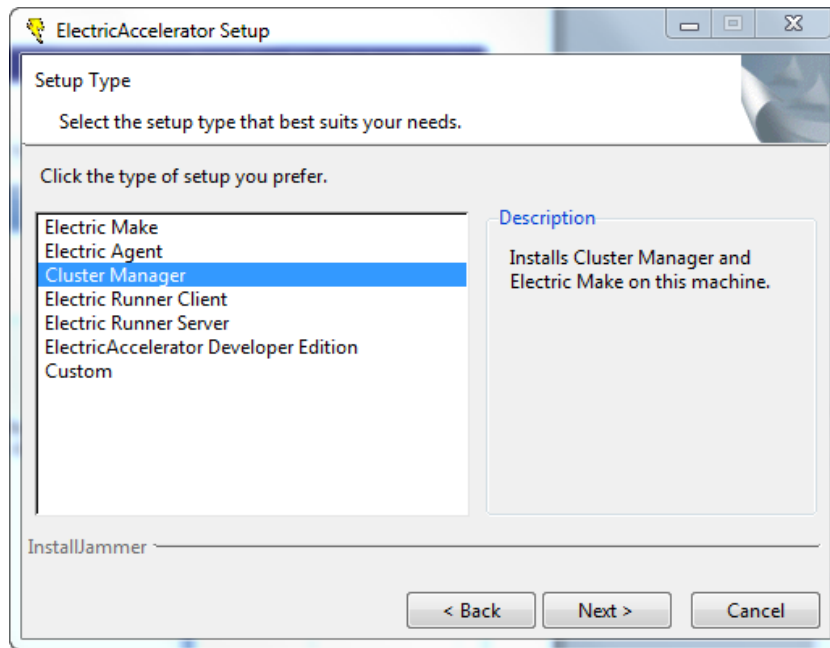
Make sure you log in as root or Administrator (You *must* be a member of the Administrator group—Administrator privileges are not sufficient.)

Note: On Windows, if you are running `rdp` on the target Cluster Manager server, ensure `rdp` is in installation mode: `change user/install`.

1. Double-click the `ElectricAccelerator-<version>` installer file to start installation. (For Windows systems running Vista or later, the administrator user must right-click the installer and select Run as administrator.)

Note: It may take a few minutes to extract the installation packages to your machine before you see the installation wizard. Also, during installation, if you see a Windows security alert pop-up, click **unblock** and continue.

2. For Windows, ElectricAccelerator requires the Microsoft Visual C++ 2005 Redistributable. If it is already installed, select the checkbox. Click **Next** to continue.
3. When the Welcome screen appears, click **Next** to continue to the Setup Type screen.
4. Select Cluster Manager (Electric Make also installs automatically with Cluster Manager).

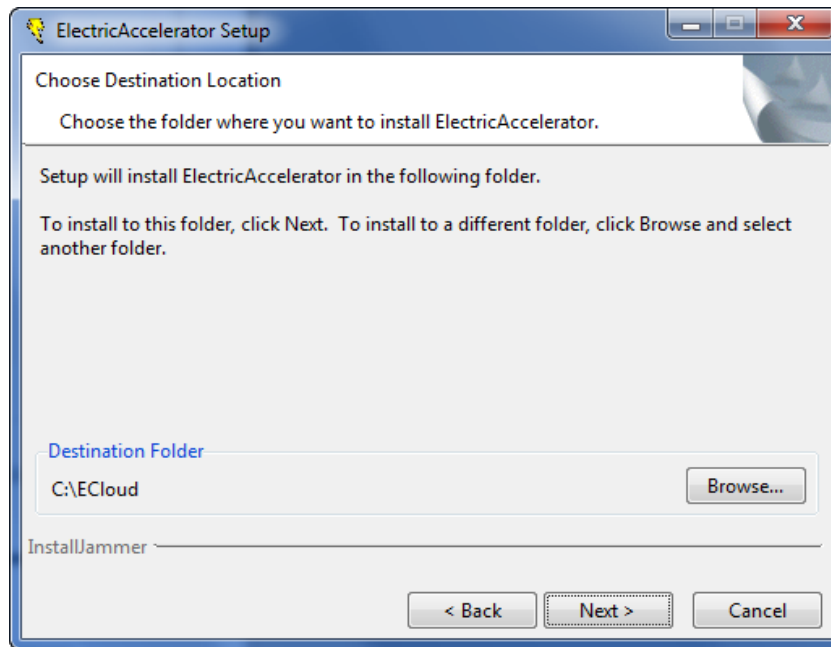


Click **Next** after making your selection.

5. On the Choose Destination Location screen, accept the default installation directory (C:\ECloud for Windows or /opt/eccloud for Linux and Solaris) or browse to select an alternative directory.

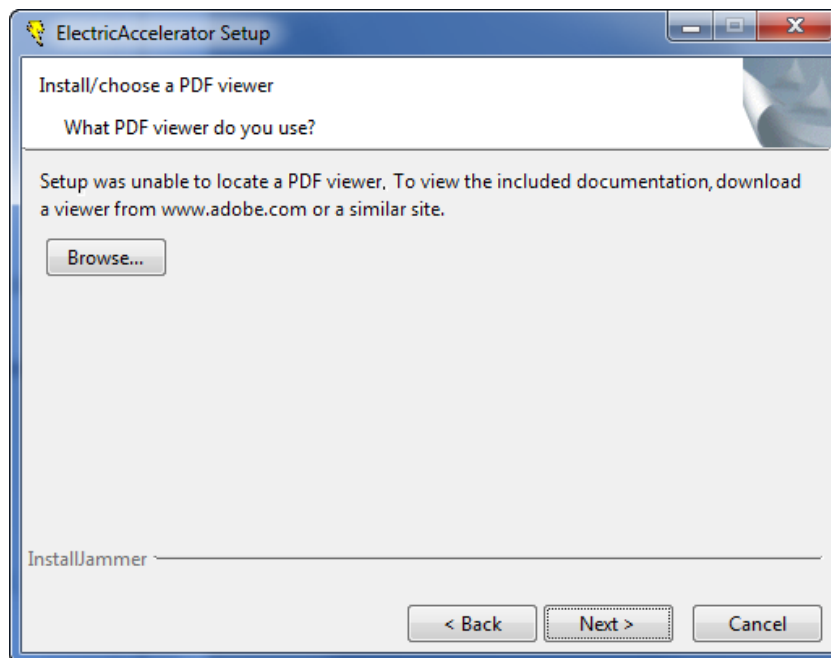
If you want to install Electric Make in a non-default location, use the same location on all agent hosts and on the Cluster Manager host.

Note: Avoid selecting an alternative directory that includes spaces in the name. Spaces can create problems when configuring connections with other command-line-based components.



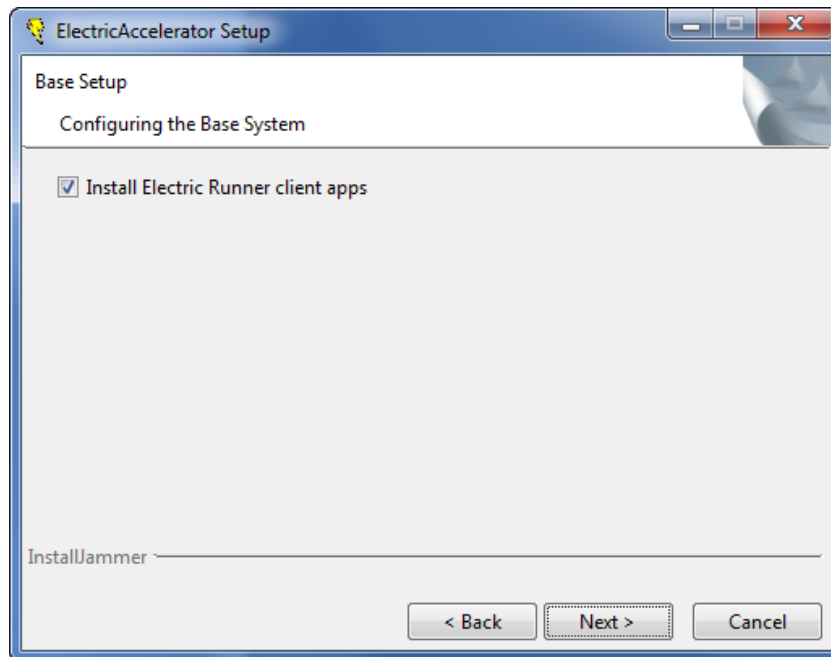
Click **Next**.

6. If the installer cannot locate a PDF viewer, it will prompt you to browse for one. If you wish, browse for a PDF viewer to use.



Click **Next**.

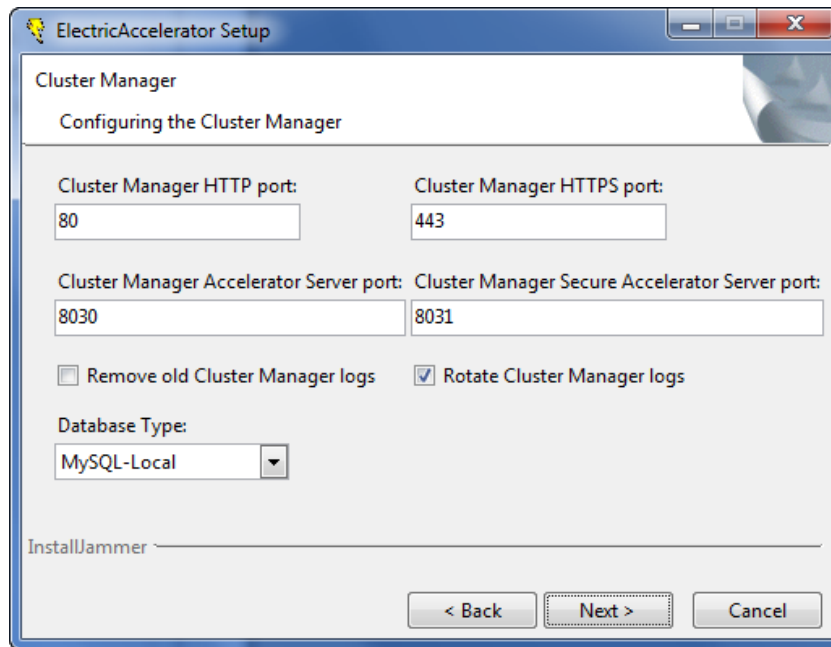
7. On the Base Setup screen, choose whether you want to install Electric Runner client applications.



Click **Next**.

8. On the Cluster Manager screen, enter Cluster Manager configuration options:
 - Accept the default ports or type-in alternative port numbers if the defaults are already in use. For example, IIS and Apache also use port 80 by default.
 - Accept the default to keep old log files or select the checkbox to remove old log files.
 - Accept the default [checked checkbox] to leave web server log rotation turned on or clear the checkbox to turn off log rotation. **Important** - Log rotation may affect Cluster Manager performance.
 - Select the type of database to use.

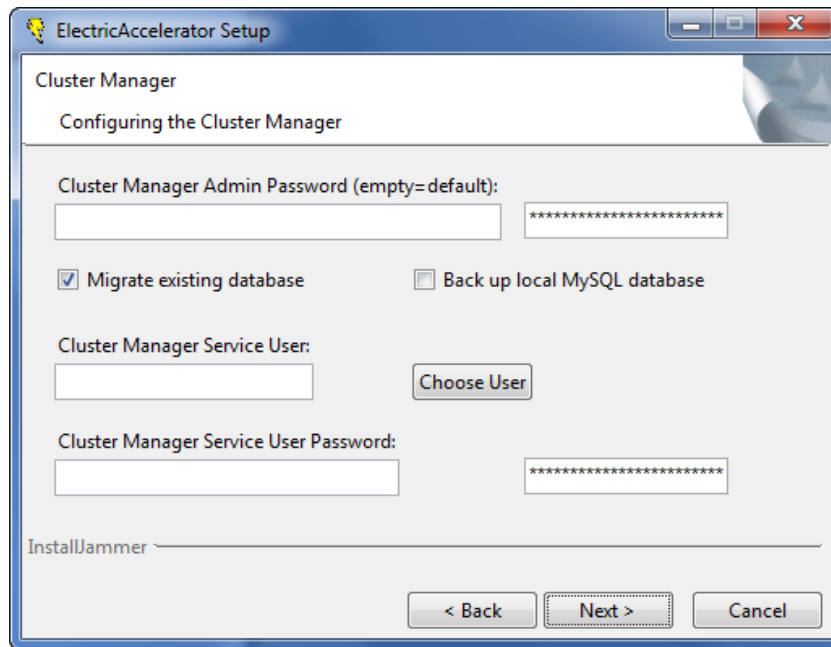
Note: If you do not use the default local MySQL database, the database must already exist and you must be able to connect to it. The installer does not validate database connections.



Click **Next**.

9. On the next Cluster Manager screen, continue to enter Cluster Manager configuration options:
 - You can change the Cluster Manager admin password or accept the default password (*changeme*).
 - You do not need to migrate the existing database or back up the local MySQL database if this is your first Cluster Manager installation.
 - Fill-in the user name or choose a user that the Cluster Manager service should run as or leave blank. Choose a password for the Cluster Manager service user or leave blank.

Note: If you plan to use a host grid, the user selected for the Cluster Manager Service User must be fully enabled to start and stop grid jobs.

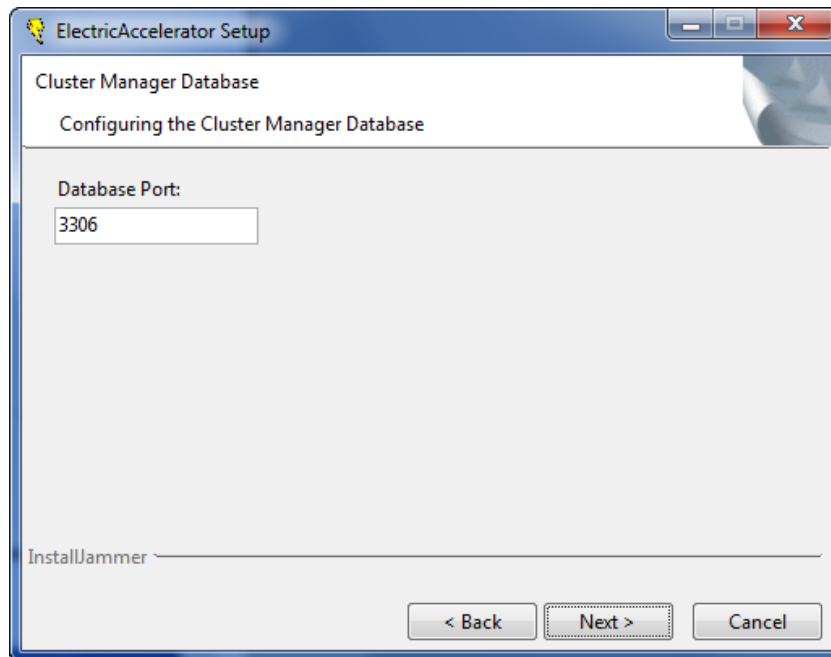


Click **Next**.

10. On the Cluster Manager Database screen, type-in the Cluster Manager database port number. For MS SQL, use port 1433.

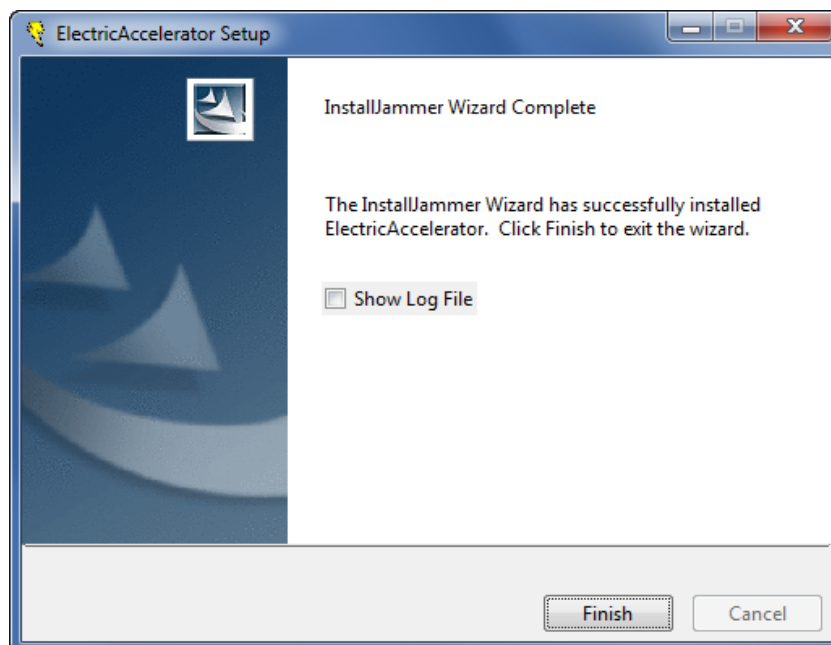
If you did not select MySQL-Local as your database type, the following additional fields are available:

- Database Host/IP Address
- Database User
- Database Password
- Database Name (Do not use a . (period) in the name.)



Click **Next**.

11. When the Start Copying Files screen appears, click **Next**.
12. The Installing screen displays while installation is completing. When installation is finished, the Complete screen displays.



Click **Finish**.

Installation is complete.

The installation log file is in the install directory's root, `C:\ECloud` or `/opt/eccloud`, by default.

If Apache fails to start properly after a fresh Cluster Manager install, reboot the system.

Electric Agent/EFS

Electric Agent/EFS software must be installed on each machine in the cluster. Electric File System (EFS) and Electric Agent (collectively, the Agent) are installed simultaneously on each host machine during Electric Agent/EFS installation. Multiple instances of Electric Agent/EFS pairings can be set up to run in parallel on a host. For example, if you have a multiprocessor machine, you may want to set up an Agent for each processor. More information on running multiple Agents is available in the ElectricAccelerator online help.

Follow upgrading instructions in [Upgrading ElectricAccelerator Components](#) if you are upgrading a previous Electric Agent installation.

Your tool chain should be installed on each agent host. You can virtualize the tool chain in some cases, but this can be tricky especially when the Windows registry is involved. For performance reasons, it is better to install the tool chain on each agent host. Many companies use a “ghosting” software for this purpose. **IMPORTANT:** “Ghosting” with the Electric Agent already installed is **not** recommended.

Additional information for Windows

For Windows Vista and Windows Server 2008 R2, the installer *automatically* does the following:

- Disables the Windows error reporting service.
- Sets HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate to 0. The default value for Windows Vista and Windows Server 2008 R2 is 1.
- Disables User Account Control (UAC) for 64-bit versions. Disabling UAC avoids popup windows for applications that require administrator privileges. If UAC is enabled, application registry access is redirected to each user’s virtual store, even if it runs under the Administrator account.

Note: If Symantec AntiVirus software is installed on your intended agent machines, disable it before installing Electric Agent/EFS to avoid serious filesystem conflicts. If Symantec AntiVirus cannot be disabled, put C:\ECloud in an exclusion list or disable the AutoProtect feature. Electric Cloud generally considers agent machines safe because they are internal machines residing behind a firewall. If you need more information, contact Electric Cloud technical support.

Installing Electric Agent/EFS

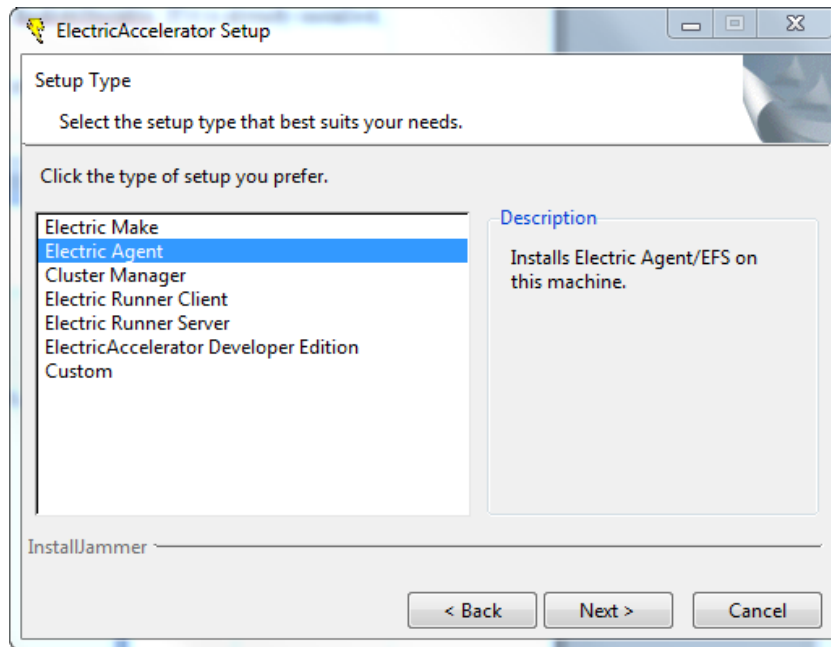
Make sure you log in as root or Administrator (You *must* be a member of the Administrator group—Administrator privileges are not sufficient.)

Note: On Windows, if you are running *rdp* on this host, ensure *rdp* is in installation mode: `change user/install`.

1. Double-click the `ElectricAccelerator-<version>` installer file to start installation. (For Windows systems running Vista or later, the administrator user must right-click the installer and select Run as administrator.)

Note: It may take a few minutes to extract the installation packages to your machine before you see the installation wizard. Also, during installation, if you see a Windows security alert pop-up, click **unblock** and continue.

2. For Windows, ElectricAccelerator requires the Microsoft Visual C++ 2005 Redistributable. If it is already installed, select the checkbox. Click **Next** to continue.
3. When the Welcome screen appears, click **Next** to continue to the Setup Type screen.
4. On the Setup Type screen, select **Electric Agent**.



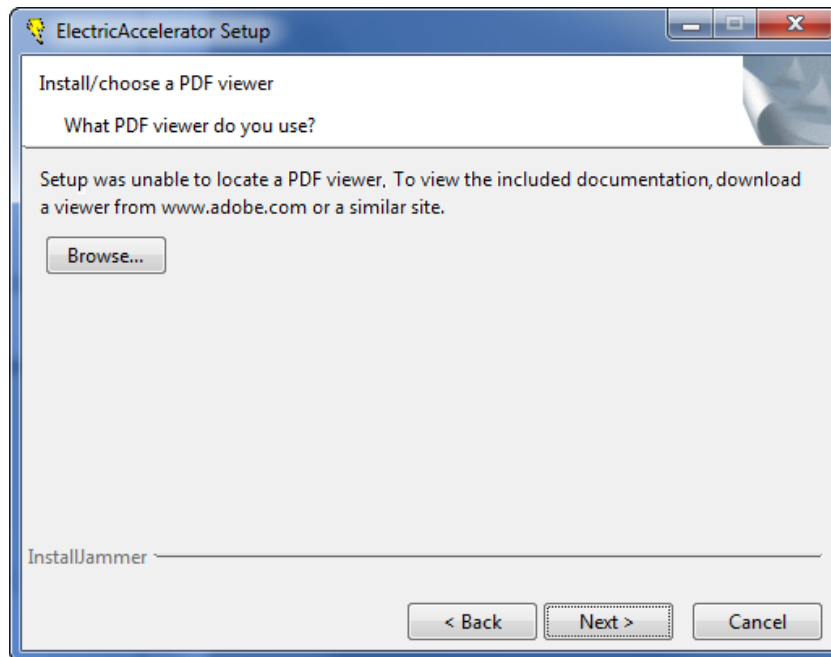
Click **Next**.

5. On the Choose Destination Location screen, accept the default installation directory (C:\ECloud for Windows or /opt/eccloud for Linux and Solaris) or browse to select an alternative directory.

Note: Avoid selecting an alternative directory that includes spaces in the name. Spaces can create problems when configuring connections with other command-line-based components.

Click **Next**.

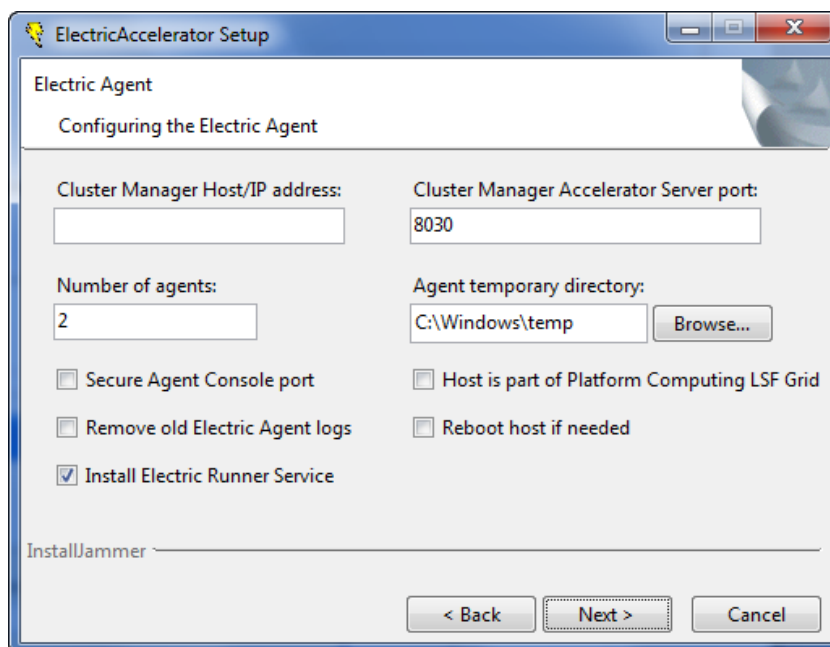
6. If the installer cannot locate a PDF viewer, it will prompt you to browse for one. If you wish, browse for a PDF viewer to use.



Click **Next**.

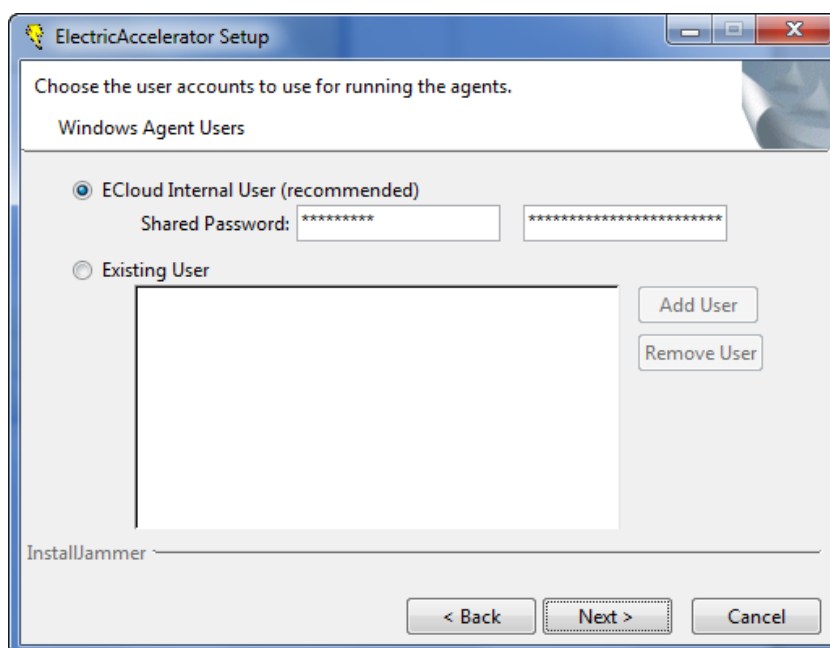
7. On the Base Setup screen, choose whether you want to install Electric Runner client applications, and click **Next**.
8. On the Electric Agent screen, enter options for the locally installed Electric Agent/EFS.
 - Type-in the Cluster Manager host name or IP address.
 - Accept the default Cluster Manager port 8030 or type-in the alternative port you supplied during the Cluster Manager server installation.
 - Enter the number of Agents to run—the installer calculates the default number of Agents based on the effective number of CPUs on the machine. Default=1, if one CPU is present. If more than one CPU is present, the default value is the number of effective CPUs. The maximum number of Agents allowed on one 32-bit machine is 32; the maximum for one 64-bit machine is 64.
 - Accept the default agent temporary directory, or choose an alternative.
 - For the Secure Agent Console Port checkbox, accept the default if you do not plan to use the secure port, or select the checkbox if you want to use the secure port.
 - If this agent machine is part of the grid, select the Host is part of Platform Computing LSF Grid checkbox. (For this field, the term “LSF” refers to all grid engines.
 - Accept the default to keep all existing log files or select the checkbox to Remove existing log files.
 - Select the checkbox to Reboot host if needed.
 - Accept the default to install the Electric Runner service, or clear the checkbox to not install this service.

Note: eRunner is a “helper” service normally installed on all agent machines. This service listens for commands to execute on agents, facilitating administration tasks. For example, with eRunner you can upgrade all agents at the same time, remotely.



Click **Next**.

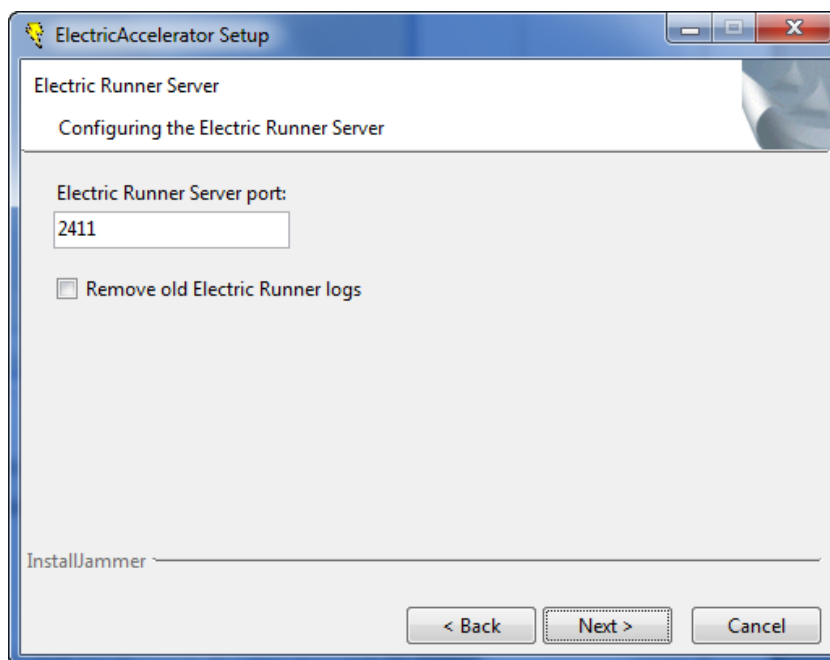
9. On Windows, select the user accounts you will use to run agents. Accept the default ECloud Internal User or select Existing User. If you select Existing User, click **Add User** to add a user account.



Click **Next**.

10. On the Electric Runner Server screen, enter options for the Electric Runner server.

- Accept the default Electric Runner Server port, or type-in an alternative.
- Accept the default to leave old Electric Runner logs, or select the checkbox to remove old logs.



Click **Next**.

11. When the Start Copying Files screen appears, click **Next**.
12. The Installing screen displays while installation is completing. When installation is finished, the Complete screen displays.

Click **Finish**.

Installation is complete.

The installation log file is in the install directory's root, C:\ECloud or /opt/eccloud, by default.

Note: Installing ElectricAccelerator on Windows systems may unset the environment variable JAVA_HOME. Reset JAVA_HOME manually.

Electric Make

Electric Make can be installed on a networked machine or on a shared file server accessible to other build machines. Wherever Electric Make is installed, it must have access to the Cluster Manager machine and Agents on the cluster host machines. After Electric Make is installed, configure Electric Make for your build environment and enable communication with Cluster Manager. For information on enabling communication with Cluster Manager, see “Configuring ElectricAccelerator” in the *ElectricAccelerator Electric Make Users Guide*.

Follow upgrading instructions in [Upgrading ElectricAccelerator Components](#) if you are upgrading a previous Electric Make installation.

Installing Electric Make

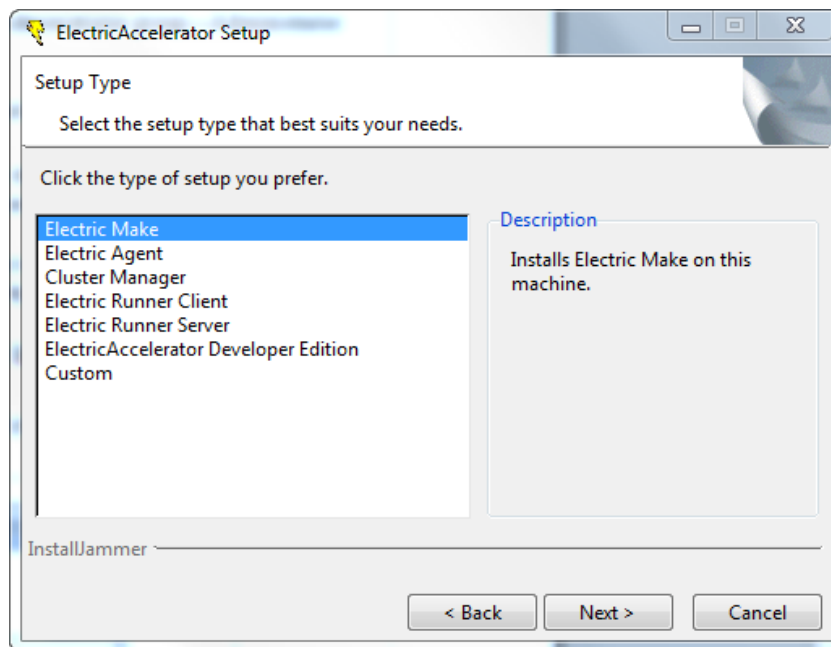
Make sure you log in as root or Administrator (You *must* be a member of the Administrator group—Administrator privileges are not sufficient.)

Note: On Windows, if you are running *rdp* on this host, ensure *rdp* is in installation mode: `change user/install`.

1. Double-click the `ElectricAccelerator-<version>installer` file to start installation. (For Windows systems running Vista or later, the administrator user must right-click the installer and select Run as administrator.)

Note: It may take a few minutes to extract the installation packages to your machine before you see the installation wizard. Also, during installation, if you see a Windows security alert pop-up, click **unblock** and continue.

2. For Windows, ElectricAccelerator requires the Microsoft Visual C++ 2005 Redistributable. If it is already installed, select the checkbox. Click **Next** to continue.
3. When the Welcome screen appears, click **Next** to continue to the Setup Type screen.
4. On the Setup Type screen, select **Electric Make**.



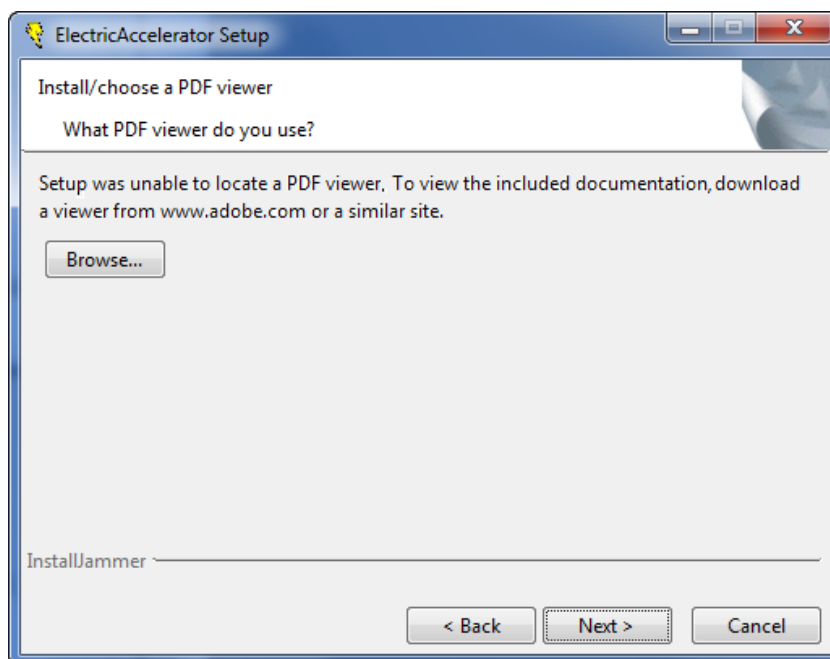
Click **Next**.

5. On the Choose Destination Location screen, accept the default installation directory (`C:\ECloud` for Windows or `/opt/eccloud` for Linux and Solaris) or browse to select an alternative directory. Ensure the Electric Make path matches the path used for the installed Agent.

Note: Avoid selecting an alternative directory that includes spaces in the name. Spaces can create problems when configuring connections with other command-line-based components.

Click **Next**.

6. If the installer cannot locate a PDF viewer, it will prompt you to browse for one. If you wish, browse for a PDF viewer to use.



Click **Next**

7. On the Base Setup screen, choose whether you want to install Electric Runner client applications, and click **Next**.
8. When the Start Copying Files screen appears, click **Next**.
9. The Installing screen displays while installation is completing. When installation is finished, the Complete screen displays.

Click **Finish**.

Installation is complete.

The installation log file is in the install directory's root, `C:\ECloud` or `/opt/eccloud`, by default.

Note: Installing ElectricAccelerator on Windows systems may unset the environment variable `JAVA_HOME`. Reset `JAVA_HOME` manually

Additional Install Options

The installer has the following additional install options:

- **Electric Runner Client** - The Electric Runner Client component installs platform-independent tools (`clusterexec`, `clusterupload`, and `clusterdownload`) that allow you to perform operations across all hosts simultaneously. The tools allow you to:
 - Start and stop agents
 - Reboot hosts
 - Run commands on hosts
 - Upload files
 - Download files

- Electric Runner Server - The Electric Runner Server component installs the server side (erunnerd) 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.
- Custom - A custom install allows you to install multiple components at once.

Note: The online help system contains additional information about Electric Runner. Go to the Cluster Manager Administration Tools section in the ElectricAccelerator Installed Tools/Files topic.

The installation of all ElectricAccelerator components is now complete. If you do not plan to use the Grid Host Manager, [Initial Configuration Tasks](#) to complete several tasks before you begin using the product.

If you are planning to use the Grid Host Manager, follow the instructions in [Integrating ElectricAccelerator with the Grid](#) before proceeding to [Initial Configuration Tasks](#).

Interactive Command-line Installation Method

Cluster Manager

Cluster Manager is a web server and can be installed on any networked machine with access to Electric Make machines and cluster hosts, or on the main build machine where Electric Make is installed.

Note: Install only one copy of Cluster Manager, regardless of the number of hosts in the cluster or the number of machines with Electric Make installed. Electric Make is installed by default when Cluster Manager is installed.

If you are overwriting a previously installed ElectricAccelerator version, [Upgrading ElectricAccelerator Components](#) before attempting to install the new version.

Installing Cluster Manager

1. Obtain a copy of the installer (ElectricAccelerator-<version>).
2. Log in as root.
3. Run `chmod +x` on the installer to ensure it is executable.
4. Run `./<installer filename> --mode console` to start ElectricAccelerator installation.
5. When the welcome message displays, enter **3** to select the **Cluster Manager** package. Press Enter. For example,

```
Please enter the appropriate number to choose the package:
1 - Electric Agent/EFS
    Installs Electric Agent/EFS on this machine.
2 - Electric Make
    Installs Electric Make on this machine.
3 - Cluster Manager
    Installs Cluster Manager and Electric Make on this
4 - Electric Runner Client
    Installs Electric Runner Client component to run commands on the cluster
.
5 - Electric Runner Server
    Installs Electric Runner Server to execute cluster commands on the local
machine.
7 - ElectricAccelerator Developer Edition
    Installs ElectricAccelerator Developer Edition on this machine.
```

After choosing option 3, provide the following Cluster Manager configuration information. Accept the defaults or type-in alternatives.

Notes:

1. Turning on log rotation may affect Cluster Manager performance.
2. If the installer cannot locate a PDF viewer, it will prompt you to provide a path to one.

```
Where do you want to install ElectricAccelerator? [opt/ecloud]
Install Electric Runner client apps? [y/n]Cluster Manager HTTP port: [80]
Cluster Manager HTTPS port: [443]Cluster Manager Accelerator Server port: [8030]
Cluster Manager Secure Accelerator Server port: [8031]
Remove old logs [y/N]
Rotate Cluster Manager logs [y/N]
Database Type: [MySQL-Local]
Cluster Manager Service User: [eacmuser]
Cluster Manager admin password: [changeme]
Migrate existing database [y/N]
Backup local MySQL database [y/N]
Database port: [0]
Database Host/IP address: [localhost]
Database User: [ecloud]
Database Password: [ecloud]
Database Name: [ecloud]
```

Note: Do not use a . (period) in the database name.

The installer installs Cluster Manager using the configuration details you entered, followed by “Installation complete” when the install completes.

The installation log file is in the install directory’s root, `/opt/ecloud` by default.

Note: Linux operating systems contain a 1024 default file descriptor limit for each process. However, this does not mean 1024 builds can be run concurrently. Depending on the size of your builds, you may be able to run only 300-400 builds concurrently because of other file descriptor activity and the collection activity between eMake and the Cluster Manager. If you must run more than 300-400 builds simultaneously, increase the default file descriptor limit.

If Apache fails to start properly after a fresh Cluster Manager install, reboot the system.

Installing Cluster Manager for a grid environment

Cluster Manager is the only ElectricAccelerator component that communicates directly with the host grid.

IMPORTANT: The user ID used to install Cluster Manager and the user ID selected for the Cluster Manager Service User must be fully enabled to start and stop grid jobs.

The Cluster Manager installation process is the same whether or not you plan to use a host grid.

Electric Agent/EFS

Electric Agent/EFS software must be installed on each machine in the cluster. Electric File System (EFS) and Electric Agent (collectively, the Agent) are installed simultaneously on each host machine during Electric Agent/EFS installation. Multiple instances of Electric Agent/EFS pairings can be set up to run in parallel on a host. For example, if you have a multiprocessor machine, you may want to set up an Agent for each processor. More information on running multiple Agents is available in the ElectricAccelerator online help.

Note: Agent/EFS builds a kernel module during installation, so you may need to take this into consideration.

Follow upgrading instructions in [Upgrading ElectricAccelerator Components](#) if you are upgrading a previous Electric Agent installation.

IMPORTANT: For RHEL 5, you must install the `kernel-devel` package version that matches the Linux kernel into which the modules will be loaded.

Installing Electric Agent/EFS

1. Obtain a copy of the installer (ElectricAccelerator-<version>).
2. Log in as root.
3. Run `chmod +x` on the installer to ensure it is executable.
4. Run `./<installer filename> --mode console` to start ElectricAccelerator installation.
5. When the welcome message displays, enter **1** to select the **Electric Agent/EFS** package. Press Enter.

After choosing option 1, provide Electric Agent/EFS configuration information. Accept the defaults or type-in alternatives.

- Use the Cluster Manager default server listening port 8030, or type-in the alternative port if you did not accept the default when you installed Cluster Manager.
- Select the number of Agents to run.
The installer calculates the default number of Agents based on the number of effective CPUs on the machine. For example, you may have one Agent per virtual CPU, a single dual-core CPU supports two Agents, three dual-cores support six Agents. The maximum number of Agents allowed on one 32-bit machine is 32; the maximum for one 64-bit machine is 64.
- Specify the Agent temporary directory. The default is `/tmp`. If you specify a different directory, it must already exist, otherwise the temporary directory defaults to `/tmp`.
- Regarding host grid usage
In a grid environment, such as LSF, Agents are not bound to a Cluster Manager during installation. Instead, the grid management software tells the agents which Cluster Manager to talk to—which means the Cluster Manager host name or port number you may have specified will not be used.
- Decide whether to install Electric Runner.
eRunner is a “helper” daemon normally installed on all agent machines. This daemon listens for commands to execute on agents, facilitating administration tasks. For example, with eRunner you can upgrade all agents at the same time, remotely.

Note: If the installer cannot locate a PDF viewer, it will prompt you to provide a path to one.

The installer installs Electric Agent/EFS using the configuration details you provided, followed by “Installation complete” when the install completes.

The installation log file is in the install directory's root, `/opt/ecloud` by default.

Installing Electric Agent/EFS in a grid environment

You must install Electric Agent/EFS on each machine in the grid that ElectricAccelerator will use. The actual installation process for Electric Agent/EFS is the same whether or not you use a grid—*except*, you must answer “yes” (**y**) to any prompt that asks whether or not you plan to use a grid environment, such as LSF.

Note: After Agent/EFS is installed, it remains dormant until the grid management software schedules an agent connect script to run on that host.

Electric Make

Electric Make can be installed on a networked machine or on a shared file server accessible to other build machines. Wherever Electric Make is installed, it must have access to the Cluster Manager machine and Agents on cluster host machines. After Electric Make is installed, configure Electric Make for your build environment and enable communication with Cluster Manager. For information on enabling communication with Cluster Manager, see “Configuring ElectricAccelerator” in the *ElectricAccelerator Electric Make Users Guide*.

Follow upgrading instructions in [Upgrading ElectricAccelerator Components](#) if you are upgrading a previous Electric Make installation.

Installing Electric Make

1. Obtain a copy of the installer (ElectricAccelerator-*<version>*).
2. Log in as root.
3. Run `chmod +x` on the installer to ensure it is executable.
4. Run `./<installer filename> --mode console` to start ElectricAccelerator installation.
5. When the installer welcome message displays, enter **2** to select the **Electric Make** package. Press Enter.

Notes:

1. If you install Electric Make in a non-default location, ensure the same location is used for all agent hosts and the Cluster Manager host.
2. If the installer cannot locate a PDF viewer, it will prompt you to provide a path to one.

The installer installs Electric Make using the configuration details you provided, followed by "Installation complete" when the install completes.

The installation log file is in the install directory's root, `/opt/eccloud` by default.

Additional Install Options

The installer has the following additional install options:

- **Electric Runner Client** - The Electric Runner Client component installs platform-independent tools (`clusterexec`, `clusterupload`, and `clusterdownload`) that allow you to perform operations across all hosts simultaneously. The tools allow you to:
 - Start and stop agents
(In a multi-operating system environment, start and stop apply the requested action only to agents that have the same operating system as the machine where you ran `clusterexec`, unless overridden by `--platform`.)
 - Reboot hosts
 - Run commands on hosts
 - Upload files
 - Download files
- **Electric Runner Server** - The Electric Runner Server component installs the server side (`erunnerd`) 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.
- **Custom** - A custom install allows you to install multiple components at once.

Note: The online help system contains additional information about Electric Runner. Go to the Cluster Manager Administration Tools section in the ElectricAccelerator Installed Tools/Files topic.

Path Settings

For Linux and Solaris, all ElectricAccelerator components are installed in the `/opt/eccloud` directory.

Scripts to add the necessary environment variables are installed in `/opt/eccloud/<arch>/conf`. `arch` is:

- `i686_Linux` for Linux, or
- `sun4u_SunOS` for Solaris SPARC or `i686_SunOS.5.10` for Solaris x86

The scripts are called `ecloud.bash.profile` (for bash shells) or `ecloud.csh.profile` (for csh). You can source the appropriate file in your shell to ensure your PATH includes ElectricAccelerator libraries.

The installation of all ElectricAccelerator components is now complete. If you do not plan to use the Grid Host Manager, go to [Initial Configuration Tasks](#) to complete several tasks before you begin using the product. If you plan to use the Grid Host Manager, follow the instructions in [Integrating ElectricAccelerator with the Grid](#) before proceeding to [Initial Configuration Tasks](#).

Silent Installation Method

If you are installing ElectricAccelerator on a series of identical machines (such as a series of cluster hosts), you can use the “silent install” method, which installs components automatically, without user interaction.

Note: On Windows, if you invoke the installer from the command line, you may receive an “Unknown Publisher” security warning. You can disregard this warning and proceed with installation.

Installer Command Line Options

Use the following command line options when performing a silent install. The options are the same values a user would normally set through the installer interface. Use this format:

```
<installer filename> [options]
```

This table lists each command line option’s equivalent in the installer UI and the variable that is set in the installer properties file. You can use the resulting properties file for running silent installs.

Notes:

1. You can use the values “yes”, “y”, “1” and “no”, “n”, “0” interchangeably within installer command line options.
2. For `--cmdbtype` and `--type`, differences exist between valid command line option arguments and valid properties file parameters. Ensure you provide the correct input for each method.

Command line option	Variable set in the installer	Description
Equivalent installer UI field		
<pre>--adminuserpassword [ARG]</pre> <p>Cluster Manager Admin Password</p>	EC_CM_ADMINUSER_PASSWORD=	Sets the Cluster Manager admin user password.
<pre>--agentallowreboot <y or n></pre> <p>Reboot host if needed</p>	EC_AGENT_REBOOT=y or n	<p>Indicates if you want to reboot after installing Agent/EFS.</p> <p>Default: n</p> <p>For Windows, if you use n, the installer does not restart the Agent service; reboot the host to ensure EFS works properly. Windows may prompt before the host is rebooted.</p> <p>For UNIX, the machine does not reboot unless required, even if you specify <code>EC_AGENT_REBOOT=y</code>.</p>

Command line option	Variable set in the installer	Description
Equivalent installer UI field		
--agentcmhost [ARG] Cluster Manager Host/IP address	EC_AGENT_CMHOST=<hostname or IP address>	Sets the Cluster Manager hostname or IP address for this host to connect to. Example: 192.205.2.19 or winlong-cm Not required if using a grid.
--agentcmport [ARG] Cluster Manager Accelerator Server port	EC_AGENT_CMPORT=8030 (default)	Sets the Cluster Manager server port number for this host to connect to. Not required if using a grid.
--agentinstallerunnerd <y or n> Install Electric Runner Service	EC_AGENT_INSTALL_ERUNNERD=y or n	y installs eRunnerd on agent machines. Default: y
--agentislsfhost <y or n> Host is part of LSF Computing LSF Grid	EC_AGENT_IS_LSF_HOST=y or n	Indicates if the Agent is on the grid. For this option/parameter, the term “LSF” refers to all grid engines. Default: n
--agentnumber [ARG] Number of Agents	EC_AGENT_AGENT_NUMBER=1 to 32	Sets the number of Agents to set up on the host. The maximum is 32 for 32-bit and 64 for 64-bit machines. For Developer Edition, the maximum is the number allowed by your license.
--agentpassword [ARG] ECloud Internal User Shared Password	EC_AGENT_WINUSER_PASSWORD=	Sets the password for ECloudInternalUser (Windows only).
--agentremovelogs <y or n> Remove old Electric Agent logs	EC_AGENT_REMOVE_LOGS=y or n	Removes old agent log files. If not, the install appends to them. Default: n
--agentsecureconsole <y or n> Secure Agent Console port	EC_AGENT_SECURE_CONSOLE_PORT= y or n	y requires an agent-generated key to be entered before commands will run on the agent console port. Default: n
--agentskipecfs <1 or 0> n/a	n/a	Prevents (y) the installer from installing EFS. Default: 0 (n)

Command line option	Variable set in the installer	Description
Equivalent installer UI field		
<pre>--agentskiplofs <1 or 0></pre> <p>n/a</p>	n/a	Prevents (y) the installer from installing LOFS. Default: 0 (n)
<pre>--agenttempdir [ARG]</pre> <p>Agent temporary directory</p>	EC_AGENT_TEMPDIR=<dir>	<p>Sets the temporary directory for all agent files.</p> <p>Windows default: C:\WINDOWS\temp UNIX default: /tmp</p>
<pre>--agentuserlist [ARG]</pre> <p>Existing User</p>	n/a	Sets a list of username/password pairs to be used as login accounts for agents (Windows only).
<pre>--backupdb <y or n></pre> <p>Back up local MySQL database</p>	EC_CM_BACKUPDB=y or n	Backs up the local MySQL database only. Remote databases are not backed up. Default: n
<pre>--baseinstallerunner <y or n></pre> <p>Install Electric Runner client apps</p>	EC_BASE_INSTALL_ERUNNER=y or n	Installs eRunner client applications on Cluster Manager and Electric Make machines. Default: y
<pre>--cmaccelport [ARG]</pre> <p>Cluster Manager Accelerator Server port</p>	EC_CM_ACCELPORT=8030 (default)	Sets the Cluster Manager Accelerator server port number for unencrypted traffic.
<pre>--cmaccelsport [ARG]</pre> <p>Cluster Manager Secure Accelerator Server port</p>	EC_CM_ACCELSPORT=8031 (default)	Sets the Cluster Manager Accelerator server port number for encrypted traffic.
<pre>--cmdbhost [ARG]</pre> <p>Database Host/IP address</p>	EC_CM_DBHOST=localhost (default)	Sets the Cluster Manager database host machine.
<pre>--cmdbname [ARG]</pre> <p>Database Name</p>	EC_CM_DBNAME=ecloud (default)	Sets the name of the database to create on the database server. If you do not use the default local MySQL database, you must provide the name of an existing database.

Command line option	Variable set in the installer	Description
Equivalent installer UI field		
<pre>--cmdbpassword [ARG]</pre> <p>Database Password</p>	EC_CM_DBPASSWORD=eccloud (default)	Sets the user password to use when connecting to the database server.
<pre>--cmdbport [ARG]</pre> <p>Database Port</p>	EC_CM_DBPORT=3306 (default)	Sets the database listening port. Use 1433 for MSSQL.
<pre>--cmdbtype <mysqllocal, mysql, oracle, or mssql></pre> <p>Database Type</p>	EC_CM_DBTYPE=MySQL-Local (default), MySQL, Oracle, or MSSQL	Sets the type of database to use for Cluster Manager.
<pre>--cmdbuser [ARG]</pre> <p>Database User</p>	EC_CM_DBUSER=root (default)	Sets the user name to use when connecting to the database server.
<pre>--cmhttpport [ARG]</pre> <p>Cluster Manager HTTP port</p>	EC_CM_HTTPPORT=80 (default)	Sets the Cluster Manager HTTP server port.
<pre>--cmhttpsport [ARG]</pre> <p>Cluster Manager HTTPS port</p>	EC_CM_HTTPSPORT=443 (default)	Sets the Cluster Manager HTTPS server port.
<pre>--cmlogrotate <y or n></pre> <p>Rotate Cluster Manager logs</p>	EC_CM_LOGROTATE=y or n	Rotates Apache logs. Default: y
<pre>--cmmigratedb <y or n></pre> <p>Migrate existing database</p>	EC_CM_MIGRATEDB=y or n	Migrates the database; this value is meaningful only if performing an upgrade. Default: n
<pre>--cmremove logs <y or n></pre> <p>Remove old Cluster Manager logs</p>	EC_CM_REMOVE_LOGS=y or n	Removes old Cluster Manager log files. If not, the install appends to them. Default: n
<pre>--cmservername [ARG]</pre> <p>n/a</p>	EC_CM_SERVERNAME=	Sets the Apache/SSL server name.
<pre>--cmserviceuser</pre> <p>Cluster Manager Service User</p>	EC_CM_SERVICEUSER=<existing LDAP user or local user>	Sets the user to run the Cluster Manager service.

Command line option	Variable set in the installer	Description
Equivalent installer UI field		
<pre>--cmerviceuserpassword</pre> <p>Cluster Manager Service User Password</p>	EC_CM_SERVICEUSER_PASSWORD=	Sets the password to use for the Cluster Manager service user. If not specified, the account that Cluster Manager runs as is used (Windows only).
<pre>--debug</pre> <p>n/a</p>	n/a	Runs the installer in debug mode.
<pre>--debugconsole</pre> <p>n/a</p>	n/a	Runs the installer with the debug console open.
<pre>--erunnerdport</pre> <p>Electric Runner Server Port</p>	EC_ERUNNERD_PORT=2411 (default)	Sets the port for eRunnerd to listen on.
<pre>--erunnerdremove logs <y or n></pre> <p>Remove old Electric Runner logs</p>	EC_ERUNNERD_REMOVE_LOGS=y or n	Removes old eRunnerd logs. If not, the install appends to them. Default: n
<pre>--finalprefix [ARG]</pre> <p>n/a</p>	n/a	Sets the location where the installed directory tree will be located. Use this option when --prefix is a temporary location that is not the final destination for the product.
<pre>--ignoreoldconf <y or n></pre> <p>n/a</p>	n/a	Ignores the previous configuration.
<pre>--localagentagentlicensefile</pre> <p><browse to location></p>	n/a	Sets the location of the ElectricAccelerator Developer Edition license file.
<pre>--localagenteinsightlicense-file</pre> <p><browse to location></p>	n/a	Sets the location of the ElectricInsight license file.

Command line option	Variable set in the installer	Description
Equivalent installer UI field		
<pre>--mode <console, silent, or standard></pre> <p>n/a</p>	n/a	Sets the mode in which to run the installer. For a console login, standard and console are identical. For a GUI machine, standard brings up the UI. You can use console in a Unix X Window environment to force the use of console mode.
<pre>--noredist <y or n></pre> <p>Do not install the redistributables</p>	EC_BASE_NOREDIST=y or n	Does not install the Microsoft Visual C++ 2005 Redistributable (Windows only). Default: n
<pre>--pdfreader [ARG]</pre> <p><browse to location></p>	UnixPDFReader=<path to PDF reader>	Sets the PDF reader to use.
<pre>--prefix [ARG]</pre> <p>Destination Folder</p>	InstallDir=<dir>	Sets the installation directory. Windows default: C:\ECcloud UNIX default: /opt/eccloud
<pre>--propertyfile [ARG]</pre> <p>n/a</p>	n/a	Sets the property file from which to read installer options.
<pre>--removezip <1 or 0></pre> <p>n/a</p>	n/a	Removes (1) zip files after installation.
<pre>--rootrun <1 or 0></pre> <p>n/a</p>	n/a	Allows the installer to run when root privileges are not present and disables the execution of installer steps that would require root privileges. This option does not change access privileges. 1 (y) or 0 (n).
<pre>--rwprefix [ARG]</pre> <p>n/a</p>	n/a	Specifies the location for read/write files. When using this option for Developer Edition installation, place the license in <rwprefixdir>/<arch>/tmp manually if you did not import it during install.
<pre>--skiprun <1 or 0></pre> <p>n/a</p>	n/a	Prevents (1) the installer from starting the agent or Cluster Manager

Command line option	Variable set in the installer	Description
Equivalent installer UI field		
--temp [ARG] n/a	n/a	Sets the temporary directory used by this program.
--test n/a	n/a	Runs the installer without installing any files.
--type <agent, emake, cm, erunner, erunnerd, localagent, or upgrade> <select install type>	InstallType=Electric Agent, Electric Make, Cluster Manager, Electric Runner Client, Electric Runner Server, Custom, or Local Agent	Sets the type of installation to perform.
--version n/a	n/a	Displays installer version information.

Creating an Installer Properties File

An installer properties file is a text file that defines installation parameters. These parameters are the same values a user would normally set through the installer interface or command line.

To create an installer properties file:

1. Run an installation with your desired settings.

This creates a properties file (install.props) in the top-level install directory.

2. Use the resulting properties file for subsequent silent installs of the same component type.

Components are installed individually so you must create an installer properties file for each ElectricAccelerator component you intend to install.

The table beginning on [page 3-20](#) details the parameters within installer properties files.

Automating a Linux/Solaris Silent Install

Make sure you have a properties file (install.props) created by a successful installation.

To perform a silent install:

1. Log in to the remote machine as root.
2. Invoke the installer:

```
# ./<installer filename> --mode silent --propertyfile <properties file>
```

Automating a Windows Silent Install

Make sure you have a properties file (install.props) created by a successful installation.

To perform a silent install:

1. Log in to the remote machine as Administrator.
2. Invoke the installer in a DOS shell:

```
<installer filename> /mode silent /propertyfile <properties file>
```

Note: Supply the full path to the properties file.

If you are performing a silent upgrade on an agent host by running the install on the host itself, you may be prompted before the machine is rebooted. This prompt occurs if others are logged in to the machine when you run the agent upgrade.

Migrating an eMake only installation to a different machine

On all platforms, you can relocate the complete eMake install directory (anything under the *install dir*, starting with the *arch* directory). A simple file copy is sufficient.

On Windows systems, you must ensure that the Visual Studio 2005 runtime libraries are installed. You can install the libraries by running `../i686_win32/bin/install_redist32.bat` and/or `install_redist64.bat` on the new machine.

You can also choose to add the new location of the bin directory to the PATH.

Installed Tool Files

ElectricAccelerator installs various tools on the Cluster Manager server, Agent/EFS hosts, and Electric Make machines. For a complete list of these files, their description/purpose, and location, see the ElectricAccelerator online help topic, “Installed Tools/Files.”

Note: In addition, if any of *your* tool applications (compilers and so on) are installed locally, Electric Cloud recommends that you install the same version of each application on all machines and keep the version in sync.

ElectricAccelerator Developer Edition Installation

Number of agents

Electric Cloud recommends installing no more than $(n - 1)$ agents, where n is the number cores on your machine. The maximum number of local agents is the number specified by your license file. Because you also have a cluster to work with, eMake uses resources communicating with remote agents as well as local ones, so overall performance may be better with less than this number.

License location

To run local agents, you must have a separate ElectricAccelerator Developer Edition license. You must import the license file during installation, or place the license file in the correct directory manually. The default location is the default root install directory:

- Windows - C:\ECloud
- Linux - /opt/ecloud

If you specify the `--rwprefix` install option and it is different than `--prefix` (the root install directory), place the license in `<rwprefixdir>/<arch>/tmp` also.

If you place the license file manually, you must rename the license file “license.xml” and restart the agent service or daemon before running ElectricAccelerator Developer Edition.

Note: If you configure more than the licensed number of agents, the configured number of agents will start, but only the licensed number of agents can participate in requested builds.

User Interface Installation Method

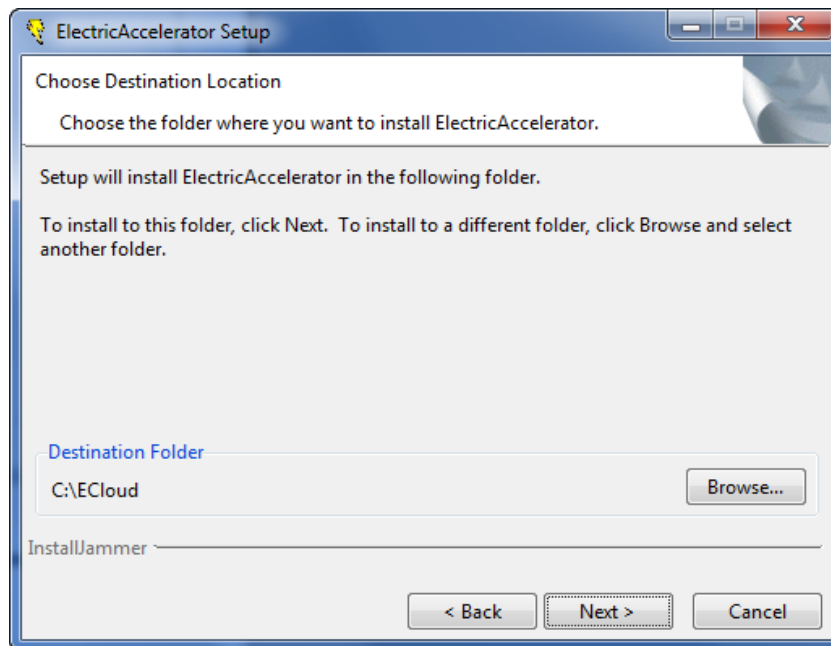
Make sure you log in as root or Administrator (You *must* be a member of the Administrator group—Administrator privileges are not sufficient.)

Note: On Windows, if you are running *rdp* on this host, ensure *rdp* is in installation mode: `change user/install`.

1. Double-click the `ElectricAccelerator-<version>` installer file to start installation. (For Windows systems running Vista or later, the administrator user must right-click the installer and select Run as administrator.)

Note: It may take a few minutes to extract the installation packages to your machine before you see the installation wizard. During installation, if you see a Windows security alert pop-up, click **unblock** and continue.

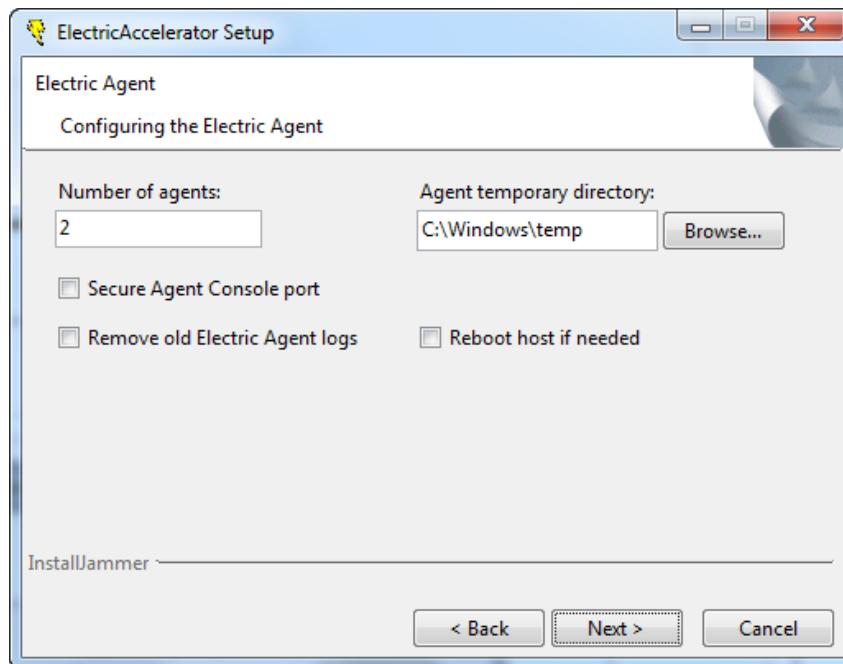
2. For Windows, ElectricAccelerator requires the Microsoft Visual C++ 2005 Redistributable. If it is already installed, select the checkbox. Click **Next** to continue.
3. When the Welcome screen appears, click **Next** to continue to the Setup Type screen and select ElectricAccelerator Developer Edition.
4. On the Choose Destination Location screen, accept the default installation directory (`C:\ECloud` for Windows or `/opt/eccloud` for Linux) or browse to select an alternative directory.



Note: Avoid selecting an alternative directory that includes spaces in the name. Spaces can create problems when configuring connections with other command-line-based components.

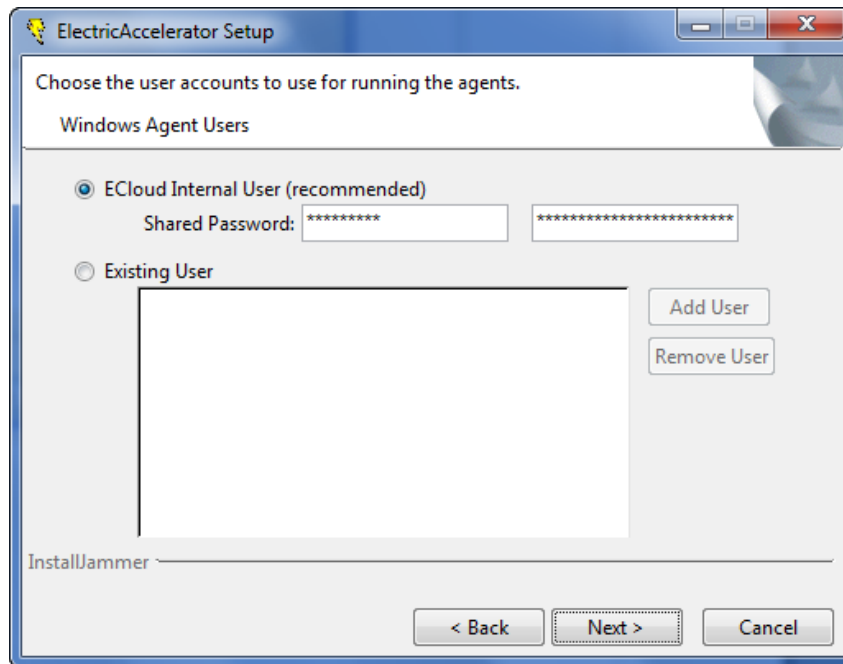
Click **Next**.

5. If the installer cannot locate a PDF viewer, it will prompt you to browse for one. If you wish, browse for a PDF viewer to use. Click **Next**.
6. On the Electric Agent screen, enter options for the locally installed Electric Agent.
 - Enter the number of agents to run—the installer calculates the default number of agents based on the effective number of CPUs on the machine. Default=1, if one CPU is present. If more than one CPU is present, the default value is the number of effective CPUs -1, up to the maximum specified by your license file.
 - Accept the default agent temporary directory, or choose an alternative.
 - For the Secure Agent Console Port checkbox, accept the default if you do not plan to use the secure port, or select the checkbox if you want to use the secure port.
 - Accept the default to keep all existing log files or select the checkbox to Remove old Electric Agent logs.
 - Select the checkbox to Reboot host if needed.



Click **Next**.

7. On the Choose ElectricInsight License screen, browse to your ElectricInsight license (if applicable) and click **Next**.
8. On the Choose ElectricAccelerator Developer Edition License screen, browse to your ElectricAccelerator Developer Edition license and click **Next**.
9. On Windows, select the user accounts you will use to run agents. Accept the default ECloud Internal User or select Existing User. If you select Existing User, click Add User to add a user account.



Click **Next**.

10. When the Start Copying Files screen appears, click **Next**.
11. The Installing screen displays while installation is completing. When installation is finished, the Complete screen displays. Click **Finish**.

Installation is complete.

The installation log file is in the install directory's root, `C:\ECloud` or `/opt/eccloud`, by default.

Note: Installing ElectricAccelerator Developer Edition on Windows systems may unset the environment variable `JAVA_HOME`. Reset `JAVA_HOME` manually.

12. If you did not import the license file during installation, you must do the following *before* running ElectricAccelerator Developer Edition:
 - Place the ElectricAccelerator Developer Edition license file in the install directory's root, `C:\ECloud` or `/opt/eccloud`, by default.
 - Rename the license file "license.xml".
 - Restart the agent service or daemon.

Linux example: Logged in as root, run `/etc/init.d/ecagent restart`

Windows example: Run `ecagent_restart`

The installation of ElectricAccelerator Developer Edition is now complete. Go to [Initial Configuration Tasks](#) to complete any necessary configuration tasks before you begin using the product.

Interactive Command-line Installation Method

Note: Agent/EFS builds a kernel module during installation, so you may need to take this into consideration.

IMPORTANT: For RHEL 5, you must install the `kernel-devel` package version that matches the Linux kernel into which the modules will be loaded.

1. Obtain a copy of the installer (`ElectricAccelerator-<version>`).
2. Log in as root.
3. Run `chmod +x` on the installer to ensure it is executable.
4. Run `./<installer filename> --mode console` to start installation.
5. When the welcome message displays, press Enter.

Provide configuration information. Accept the defaults or type-in alternatives.

- Select the location to install ElectricAccelerator Developer Edition. `/opt/eccloud` is the default location.
- If the installer cannot locate a PDF viewer, it will prompt you to provide the path to one. If you wish, provide the path to a PDF viewer.
- Select the number of agents to run.
The installer calculates the default number of agents based on the effective number of CPUs on the machine. Default=1, if one CPU is present. If more than one CPU is present, the default value is the number of effective CPUs -1, up to the maximum specified by your license file.
- Specify the agent temporary directory. The default is `/tmp`. If you specify a different directory, it must already exist, otherwise the temporary directory defaults to `/tmp`.
- Decide if you plan to use the Secure Agent Console port.
- Decide if you want to remove old Electric Agent logs.
- Decide if you want to reboot the host if needed.
- Supply the location of the ElectricInsight license file (if applicable).
- Supply the location of the ElectricAccelerator Developer Edition license file.

The installer installs ElectricAccelerator Developer Edition using the configuration details you provided, followed by “Installation complete” when the install completes.

The installation log file is in the install directory’s root, `/opt/eccloud` by default.

6. If you did not import the license file during installation, you must do the following *before* running ElectricAccelerator Developer Edition:
 - a. Place the ElectricAccelerator Developer Edition license file in the install directory’s root, `/opt/eccloud`, by default. If you specified the `--rwprefix` install option, place the license in `<rwprefixdir>/<arch>/tmp` instead.
 - b. Rename the license file “license.xml”.
 - c. Restart the agent daemon, for example: logged in as root, run `/etc/init.d/ecagent restart`

Post-Installation Activities

Microsoft Visual Studio

If you intend to use ElectricAccelerator to build Microsoft Visual Studio projects...

Running `eccheckvsinst`

Run `eccheckvsinst` to check agent installations that will be involved in Visual Studio-based builds. The utility is located in `C:\ECloud\i686_win32\unsupported`.

Invoke the utility on the target installations and redirect the output to a text file. Repeat on all agents in the cluster. Then use a text diff tool to compare the various files. There should be no difference except the hostname (as noted in [System Requirements and Supported Platforms](#)).

```
eccheckvsinst > eccheckvsinst.agent1
eccheckvsinst > eccheckvsinst.agent2
diff eccheckvsinst.agent1 eccheckvsinst.agent2
```

Before building the Visual Studio project

Before you can use ElectricAccelerator to build your Visual Studio project, you must install Visual Studio on each agent host, and then log in and run `devenv` as the user that owns the respective agent processes (usually `ECloudInternalUser1`, `ECloudInternalUser2`, and so on). You can use the `psexec` tool (downloadable from <http://technet.microsoft.com/en-us/sysinternals/bb896649.aspx>) to eliminate the need to log in and log out multiple times. Using this tool is more efficient because Visual Studio stores user settings in the registry and creates files in "My Documents." Also, ensure you can build your solutions/projects with the Visual Studio setup on each host.

If you install Visual Studio **after** installing ElectricAccelerator, register the Visual Studio add-in by running `install_ecaddin<N>.bat` where N is 70, 71, 80, 90, or 100, depending on your Visual Studio version.

If you are using Visual Studio 2005 or later, reduce the number of parallel builds Visual Studio performs:

1. In Visual Studio, select Tools > Options.
2. In the Options dialog, open Projects and Solutions > Build and Run.
3. Set maximum number of parallel project builds to 1.

The *Solution Support Add-in Technical Notes* contains additional information about using ElectricAccelerator with Visual Studio.

Uninstalling Microsoft Visual Studio

Uninstall the add-in by running `uninstall_ecaddin<N>.bat` where N is 70, 71, 80, 90, or 100, depending on your Visual Studio version. These bat files are in the `ElectricAccelerator\bin` directory.

Installing an Apache Server Certificate

Description

After installation, you may want to create a new Apache certificate.

By default, ElectricAccelerator generates a temporary self-signed certificate during installation. This certificate is used whenever a browser makes an HTTPS connection to the Apache server. During ElectricAccelerator installation, Apache is configured to look for a private key file named `$HOSTNAME.key` and a certificate named `$HOSTNAME.crt`. These files are in `$DATADIR/apache/conf/ssl.key` and `$DATADIR/apache/conf/ssl.crt` respectively. `$DATADIR` is the directory where ElectricAccelerator data files were installed. To find these files on Windows, go to `C:/ECloud/i686_win32`.

Because the certificate is self-signed, browsers complain that it is an untrusted certificate. Most organizations will want to generate a new certificate signed by a recognized certificate authority (CA) to get rid of the browser warnings. The following list summarizes the process:

- Generate a new certificate and private key
- Send the request to the CA
- Install the signed certificate

Generate a new certificate and private key

Locate `openssl` binary and `openssl.cnf` in `$DATADIR/apache/bin/ssl`.

Copy `openssl.cnf` into a temporary directory, then generate a new private key and certificate.

Note: Enter the appropriate information for your organization when prompted. The most important field is the Common Name, which is the fully qualified name of the host running the Apache server. This name must match the host portion of the URL used to connect to the ElectricAccelerator web interface.

```
$ openssl req -config openssl.cnf -new -out $HOSTNAME.csr
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated into
your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields, but you can leave some blank.
For some fields, there will be a default value, if you enter '.', the field will
remain blank.
-----
Country Name (2 letter code) []:US
State or Province Name (full name) []:California
Locality Name (eg, city) []:Mountain View
Organization Name (eg, company) []:Electric Cloud
Organizational Unit Name (eg, section) []:
Common Name (eg, your websites domain name) []:myserver.mycompany.com
```

Note: The Common Name is the fully qualified name of the host machine where you want the certificate.

```
Email Address []:user@mycompany.com
```

Note: This email address should be for the user to contact if there are issues concerning the certificate.

Please enter the following 'extra' attributes to be sent with your certificate request.

```
A challenge password []:
```

This information generates a new private key in `privkey.pem` and a signing request in `$HOSTNAME.csr`. To use the private key without having to enter a challenge password each time the server starts, issue the following command to strip out the password:

```
$ openssl rsa -in privkey.pem -out $HOSTNAME.key
Enter pass phrase for privkey.pem:
writing RSA key
```

This creates a PEM encoded private key file named `$HOSTNAME.key` without the password.

Send the request to the CA

The `$HOSTNAME.csr` file generated in the previous section is a request for a certificate authority to sign the certificate. When you send this file to the CA, the CA verifies the information inside and sends you a signed certificate in response. The signed certificate includes the original certificate and the signature of the CA. Name the signed certificate '`$HOSTNAME.crt`'.

Install the key and signed certificate

Copy the two files, `$HOSTNAME.key` and `$HOSTNAME.crt`, into `$DATADIR/apache/conf/ssl.key` and `$DATADIR/apache/conf/ssl.crt`, then restart the Apache server. Ensure the `$HOSTNAME.key` file is readable only by the user running the Apache server process.

Also, delete the contents of the temporary directory you created because this directory contains the cryptographic information used to generate the key.

Testing the Cluster Manager Installation

After you provide the license file, launch a web browser and go to localhost (for Windows) or the host's IP address (for UNIX). If the install was successful, the ElectricAccelerator home page appears.

Uninstalling ElectricAccelerator

Use the ElectricAccelerator uninstaller to remove ElectricAccelerator software components from any machine.

Note: The uninstaller removes all ElectricAccelerator components from a machine at the same time. If you are uninstalling as part of an upgrade procedure, follow the instructions in [Upgrading ElectricAccelerator Components](#).

Uninstalling all ElectricAccelerator components from a Linux or Solaris machine

1. Log in as root and change to the `tmp` directory by entering `# cd /tmp`
2. Copy the uninstaller `uninstall-accelerator` to the `/tmp` directory by entering:
for Linux - `# cp /opt/eccloud/uninstall-accelerator /tmp`
for Solaris - `# cp /opt/eccloud/uninstall-accelerator /tmp`
3. Invoke the uninstaller by entering `./uninstall-accelerator`
4. Enter the default `y` to confirm the uninstall. Press Enter.

Note: The ElectricAccelerator uninstaller does not remove files created after initial installation. If you want to remove all files, go to the `/opt/eccloud` directory, which includes the "install" directory, and delete it manually. Ensure you also remove the file `ecagent.conf`. The file's location for Linux - `/etc/sysconfig/ecagent.conf` and for Solaris - `/etc/ecagent.conf`.

Performing a Linux/Solaris silent uninstall

Silent uninstall is performed by adding a flag to the uninstaller. When you invoke the uninstall command:

```
./uninstall-accelerator --mode silent
```

No second opportunity to confirm the uninstall request is displayed. The uninstall begins immediately.

Uninstalling all ElectricAccelerator components from a Windows machine

There are two methods for uninstalling ElectricAccelerator:

First method:

1. Go to the Electric Cloud installation directory and click the `uninstall-accelerator.exe` file.

Note: For Windows systems running Vista or later, the administrator user must right-click the uninstaller and select Run as administrator.

2. After uninstalling ElectricAccelerator, you may want to delete the ECloud directory to ensure all files are deleted.

Second method:

1. From the Control Panel, choose Add or Remove Programs.
2. Select **ElectricAccelerator** from the list and click **Change/Remove** to open the uninstaller.
3. Click **Yes** to confirm the uninstall.
4. If Electric Agent/EFS was installed, you may be prompted to reboot.

Note: The ElectricAccelerator uninstaller will not remove files created after installation. If you want to remove all files, go to the C:\ECloud directory and delete it manually.

Performing a Windows silent uninstall

Run the following command to perform a Windows silent uninstall:

```
uninstall-accelerator.exe /mode silent
```


Chapter 4: Upgrading ElectricAccelerator Components

You can upgrade ElectricAccelerator software components using a traditional interface—text-based for Linux/Solaris and GUI for Windows. You can also upgrade ElectricAccelerator using a “silent” install. Not all instructions are the same for each platform. Follow the instructions carefully for your particular platform.

- Ensure a build is not running before beginning a Cluster Manager or Agent upgrade.
- If you are changing Cluster Manager ports during the upgrade, Electric Cloud recommends closing your browser windows to avoid potential conflicts and to ensure your settings take affect.
- The Cluster Manager server and agent hosts must run the same version of ElectricAccelerator.

Note: Installing any component completely replaces the current installation on that machine. Only one ElectricAccelerator installation can exist on a machine and it is whatever combination of components were chosen from the most recent install. For example, installing the Cluster Manager on a machine with the Electric Agent, results in a machine with Cluster Manager only, not a machine with the Electric Agent and Cluster Manager.

If you plan to install Electric Agent on a machine that has ElectricAccelerator Developer Edition installed, Electric Cloud recommends doing a full uninstall of ElectricAccelerator Developer Edition first.

Upgrade Notes

Linux platforms require additional libraries to run MySQL on the Cluster Manager

To run MySQL 5.5.16 on a Linux Cluster Manager machine, it must have the 32-bit version of the following libraries:

- libaio.so.1
- libncurses.so.5

You may need to add those libraries to the Cluster Manager machine.

RHEL example using a package manager:

```
yum install libaio.i686
yum install ncurses-libs.i686
```

RHEL example without a package manager:

```
up2date --nox --arch=i386 libaio
up2date --nox --arch=i386 ncurses
```

Ubuntu example using a package manager:

```
apt-get install libaio1
```

Remote database configuration upgrade triggers warning

The Cluster Manager can connect to a remote database (for example, MySQL on a different machine), and the user can shut down the local `mysql` by issuing the `/etc/init.d/clustermanager stopmysql` command on UNIX or by stopping the `mysql` service on Windows.

During the upgrade, the installer cannot reach the local database (because it is shut down already), so it gives this prompt:

```
Unable to contact MYSQL database. Please enter a correct username/password:
MYSQL Root User (empty=no database migration):
```

Leaving the value empty allows the installer to finish. Upon completion, a warning message is displayed. This is the expected behavior and does not indicate an actual issue.

Duration of database schema upgrade triggers extraneous failure message

During the ElectricAccelerator upgrade process, the installer attempts to connect to the Cluster Manager. Because the Cluster Manager handles the database schema upgrade, having a large database may cause the Cluster Manager to take a long time to finish the upgrade. A failure message stating that the Cluster Manager failed to come up may be displayed. This dialog does not indicate an actual failure, it means only that the Cluster Manager is still processing the schema upgrade. The Cluster Manager will load after the schema upgrade completes.

Backing up the database

To back up and restore your database, use the `exportData` and `importData` `cmtool` commands as described in the *ElectricAccelerator cmtool Reference and Users Guide*.

Upgrading Multiple Agents on Cluster Hosts

To upgrade multiple agents, follow this procedure:

1. Copy installer executables to the `<ECloud install>/<arch>/bin` directory.
2. From the existing Cluster Manager installation directory, run the following:

```
ecclusterupgrade
```

If you want to limit which operating systems to update, run

```
ecclusterupgrade --targetos=<os>
```

`--targetos=<os>` indicates which operating system to update, which allows you to update any operating system type from any operating system by specifying the desired target operating system. Available values are `windows`, `linux`, and `solaris`.

3. Depending on your setup, you may require the following options as well:

`--agentinstalldir=<path>` indicates the agents' install location. The upgrade expects that all agents you want to upgrade have the same install location. If this is not true, you must run the script multiple

times using the relevant `--agentinstalldir` path names. You must specify the `--hosts` option also to limit the upgrade to those machines that use the “non-standard” agent install directory.

`--agentinstalldir=<path>` specifies the location for the installer. You can use this option instead of copying the installer to the `bin` directory as mentioned in step 1. Generally, the installer is placed in the `bin` directory. You must specify this option if the installer is not in this directory.

If you have multiple installer versions, you can use:

`--installerversion=<version>` to select the version you want to use. This option is needed only if there are multiple installer versions in the `local/--installerdir` location.

or

`--installer=<filename>` to explicitly select an installer. Use an absolute file name for this option.

Windows 7 Cluster Manager and the “administrator” account

If you run `ecclusterupgrade` on a Windows 7 Cluster Manager under a user account other than the “administrator” account, you may encounter a “permission denied” error. This is due to a Windows 7 issue. The workaround is to run `cmd` using “Run as administrator” and then run `ecclusterupgrade`.

Solaris Cluster Manager and `ecconfig`

If you have used `ecconfig` to change the Cluster Manager/port any time after you initially ran the installer, you must manually update the Cluster Manager/port (where the agent hosts point to) in `/opt/ecloud/install.props` before running `ecclusterupgrade`.

Available `ecclusterupgrade` options:

Option	Description
<code>--admin-password=value</code>	Password of the admin user on the Cluster Manager, defaults to 'changeme'.
<code>--agentinstalldir=path</code>	The agent side install directory. You must specify the <code>--hosts</code> option also to limit the upgrade to those machines that use the “non-standard” agent install directory.
<code>--batchsize=int</code>	The number of hosts to process per iteration. This allows you to break the upgrade of a large cluster into more manageable smaller batches to avoid timeouts. Default is 20, the recommended maximum.
<code>--cm=host[:port[:securePort]]</code>	Cluster Manager whose hosts are to be upgraded, defaults to localhost:8030:8031
<code>--cm-secure-port=port</code>	Cluster Manager secure port, defaults to 8031
<code>--gen-hostlist</code>	Query the CM to get the list of hosts, write it to a file, and exit.
<code>--help</code>	This page.

Option	Description
<code>--hosts="host1 host2 ..."</code>	List of hosts to upgrade. Can be a pattern like <code>host[1-3]</code> , <code>host[2,5]</code> . This switch supercedes <code>--cm</code> .
<code>--installer=filename</code>	The installer executable to use. Use an absolute filename for this option. If the installer is not in the expected directory, you must provide the full path to the installer, even when you specify <code>--installerdir</code> .
<code>--installerdir=dirname</code>	The directory with the installer executables.
<code>--installerversion=version</code>	The installer version (for example, 6.2.0.12345). This option is needed only if there are multiple installer versions in the <code>local/--installerdir</code> location.
<code>--installtimeout=value</code>	The number of minutes to wait for the install on the agent to finish. Default is 20 minutes.
<code>--logfile=value</code>	Log file to write messages to. By default, messages are printed to: <code>/var/tmp</code> (Linux and Solaris) or <code>C:\</code> (Windows) in the form of <code>ecloud_node_upgrade_YYYYMMDD_HHMMSS.log</code>
<code>--newcm=hostname/ip</code>	The host name or IP address of the new Cluster Manager where you want to migrate the upgraded agents.
<code>--remotetmpdir=path</code>	The location where remote install is run.
<code>--targetos=os</code>	Agent machine OS (windows, linux, solaris)

User Interface Upgrade Method

Electric Agent/EFS

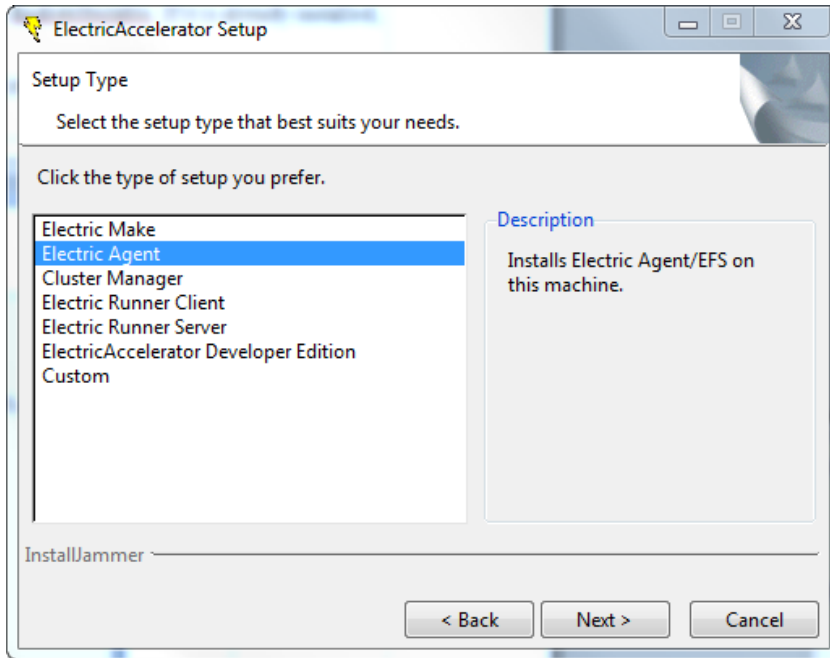
You can upgrade Electric Agent/EFS software on a single host or upgrade all cluster hosts simultaneously. You may want to review [Upgrading ElectricAccelerator Components](#) to upgrade all hosts in a cluster.

Note: When upgrading, be sure to run the installer on the Cluster Manager. Also, do not run the installer from a directory path that contains spaces (such as the Windows desktop).

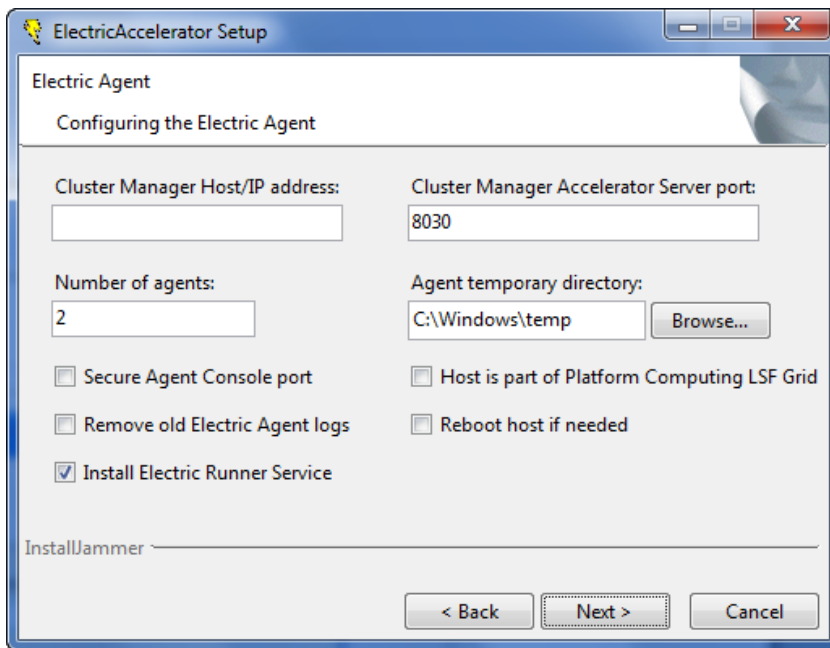
Upgrading Agents on a single cluster host

The Agent/EFS upgrade process for a single host machine is similar to a new Agent/EFS installation, with one *very important* exception. The following screen examples and explanations illustrate this exception.

After invoking the installation wizard, select **Electric Agent** as illustrated in the next screen.



Click **Next** to see the next screen—Electric Agent—which is filled-in with your previous configuration information. The maximum number of Agents allowed on one 32-bit machine is 32; the maximum for one 64-bit machine is 64.

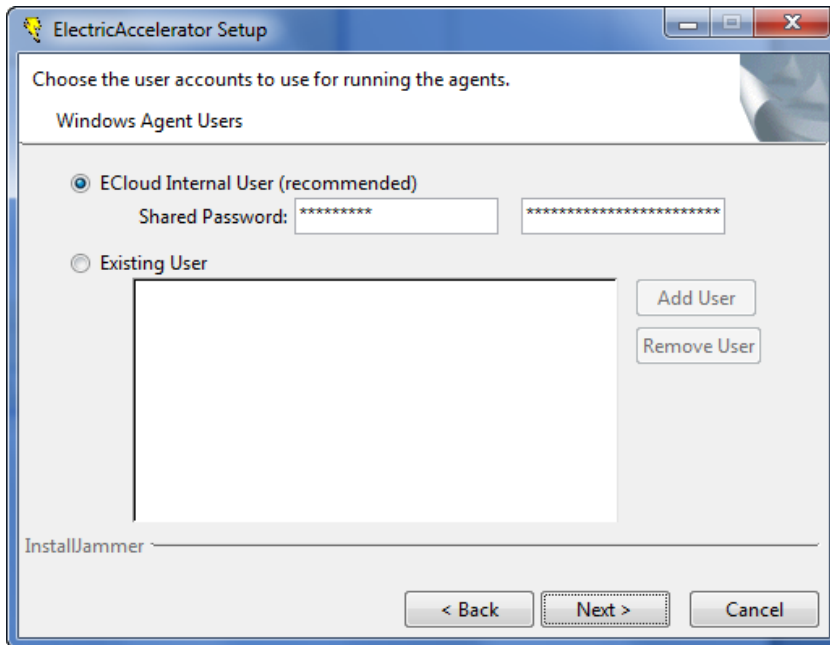


Click **Next** to continue to the next screen.

On Windows, if you previously specified an Agent to run as a specific user, the selection is filled-in as you specified.

At this point, you can revert to the ElectricAccelerator default user if you need the CSRSS, svchost, or OLE support service(s) for your builds—these services are not available to you because you previously specified a specific user. On a single host, use of these services requires each agent service to run as a different user.

To change the service user to allow access to the CSRSS, svchost, and OLE support services, select ECloud Internal User. This means the ECloud user account is used to run agent services. If you do not want to use ElectricAccelerator default accounts, use `ecconfig` on each agent host to assign distinct users to each agent.



To use the `ecconfig` option *after* upgrading your agent machines, refer to [Using the `ecconfig` command-line option](#).

Go to [Installing ElectricAccelerator](#) to complete the Agent upgrade on a single host machine—the upgrade process from this point forward is the same as the initial Agent/EFS installation.

Reboot the agent machine after the upgrade.

The installation log file is in the install directory's root, `C:\ECloud` or `/opt/ecloud`, by default.

[Using the `ecconfig` command-line option](#)

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

Argument	Description
<code>-cm value</code>	Host:port of the Cluster Manager the agent host should point to.
<code>-numagents value</code>	This is the number of agents to start <xx>. If you modify <code>numagents</code> , you must restart the agents for your change to take effect. After the restart, the server database is updated and the agents become active.

Argument	Description
<code>-agentusers value</code>	Specify agent usernames and passwords: <code>-agentusers "username1/password1 username2/password2...usernameN/passwordN."</code> Note: You must specify the same number of username/password pairs as the number of agents. The argument must be encapsulated in quotes.
<code>-agentuser value</code>	Specify a username and password for a specific agent: <code>-agentuser "agentID username/password"</code> where agent ID is the numerical ID of the agent to configure (between 1 and numagents).
<code>-lsfhost value</code>	y or n. Type "y" to specify this machine is part of an LSF grid or "n" for no.
<code>-secureconsole value</code>	y or n. Type "y" to secure the agent console port or "n" for no.
<code>-tempdir value</code>	This is the location where agents should store temp files. Blank for system tempdir.
<code>-help or -?</code>	Print this message.

Cluster Manager

Upgrading Cluster Manager follows the same procedure as your initial Cluster Manager installation. See [Installing ElectricAccelerator](#).

Note: If you want to upgrade the Cluster Manager *and* migrate the database to a different machine, you must first upgrade the Cluster Manager (to update the existing database's schema), then migrate the database to the new machine.

Upgrading Cluster Manager

1. Log in as root or Administrator. Locate your copy of the `ElectricAccelerator-<version>` install file.
2. Double-click `ElectricAccelerator-<version>` to start the upgrade.
3. The Welcome screen appears after the installation packages are extracted.
4. Proceed with the "upgrade installation" in the same way you initially installed Cluster Manager.
5. Ensure you type-in the correct value for your Cluster Manager admin password. The encrypted default value is *changeme*. The upgrade fails if the correct password is not used. You can change the Cluster Manager admin password during the upgrade *only* if the current password is the default.
6. Accept the existing Cluster Manager configuration or update your current configuration settings.

The installation log file is in the install directory's root, `C:\ECloud` or `/opt/ecloud`, by default.

Note: This version of ElectricAccelerator uses MySQL 5.5 (versions 6.0 and earlier use MySQL 5.0). If you did not change the default MySQL password, upgrading Accelerator automatically upgrades all tables in MySQL 5.0 to be compatible with MySQL 5.5. If you changed the default MySQL password and you wish to use MySQL 5.5, you must run the `mysql_upgrade -u root -p` script manually to upgrade the MySQL schema. The `mysql_upgrade` script is located in the `mysql/bin` folder.

Electric Make

Upgrading Electric Make

To upgrade an Electric Make installation, simply overwrite the existing Electric Make version by installing the new version according to instructions provided in [Installing ElectricAccelerator](#).

Note: If you are considering an Electric Make uninstall prior to upgrading your current version, use caution. If Electric Make and Cluster Manager are installed on the same machine, you will remove Cluster Manager when you uninstall Electric Make.

The [Upgrading ElectricAccelerator Components](#) contains detailed compatibility information.

Interactive Command-line Upgrade Method

Electric Agent/EFS

You can upgrade Electric Agent/EFS software on a single host or upgrade all cluster hosts simultaneously. You may want to review [Upgrading ElectricAccelerator Components](#) to upgrade all hosts in a cluster.

Note: After agents on the grid are upgraded, they do not re-register with the Cluster Manager until a build runs. This means the Cluster Manager has old information about the agents and EFS (including outdated version information) until the next build runs and “wakes up” the agents. Initially, after an upgrade, the install log may show the upgrade has failed, but only until the next build is run.

Upgrading Agents on a single cluster host

1. Obtain a copy of the installer (`ElectricAccelerator-<version>`).
2. Log in as root.
3. Run `chmod +x` on the installer to ensure it is executable.
4. Run `./<installer filename> --mode console` to start the ElectricAccelerator upgrade.
5. Enter **1** to select the **Electric Agent/EFS** package. Press Enter.

Enter configuration details for the Agent/EFS upgrade.

Note: Use the Cluster Manager default port 8030, or type-in the alternative port you specified during the initial installation.

The installer upgrades Electric Agent/EFS using the configuration details you entered, followed by “Installation complete” when the upgrade completes.

6. You may receive a message to reboot the agent machine after upgrading Agent/EFS—rebooting may not be required.

Note: On Linux, the installer dynamically builds the EFS kernel module if it detects it does not have a prebuilt version matching your Linux kernel version.

The installation log file is in the install directory's root, `/opt/eccloud` by default.

Cluster Manager

Upgrading Cluster Manager follows the same procedure as the initial Cluster Manager installation. See [Installing ElectricAccelerator](#).

Upgrading Cluster Manager

1. Obtain a copy of the installer (`ElectricAccelerator-<version>`).
2. Log in as root.
3. Run `chmod +x` on the installer to ensure it is executable.
4. Run `./<installer filename> --mode console` to start the ElectricAccelerator upgrade.
5. Enter **3** to select the **Cluster Manager** package. Press Enter.

Enter configuration details for the Cluster Manager upgrade.

- Accept the default listening port you specified during initial Cluster Manager installation.
- Accept the default database listening port you specified during initial Cluster Manager installation.
- You can change the Cluster Manager admin password during the upgrade **only** if the current password is the default.
- Migrate data from the current Cluster Manager version to preserve previously generated data. However, if the database schema has changed significantly, migration is not possible and a warning message is displayed.

If you want to upgrade the Cluster Manager **and** migrate the database to a different machine, you must first upgrade the Cluster Manager (to update the existing database's schema), then migrate the database to the new machine.

Note: Turning on log rotation may affect Cluster Manager performance.

The installer upgrades Cluster Manager using the configuration details you entered, followed by "Installation complete" when the upgrade completes.

The installation log file is in the install directory's root, `/opt/ecloud` by default.

Note: This version of ElectricAccelerator uses MySQL 5.5 (versions 6.0 and earlier use MySQL 5.0). If you did not change the default MySQL password, upgrading Accelerator automatically upgrades all tables in MySQL 5.0 to be compatible with MySQL 5.5. If you changed the default MySQL password and you wish to use MySQL 5.5, you must run the `mysql_upgrade -u root -p` script manually to upgrade the MySQL schema. The `mysql_upgrade` script is located in the `mysql/bin` folder.

Electric Make

Upgrading Electric Make

To upgrade an Electric Make installation, simply overwrite the existing Electric Make version by installing the new version according to instructions provided in [Installing ElectricAccelerator](#).

Note: If you are considering an Electric Make uninstall prior to upgrading your current version, use caution. If Electric Make and Cluster Manager are installed on the same machine, you will remove Cluster Manager when you uninstall Electric Make.

The [Upgrading ElectricAccelerator Components](#) contains detailed compatibility information.

ElectricAccelerator Version Compatibility Matrix

In general, Electric Cloud tries to maintain the system so that older versions of Electric Make are compatible with newer versions of Cluster Manager and the Agent. However, we do not ensure cross-version compatibility between other components, and we do not ensure newer Electric Make versions will work with older clusters. The following matrix illustrates Electric Cloud's compatibility philosophy:

	Newer Agent	Older Agent	Newer Cluster Manager	Older Cluster Manager	Newer Electric Make	Older Electric Make
Electric Make	Yes	No	Yes	No	No	No
Electric Agent	N/A	N/A	No	No	No	Yes
Cluster Manager	No	No	N/A	N/A	No	Yes

Additional Explanation

Electric Make and the Agent use a custom protocol to communicate with each other. Electric Cloud evolves this protocol in a backwards compatible manner so older Electric Make versions are compatible with newer Agent versions. This means you can upgrade the cluster anytime you want, then upgrade Electric Make installations at your leisure.

Electric Make also uses a custom protocol to communicate with remote instances of itself, for example, during remote parse jobs. This protocol is less tolerant than the Electric Make/Agent protocol. In general, it is best to have the same version of Electric Make on the cluster as you have on your build machine—Electric Cloud simplifies this because old versions of Electric Make are preserved on the cluster whenever the cluster is upgraded. For example, if your cluster originally had version 5.x installed and was upgraded to 6.x, the 5.x Electric Make version remains on the cluster—in a special directory in the install tree. You do not have to do anything to ensure this version is used when you run a build using a 5.x Electric Make version (the system handles that for you).

Note: Old versions are NOT installed! What is already there during upgrades is preserved. So, if you install 6.1 on a fresh agent machine, you will NOT be able to use a 6.0 Electric Make with that agent unless you obtain archives of that earlier eMake version from the Electric Cloud FTP site. See [Upgrading ElectricAccelerator Components](#).

Electric Make and the Cluster Manager use yet another protocol to communicate with each other. Again, Electric Cloud evolves this protocol so old Electric Make versions can work with newer Cluster Managers. For example, you should have no problem using a 5.x Electric Make version with a 6.x Cluster Manager.

Finally, the Agent uses a custom protocol to communicate with the Cluster Manager. Electric Cloud has not tried to maintain compatibility between these two components—generally, if the agent version is “close” to the Cluster Manager version, it is compatible. However, you cannot use 6.x Agents with a 5.x Cluster Manager, or vice-versa.

Installing the backward-compatible eMake package on agents

There may be different reasons why you want to install the backward-compatibility package:

- You have new agents in the cluster that have only the most recent version of ElectricAccelerator, and you have old agents that were upgraded to the most recent version. To allow old eMakes to use the new agent machines, you must install the backward-compatibility package on the new agents.
- You encountered incompatibility error messages, indicating the eMake and agent software versions are not compatible:

```
ERROR EC1050: 049300C4:
```

```
The protocol version of Electric Make (45) is incompatible with the software installed on the Electric Cloud cluster 192.168.6.8 (55).
```

```
You need to upgrade your copy of Electric Make; aborting build
```

```
ERROR EC1050: 049300C4:
```

```
The protocol version of this application (45) is incompatible with the software installed on the Electric
```


Accelerator cluster 192.168.6.8 (55).

You need to upgrade your client application; aborting build

Electric Cloud provides archives of eMake versions on the Electric Cloud FTP site.

Installing on a single agent machine

1. Open a terminal window.
2. Change to the Electric Cloud installation directory (for example `c:/ECloud` or `/opt/ecloud`).
3. Download the appropriate set of archive files (for example, `i686_win32_bcp.zip` [all previous eMake versions] or `i686_win32_5.2.5_bcp.zip` [a single eMake version]) from the Electric Cloud FTP site: `accelerator/emake_archive/<platform>`.
4. Unpack the archive in the current directory (`<installDir>`). For Windows, make sure the extraction destination is `<installDir>` without any sub-directories, such as `c:/ECloud`.
5. Repeat the process for each new agent machine.

Installing on multiple agent machines

1. Open a terminal window.
2. Change to a temporary directory (such as `/var/tmp`, or `c:/Windows/Temp`).
3. Download the appropriate set of archive files (for example, `i686_win32_bcp.zip` [all previous eMake versions] or `i686_win32_5.2.5_bcp.zip` [a single eMake version]) from the Electric Cloud FTP site: `accelerator/emake_archive/<platform>`.
4. Unpack the archive in the current directory. For Windows, make sure the extraction destination is the temporary directory (or the directory from step 2) without any sub-directories, such as `c:/Windows/Temp`.
5. Run (as an example): `clusterupload --cm <cmname> i686_win32 c:/ECloud`.

Note: For additional information about how to indicate a range/group of hostnames/IPs for the `clusterexec` command, see the online help system in the Cluster Manager GUI.

Protocol changes

Occasionally, ElectricAccelerator may contain protocol changes that make it necessary to update the eMake version on the build machines. An eMake-only installer is available for this type of update.

Chapter 5: Initial Configuration Tasks

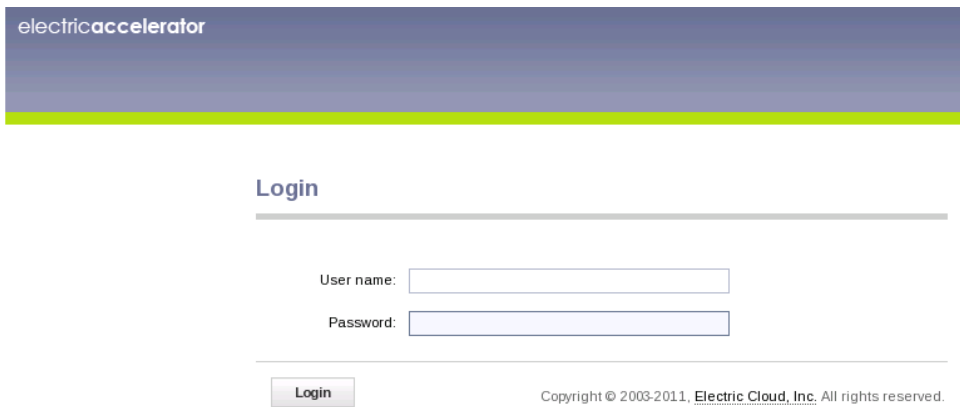
This chapter provides information to get you started—in addition to the initial configuration tasks, there is “a look at the home page” section, and a documentation roadmap for other ElectricAccelerator information and product help.

This chapter contains the following sections:

- Logging In and Licensing
- A look at the ElectricAccelerator Home page
- Enabling Resource Management
- Agent Log Rotation
- Using the `accelerator.properties` file
 - Changing Build and Accelerator Log Locations
 - Disabling Agents by Default
 - User Authentication
 - Setting Up Email Notification
 - Adding Custom Protocol Mismatch Error Text
 - Using the Database Connection Monitor
- Platform-specific Information
 - RHEL 6 and Accelerator services started from a shell
 - Solaris 10 Configuration/debugging
- Using the Web Interface
- Documentation roadmap to locate other ElectricAccelerator product information and help

Logging In and Licensing

From the machine where Cluster Manager is installed, open a browser window and type-in `localhost` to display the Login screen. If you are accessing the Cluster Manager machine remotely, type-in the machine name, **not** `localhost`. If you are not using port 80, use `host:port`.



The image shows the login interface for ElectricAccelerator. At the top, there is a header bar with the text "electricaccelerator" in a sans-serif font. Below this, the word "Login" is centered. Underneath "Login" is a horizontal line. Below the line are two input fields: "User name:" followed by a text box, and "Password:" followed by a text box. Below the password field is a "Login" button. At the bottom right, there is a copyright notice: "Copyright © 2003-2011, Electric Cloud, Inc. All rights reserved."

Type-in your user name and password. For a new installation, the default admin account user name and password is admin/changeme; for an upgrade, the default admin account user name and password is admin/<previous password>.

Click **Login**.

When the Security Alert screen appears, click **Yes** to proceed or **View Certificate** to install the certificate.



If you clicked **Yes**, the ElectricAccelerator Home page appears next. If you have not installed your ElectricAccelerator license, an advisory message appears in the top-left portion of the web page as shown in the following screen.

The screenshot shows the ElectricAccelerator Home page. At the top, there is a navigation bar with the logo 'electricaccelerator' on the left and 'Logged in as 'admin'' | Logout | About | Help on the right. Below the navigation bar, there is a row of tabs: Home, Builds, Agents, Reports, Messages, and Administration. The 'Home' tab is selected. Below the tabs, there is a light blue banner with a yellow warning icon and the text 'A Server license for ElectricAccelerator is required.' To the right of the banner is a button labeled 'Import License'. Below the banner, there is a 'Reset Session' link. The main content area is divided into two sections. The first section is titled 'Server Information' and contains the following data: Running Builds: 0, Active Agents: 0 (License Limit: 100), Inactive Agents: 0, Disabled Agents: 0, Resource Manager: built in, and Database Type: mysql. The second section is titled 'Comments' and contains a table with columns: Comment, Modified By, Last Modified, and Actions. The table is empty, and the text 'Currently, there are no records to display in this list.' is shown below the table.

On the right-side of this page, click **Import License** and the Import License page appears.

The screenshot shows the ElectricAccelerator Administration page. At the top, there is a navigation bar with the logo 'electricaccelerator' on the left and 'Logged in as 'admin'' | Logout | About | Help on the right. Below the navigation bar, there is a row of tabs: Home, Builds, Agents, Reports, Messages, and Administration. The 'Administration' tab is selected. Below the tabs, there is a row of sub-tabs: Filters, Permissions, Users, Groups, User Settings, Server Settings, Database Configuration, and Licenses. The 'Licenses' sub-tab is selected. Below the sub-tabs, there is a section titled 'Licenses' with a '+ Import License' button. Below the button, there is a table with columns: Product Name, Feature Name, Customer Name, Grace Period (days), Expiration Date, Max Agents, and Actions. The table is empty, and the text 'Currently, there are no records to display in this list.' is shown below the table.

You should have already received your license file. If not, contact Electric Cloud technical support.

Use one of the following methods to install your license:

- Browse for your license file and click **Upload File**.
- Open your ElectricAccelerator license file, copy and paste the text to the Data field and click **OK**.

electricaccelerator

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

Home Builds Agents Reports Messages Administration

Fillers Permissions Users Groups User Settings Server Settings Database Configuration Licenses

Import License

License File:

Data:

After the ElectricAccelerator license is installed, you can get familiar with home page features, complete initial configuration tasks, and begin using ElectricAccelerator.

ElectricAccelerator 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.

Server Information

This section provides at-a-glance information, showing the number of currently running builds, the number of active, inactive, or disabled agents, the license limit, the type of resource manager, and the database type.

electricaccelerator

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

Home Builds Agents Reports Messages Administration

Fillers Permissions Users Groups User Settings Server Settings Database Configuration Licenses

Server Information

Running Builds:	0
Active Agents:	0 (License Limit: 100)
Inactive Agents:	0
Disabled Agents:	0
Resource Manager:	built in
Database Type:	mysql

Comments

Comment	Modified By	Last Modified	Actions
Currently, there are no records to display in this list.			

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 Build Details, Build Class Details, and Agent Details pages also.

To add a new comment on the Home page, click the **New Comment** link and the New Server Comment page is displayed.

Type-in the information you want to share or track, click **OK**.

Return to the home page to see your new comment displayed in the Comment box.

Note: While home page comments are intended for server-specific information, you might want to 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 ElectricAccelerator web page.

At any time, you may click **Edit Comment** or **Delete Comment** to update or delete a comment. When you click **Delete Comment**, a message box appears so you can confirm or cancel your delete request.

The Cluster Manager UI supports a maximum of 32,000 server comments. The Cluster Manager will not load if the number comments exceeds 32,000.

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

For more information on using comments for builds, build classes, and agents, click the Help link within the ElectricAccelerator product to view an online help topic on any web page that contains a Comment box. For general web interface navigation, review the “Navigating the Web Interface” help topic within the online help system.

Enabling Resource Management

The ElectricAccelerator Resource Manager feature allows you to select a subset of available agents to run builds. For example, you might have a build that requires a certain hardware feature. If so, you can instruct ElectricAccelerator to use agents on hosts only containing that hardware.

Note: You may choose not to use the Resource Manager feature, but if you use this feature, note that choosing one Resource Manager precludes using the other one. For example, if you choose the Resource Manager for the grid, you cannot use the “built-in” Resource Manager at the same time—Resource Managers are mutually exclusive.

The following two Resource Managers are available through the ElectricAccelerator web UI:

- ElectricAccelerator's "built in" Resource Manager

To enable the "built in" Resource Manager for your build, go to Administration > Server Settings.

From the Server Settings page, for Resource Manager Type, select **built in** if you want to use ElectricAccelerator's internal Resource Manager. For more information, see [ElectricAccelerator online help topics](#).

- Resource Manager for LSF grid integration

After installation is complete and the steps in [Chapter 6](#) have been performed, start or restart Cluster Manager and log in to the Cluster Manager web interface as an administrator. Go to Administration > Server Settings. If everything is available, the LSF radio button will be accessible. If this button is not available, review the Accelerator Log file—LSF may not be available to the process running the ElectricAccelerator server.

If the LSF radio button is available, select it and Cluster Manager will dynamically request hosts as needed to support running Electric Make.

IMPORTANT: The LSF client must be installed on the Cluster Manager host.

Note: Currently, LSF is the only grid Resource Manager available through the web UI. [Integrating ElectricAccelerator with the Grid](#) contains additional information about configuring your environment for use with a grid.

Enabling Resource Management provides an additional subtab in the web UI. From any ElectricAccelerator web page, click the Agents tab to see the newly activated Resources subtab. ElectricAccelerator's online help system contains more Resource Manager information and help for using the Resources and the Resource Details pages.

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.

Configuring log rotation

To configure log rotation, use the agent's `[session logger]` command. The preferred method is to add the desired options to the `runagent.local` file, which is located in the `<ECloud install>\<arch>\bin` directory.

Use this format: `session logger <option>`

For example: `session logger retaincount 10`

Available log rotation options:

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

Alternate methods for configuring log rotation

If you do not use the `runagent.local` file, you can

- use `cmtool` (for example, `cmtool --cm=<cm> runAgentCmd "session logger maxfilesize 5000000000"`)

Note: `cmtool` is not available for local agents (ElectricAccelerator Developer Edition).

or

- telnet to the agent (for example, see the following information)

```
-bash-3.00# telnet localhost 2419
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
qa_solx10_64_1@192.168.6.8: session logger retaincount 5

qa_solx10_64_1@192.168.6.8: session logger retaincount
5

qa_solx10_64_1@192.168.6.8: session logger maxfilesize
1073741824
qa_solx10_64_1@192.168.6.8: session logger maxfilesize 5000
qa_solx10_64_1@192.168.6.8: session logger maxfilesize
5000

qa_solx10_64_1@192.168.6.8: session trace all
all
```

Using the accelerator.properties File

The `accelerator.properties` file allows you to modify parameters that affect a range of Accelerator behaviors. The file resides on the Cluster Manager in the following default location:

- Linux: `/opt/eccloud/i686_Linux/conf`
- Solaris SPARC: `/opt/eccloud/sun4u_SunOS/conf`
- Solaris x86: `/opt/eccloud/i686_SunOS.5.10/conf`
- Windows: `C:\ECloud\i686_win32\conf`

The following sections discuss parameters configurable through the `accelerator.properties` file.

Restart the Cluster Manager service after modifying the `accelerator.properties` file.

Changing Build and Accelerator Log Locations

Modify the following parameters to change the location of build logs and accelerator logs:

- `ACCELERATOR_BUILD_LOGS_PATH=build_logs`
- `ACCELERATOR_LOG=logs/accelerator.log`

Disabling Agents by Default

Agents are enabled by default. If you wish to disable them by default, change the `ACCELERATOR_AGENTS_STARTUP_ENABLED` property to `false`. This change affects agents installed after this point; existing agents on the Agents tab are not affected. If you want the property to change the behavior of existing agents, you must restart the Cluster Manager process.

User Authentication

ElectricAccelerator uses account information from multiple sources. In most cases, the primary account information source is an external LDAP or Active Directory repository; both user and group information is retrieved from the repository. *Local* users and groups can be defined within ElectricAccelerator.

Getting Started

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.
 - Locate the following commented-out text string:
`#ACCELERATOR_SECURITY_CONTEXT_FILES=conf/securityContext.xml`
 - If you did not name your template copy `securityContext.xml`, replace `securityContext.xml` with the filename you chose.
 - Uncomment the `#ACCELERATOR_SECURITY_...` text string by removing the lead “#” sign.
 - Comment out `ACCELERATOR_SECURITY_CONTEXT_FILES=` (which immediately follows the line you uncommented)

4. Use the LDAP information and examples in the following sections to fill-in your own copy of the LDAP template.
5. Restart the Cluster Manager.
6. After the Cluster Manager is running, log in to the Cluster Manager UI as “admin”.
7. Go to Administration > Permissions.
8. Click **Enable User** or **Enable Group**.
9. Search for the desired user or group (or search for “*” to see all). If you set it up correctly, the requested LDAP users are visible.
10. Select the appropriate users or groups to enable using the corresponding checkbox(es).
11. Ensure that the desired permissions are set for the users or groups. (The online help system contains additional information about permissions.)

Note: If you experience permissions issues while reading the `ldap_template.xml` or `ad_template.xml` copy, verify that the account that runs the Cluster Manager also has read permission for that file. To diagnose problems that cause Cluster Manager startup issues, including LDAP/Active Directory configuration problems, check

```
<ECloud install>/<arch>/logs/accelerator.log.
```

Configuring LDAP

A number of options must be customized for any LDAP configuration. The following sample configuration is from the `ldap_template.xml` file. After the sample, see a list of properties with a description.

Sample LDAP Configuration

```
<bean id="ldapDirectoryProvider"
      class="com.electriccloud.security.ldap.LdapDirectoryProviderImpl">
  <property name="providerName" value="LDAP"/>
  <property name="url" value="ldap://myldaphost/dc=company,dc=com"/>
  <property name="userBase" value="ou=People"/>
  <property name="userSearchFilter" value="uid={0}"/>
  <property name="userSearchSubtree" value="false"/>
  <property name="userNameAttribute" value="uid"/>
  <property name="fullUserNameAttribute" value="gecos"/>
  <property name="emailAttribute" value="mail"/>
  <!--<property name="managerDn" value="uid=admin,ou=People,dc=company,dc=com"/> -->
  <!--<property name="managerPassword" value="password"/> -->
  <!--<property name="groupBase" value="ou=Group"/> -->
  <!--<property name="groupMemberFilter" value="(member={0})"/> -->
  <!--<property name="groupMemberAttributes" value="member"/> -->
  <!--<property name="groupSearchFilter" value="(objectClass=groupOfNames)"/> -->
  <!--<property name="groupNameAttribute" value="cn"/> -->
  <!--<property name="verifyCerts" value="true"/> -->
</bean>
```

The following properties configure LDAP mapping:

Property	Description
<code>emailAttribute</code>	(optional) The attribute in a user record that contains the user's email address. If the attribute is not specified, the account name and domain name are concatenated to form an email address.
<code>fullUserNameAttribute</code>	(optional) The attribute in a user record that contains the user's full name (first and last name) for display in the UI. If this attribute is not specified or the resulting value is empty, the user's account name is used instead.
<code>managerDn</code>	(optional) The DN of a user who has read-only access to LDAP user and group directories. If this property is not specified, the server attempts to connect as an unauthenticated user. Not all servers allow anonymous read-only access. Note: This user does not need to be an admin user with modify privileges.
<code>managerPassword</code>	(optional) If the <code>managerDn</code> property is set, this password is used to authenticate the manager user.
<code>providerName</code>	This human readable name is displayed in the user interface to identify users and groups that come from this provider.
<code>userBase</code>	This string is prepended to the <code>basedn</code> to construct the directory DN that contains user records.
<code>userNameAttribute</code>	The attribute in a user record that contains the user's account name.
<code>userSearchFilter</code>	This LDAP query is performed in the context of the user directory to search for a user by account name. The string "{0}" is replaced with the user's login ID. Typically, the query compares a user record attribute with the substituted user login ID.
<code>userSearchSubtree</code>	This property, if true, searches the entire subtree as identified by context. If false, (the default) then only the level identified by the context is searched.
<code>url</code>	The LDAP server URL is in the form <code>protocol://host:port/basedn</code> . Protocol is either <code>ldap</code> or <code>ldaps</code> (for secure LDAP). The port is implied by the protocol, but can be overridden if it is not at the default location (389 for <code>ldap</code> , 636 for <code>ldaps</code>). The <code>basedn</code> is the path to the top level directory that contains users and groups at this site. This is typically the domain name where each part is listed with a <code>dc=</code> and separated by commas. Note: Spaces in the <code>basedn</code> must be URL encoded (%20).
<code>verifyCerts</code>	(optional) This property enables certificate verification for LDAPS connections if true.

In addition to user information, the LDAP server can be queried for **group information**. This query is optional because the local group mechanism can refer to LDAP users as well as local users. However, the following elements can be used to tell the server how to map groups in LDAP:

Property	Description
<code>groupBase</code>	(optional) This string is prepended to the <code>basedn</code> to construct the directory DN that contains group records.
<code>groupMemberAttributes</code>	(optional) A comma separated attribute names list that identifies a group member. Most LDAP configurations only specify a single value, but if there is a mixture of POSIX and LDAP style groups in the directory, multiple attributes might be required.
<code>groupMemberFilter</code>	(optional) This LDAP query is performed in the groups directory context to identify groups that contain a specific user as a member. There are two common forms of group record in LDAP directories: POSIX style groups where members are identified by account name, and <code>groupOfNames</code> or <code>uniqueGroupOfNames</code> records where members are identified by the full user DN. Both forms are supported, so the query is passed two parameters: "{0}" is replaced with the full user record DN, and "{1}" is replaced with the user's account name.
<code>groupNameAttribute</code>	(optional) The group record attribute that contains the name of the group.
<code>groupSearchFilter</code>	(optional) This LDAP query is performed in the context of the groups directory to enumerate group records.

Determining LDAP Mapping

A typical POSIX user record in LDAP looks similar to the example below. To set up a mapping for this record, it is necessary to identify various record components. First, identify the path in the directory that contains user records. In this example, the build user has a distinguished name (dn) of `uid=build,ou=People,dc=mycompany,dc=com`. This name uniquely identifies the build user account and this path splits into three parts:

- **base dn:** `dc=mycompany,dc=com`
- **user base:** `ou=People`
- **user element:** `uid=build`

The `baseDn` is the parent of the directory that contains users. This value is combined with the protocol and server to form the URL. In this case, the URL ends up as `ldaps://dir/dc=mycompany,dc=com`. Next, the `userBase` is the portion of the path that identifies the directory containing all user account records. This value is used directly as the `userBase` configuration element. The remaining portion identifies the user without the `People` directory: `uid=build`. The user name is replaced in this value with the string "{0}" to form the `userSearchFilter: uid={0}`. This query allows the server to search for a user's account name by looking for a record with a matching `uid` attribute. The final mapping step is to identify user record attributes that hold the account name, full user name, and (optionally) the user's email address. In this example, the account name is `uid` (identified earlier), the full user name attribute is `gecos`, and there is no email attribute.

At this point, the server is able to authenticate a user, look up a user by name, and determine the user's full name. For many installations this is sufficient.

Sample LDAP User Record

```
# build, People, electric-cloud.com
dn: uid=jdoe, ou=People, dc=mycompany,dc=com
loginShell: /bin/bash
```

```
uidNumber: 508
gidNumber: 508
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
uid: jdoe
gecos: John Doe
cn: John
homeDirectory: /net/filer/homes/build
```

Also, you can configure the server to look for **LDAP groups** that refer to user accounts. A typical group record is shown below. Like a user account, an LDAP group has a distinguished name with a baseDn, a group base, and a group element. In this case, the base dn is still `dc=mycompany,dc=com`. The `groupBase` configuration element is `ou=Group`, and the group name is `cn=build_users`.

The server must identify records in the directory that correspond to groups—it does this by applying the `groupMemberFilter` to the records in the `groupBase` directory. In this case, group records are members of the `posixAccount` object class, so the filter can be set to `objectClass=posixGroup`. To display a group by its name, the server must know which attribute represents the group name. In this case, set the `groupNameAttribute` to `cn`.

Finally, the server needs a filter to determine which accounts belong to the group and the attribute name that represents a single group member. Group membership can be identified in one of two ways. Either the members are listed by account name, or by their LDAP distinguished name. In this case, POSIX group membership is determined by account name, so set the `groupMemberAttributes` property to `memberUid`, and set the `groupMemberFilter` to `memberUid={1}`.

Sample LDAP Group Record

```
# build_users, Group, mycompany.com
dn: cn=build_users,ou=Group,dc=mycompany,dc=com
objectClass: posixGroup
objectClass: top
gidNumber: 100
memberUid: jdoe
memberUid: mary
cn: build_users
```

Sample Active Directory Configuration

The following XML element defines parameters needed to connect to an Active Directory server and the query to use for looking up user information.

```
<bean id="adDirectoryProvider"
      class="com.electriccloud.security.ldap.ActiveDirectory Provider">
  <property name="providerName" value="ActiveDirectory"/>
  <property name="url" value="ldap://server/dc=company,dc=com"/>
  <property name="managerDn" value="cn=myuser,cn=Users,dc=company,dc=com"/>
  <property name="managerPassword" value="mypw"/>
  <property name="userBase" value="cn=Users"/>
  <property name="userSearchFilter" value="(sAMAccountName={0})"/>
  <property name="userSearchSubtree" value="false"/>
  <property name="userNameAttribute" value="sAMAccountName"/>
  <property name="fullUserNameAttribute" value="name"/>
  <property name="emailAttribute" value="userPrincipalName"/>
  <!-- <property name="groupBase" value=""/> -->
  <!-- <property name="groupMemberFilter" value="member={0}"/> -->
  <!-- <property name="groupMemberAttributes" value="member"/> -->
```

```

<!-- <property name="groupSearchFilter" value="(objectClass=group)"/> -->
<!-- <property name="groupNameAttribute" value="cn"/> -->
    <property name="pageSize" value="500"/>
</bean>

```

Setting Up Email Notification

To configure the Cluster Manager to send email notifications through SMTP, define your SMTP connection properties. SMTP connection properties are contained in `accelerator.properties`. The following properties must be defined:

Property	Description
<code>ACCELERATOR_MAIL_HOST</code>	The name of the SMTP mail server.
<code>ACCELERATOR_MAIL_PORT</code>	The mail server port. Typically, this is '25' for SMTP or '465' for Secure SMTP.
<code>ACCELERATOR_MAIL_PROTOCOL</code>	The mail server protocol. This can be 'smtp' for SMTP or 'smtps' for Secure SMTP.

If your mail server requires an authenticated user to send mail, you must also set the following properties:

Property	Description
<code>ACCELERATOR_MAIL_USERNAME</code>	The user name required for authentication.
<code>ACCELERATOR_MAIL_PASSWORD</code>	The password to use for authentication.

For example:

```

ACCELERATOR_MAIL_HOST=smtp.electric-cloud.com
ACCELERATOR_MAIL_PORT=25
ACCELERATOR_MAIL_PROTOCOL=smtp
ACCELERATOR_MAIL_USERNAME=cm
ACCELERATOR_MAIL_PASSWORD=mypass

```

This example sets up Cluster Manager to send mail as the 'cm' user through the `smtp.electric-cloud.com` server listening for SMTP connections on port 25.

After making changes to the mail configuration, Cluster Manager must be restarted to use the changes.

Configuring the Mail Policy

On the Administration > Server Settings page, various aspects of the global email notification policy can be configured. Cluster Manager sends mail notifications when messages are added to the message log. The following settings control how and when mail is sent:

Setting	Description
Email Interval	The number of minutes between email notifications.
Email Item Limit	The maximum number of messages that will be sent in a single email at the end of each interval. If more messages than the limit arrive, they are ignored.
Email From	The name you use in the 'From' line in the email header.
Email Prefix	A string prepended to every subject line of each outgoing email. This string can be used to assist with mail filtering.

User Email Preferences

Each user that logs in to the Cluster Manager web interface can configure which mail notifications to receive. By default, a user receives no email notifications.

For users to configure their own mail notifications (for messages and build completions), they must 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. By default, the User field is set to None.

To receive notifications about messages in the message log, go to the Messages > Message Policies page. Select the Watch Messages checkbox to enable message notifications. Watch Level controls the minimum severity level of messages that generate a notification.

To receive notifications when builds complete, go to the Builds > Build Classes page. For each build class in the list, select the checkbox in the Notify column to control whether the current user is notified when a build of that class completes. Users can selectively enable notifications for a set of build classes that interest them.

Adding Custom Protocol Mismatch Error Text

You can insert custom error text that is displayed when there is a protocol mismatch.

In the `accelerator.properties` file, add your custom text to the `ACCELERATOR_CLIENT_MISMATCH_DETAIL` property.

Using the Database Connection Monitor

You can configure the Cluster Manager to monitor the connection to the database. The following properties in `accelerator.properties` controls the monitor's behavior:

- `ACCELERATOR_DB_WATCHDOG_FAILURE_THRESHOLD=3`

By default, if the database connection fails three times in a row, the server is restarted. You can change the threshold to a number that meets your needs. Changing the threshold to 0 turns off database connection monitoring.

- `ACCELERATOR_DB_WATCHDOG_INTERVAL=20000`

By default, the monitor check the database connection every 20000 milliseconds (20 seconds). You can change the threshold to an interval that meets your needs.

Platform-specific Information

RHEL 6 and Accelerator services started from a shell

On RHEL 6, if ElectricAccelerator services are started from a shell (as opposed to at boot time), you may encounter errors such as

```
java.lang.OutOfMemoryError: unable to create new native thread
```

This type of error occurs because RHEL 6 greatly decreased the default maximum number of threads per user.

As a workaround, comment out the `nproc` line in the `/etc/security/limits.d/90-nproc.conf` file.

Solaris 10 Configuration/debugging

In order to get the maximum useful information from any core dumps that may occur under Solaris 10 systems (SPARC and x86), run `coreadm -G all -I all` (as superuser) at any time. This reconfigures the machine (persistently, in `/etc/coreadm.conf`) to include the maximum information in core dumps.

Using the Web Interface

After verifying your agents are installed correctly and visible to Cluster Manager, setting up build classes, adding any additional users or groups, you can manage various build aspects, view build results, manage agent activity, enable or disable groups of agents, and add comments to specific builds, build classes, or agents.

Accessing Other ElectricAccelerator Information

The following list is an overview of more product information and help.

- *ElectricAccelerator Electric Make Users Guide* includes:
 - Configuring ElectricAccelerator, defining your build, configuring sources/tools
 - Starting a build, Electric Make settings and options
 - Organizing Builds, Build Classes
 - Make compatibility, output, and support
 - Third-party Integration products, including Cygwin and Rational ClearCase
 - Dependency Management, including eDepend, the Ledger file, and managing the history data file
 - Annotation
 - Performance tuning, managing files, running multiple agents
 - Electric Make command-line options and environment variables
 - Using Electrify
- Help available within the ElectricAccelerator product includes:
 - Online help available from the Help link on each web page
 - Including access to the *ElectricAccelerator Electric Make Users Guide* and *cmtool Reference and Users Guide* PDF files
 - Inline help describes what to type in web page fields (accessed by clicking the Show/Hide links)
- *ElectricAccelerator cmtool Reference and Users Guide* includes:
 - An introductory chapter to familiarize you with cmtool—the ElectricAccelerator command-line tool application
 - A complete list of cmtool commands and arguments

- *ElectricAccelerator Release Notes* includes:
 - Latest feature changes, improvements, and known issues
 - Install/upgrade notes
- *ElectricAccelerator Visual Studio IDE Add-in Technical Notes* includes:
 - Building Visual Studio solutions and projects from within the Visual Studio IDE using Electric Make
- *ElectricAccelerator Solution Support Add-in Technical Notes* includes:
 - Converting Visual Studio projects into NMAKE makefiles
- Information on the Electric Cloud Support Web Site (<https://electriccloud.zendesk.com/home>) includes:
 - Knowledge base articles
 - User forums

Chapter 6: Integrating ElectricAccelerator with the Grid

Grid Host Manager

Grid Host Manager allows ElectricAccelerator to communicate and integrate with a host grid such as LSF or Oracle

Grid Engine. You must enable Grid Host Manager using `cmttool`:

```
cmttool modifyServer --resourceManagerType cloud
```

After you enable Grid Host Manager, the following scripts become available so you can communicate with the host grid server.

Script	Description	Input argument	Output
<code>checkResource.tcl</code>	Checks whether a resource string is valid.	Resource string in "bsub -R" (same as in --emake-resource)	"true" if the string is valid, "false" otherwise.
<code>initialize.tcl</code>	Checks whether the grid setup is ready (for example, already "source profile.lsf" from LSF, can run b* commands, and so on)	none	"true" if the grid setup is ready and initialized, "false" otherwise.

Script	Description	Input argument	Output
killJob.tcl	Kills a grid job based on job ID.	Grid job ID	none
queryJobs.tcl	Queries the status of all grid jobs	none	A string format: job1Id:host1: status, job2Id:host2: status, job3Id:host: status...
submitJob.tcl	Submits a grid job	Resource string, OS, host:port of the Cluster Manager	Grid job ID

Examples

checkResource.tcl

```
# returns "true" if the resource is correct, "false" otherwise.

# define all the variables
set executable "lshosts"
set resource [lindex $argv 0]
set timestamp [clock format [clock seconds] -format {%Y-%m-%d %I:%M:%S %p}]
set log "$timestamp: $executable $resource"

# main program
if { ![catch { exec $executable -R $resource } output] } {

    # logging the output
    exec echo "$log\n$output" >> cloud/cloud.log

    if [regexp {Unknown resource} $output match id] {
        puts -nonewline "false"
    } else {
        puts -nonewline "true"
    }
} else {
    # logging the error
    exec echo "$log\n$output\n$::errorInfo" >> cloud/cloud.log

    puts "error in executing $executable"
    puts $::errorInfo
    exit 1
}
```

initialize.tcl

```
# returns "true" if the cloud is setup correctly, "false" otherwise.

set timestamp [clock format [clock seconds] -format {%Y-%m-%d %I:%M:%S %p}]
set log "$timestamp: initial cloud setup check"

exec echo "$log" >> cloud/cloud.log
puts -nonewline "true"
```

submitJob.tcl

```

# submitJob.tcl - submit jobs in the cloud setup e.g. LSF/SGE
#
# this script accepts three arguments:
# 1. resource - for bsub -R <resource>
# 2. os - for LSF to invoke agentConnect in different paths under different
OSes.
# 3. host:port - the argument for agentconnect

# define all the variables
set executable "bsub"
set resource [lindex $argv 0]
set hostport [lindex $argv 2]
set queue "eaQueue"
switch [lindex $argv 1] {
    "linux" { set agentConnect "/opt/ecloud/i686_Linux/bin/agentconnect" }
    "windows" { set agentConnect "c:\\ecloud\\i686_win32\\bin\\agentconnect" }
    "solaris" { set agentConnect "/opt/ecloud/sun4u_SunOS/bin/agentconnect" }
    "solarisx86" { set agentConnect "/opt/ecloud/i686_SunOS.5.10/bin-
/agentconnect" }
    default { exit 1 }
}
set timestamp [clock format [clock seconds] -format {%Y-%m-%d %I:%M:%S %p}]
set log "$timestamp: $executable $resource [lindex $argv 1] $hostport"

# main program
if { ![catch { exec $executable -R $resource -q $queue $agentConnect $hostport }
output] } {
    # logging the output
    exec echo "$log\n$output" >> cloud/cloud.log

    if [regexp {Job <(\d+)>} $output match id] {
        puts $id
    } else {
        puts $output
        exit 1
    }
} else {
    # logging the error
    exec echo "$log\n$output\n$::errorInfo" >> cloud/cloud.log

    puts "error in executing $executable"
    puts $::errorInfo
    exit 1
}

```

queryJobs.tcl

```

# queryJobs.tcl - query jobs status in the cloud setup e.g. LSF/SGE
#
# this script needs to return the following format:
# job1Id:host1:status,job2Id:host2:status,job3Id:host:status...

# define all the variables
set executable "bjobs"
set queue "eaQueue"
set status(PEND) "pending"
set status(PSUSP) "pending"

```

```
set status(SSUSP) "pending"
set status(USUSP) "pending"
set status(RUN) "running"
set status(DONE) "end"
set status(EXIT) "failed"
set status(UNKWN) "failed"
set status(ZOMBI) "end"

set timestamp [clock format [clock seconds] -format {%Y-%m-%d %I:%M:%S %p}]
set log "$timestamp: $executable"

# main program
set result ""
if { ![catch { exec $executable -q $queue } output] } {
    # logging the output
    exec echo "$log\n$output" >> cloud/cloud.log

    if [regexp {JOBID\s+USER} $output] {
        # split the lines into a list
        set data [split $output "\n"]

        # parse the first line to get all the indices
        set line [lindex $data 0]
        set idStart 0
        set idEnd [expr [string first USER $line] - 1]
        set statStart [string first STAT $line]
        set statEnd [expr [string first QUEUE $line] - 1]
        set hostStart [string first EXEC_HOST $line]
        set hostEnd [expr [string first JOB_NAME $line] - 1]

        # remember to ignore the first line
        foreach line [lrange $data 1 end] {
            # do some line processing here
            append result [string trim [string range $line $idStart $idEnd]]:
            append result [string trim [string range $line $hostStart $hostEnd]]:
            append result [set status([string trim [string range $line $statStart
$statEnd]])],
        }

        # trim the last comma
        set result [string trimright $result ,]
    } else {
        exit 1
    }
} else {
    exec echo "$log\n$output" >> cloud/cloud.log

    if { ![regexp {No unfinished job found} $output] } {
        # logging the error
        exec echo $::errorInfo >> cloud/cloud.log

        puts "error in executing $executable"
        puts $::errorInfo
        exit 1
    }
}

puts $result
```

killJob.tcl

```
# killJob.tcl - kill jobs in the cloud setup e.g. LSF/SGE
#
# this script accepts one argument:
#   id: for bkill <id>

# define all the variables
set executable "bkill"
set id [lindex $argv 0]
set timestamp [clock format [clock seconds] -format {%Y-%m-%d %I:%M:%S %p}]
set log "$timestamp: $executable $id"

# main program
if { ![catch { exec $executable $id } output] } {
    # logging the output
    exec echo "$log\n$output" >> cloud/cloud.log

    if [regexp {Job <(\d+)> is being terminated} $output match id] {
        exit 0
    } else {
        puts $output
        exit 1
    }
} else {
    # logging the error
    exec echo "$log\n$output\n$::errorInfo" >> cloud/cloud.log

    puts "error in executing $executable"
    puts $::errorInfo
    exit 1
}
```

LSF-specific Information

After you install Cluster Manager, log in with the Cluster Manager Service User ID (fully enabled to start and stop LSF jobs) and run a sample LSF job to make certain the user ID is ready.

```
bsub -u myemail@mycompany.com -x -q <queueName> -J <jobName> ls
```

If the job runs and results are returned, the user ID should be appropriate to use for Cluster Manager. (If the appropriate result is not returned, you may need to restart the server after making the necessary changes.)

Update the ElectricAccelerator properties file in the `conf` directory. Edit the following entries in the `accelerator.properties` file.

Agent Connect Script Initial Wait Interval

Use this option when the `agentconnect` script is run. The value is the number of seconds a host waits for agents to be activated and assigned to a build. The typical setting is between 10 and 60 seconds. A short setting might not allow enough time for LSF to schedule the job and start the agents. A setting too long and hosts that will never be used can tie up resources.

```
HM_LSF_INITIAL_WAIT=30
```

Agent Connect Script Idle Host Interval

This option identifies LSF hosts that are no longer needed. When the ElectricAccelerator Cluster Manager sees a host unused for the number of seconds specified in this setting, it cancels the corresponding LSF job. Typical values for this setting are between 3 and 30 seconds. Use a longer setting if your LSF grid is continuously using ElectricAccelerator.

```
HM_LSF_IDLE_HOST_TIME=5
```

Agent Connect Script NotFound Job Cleanup Interval

Use this option to clean up LSF jobs reported as “NOTFOUND” for the number of seconds specified as the value. There is a case in LSF that queries the LSF job against `mbatchd` and returns a “job is not found” result when actually the job is still running. Without this setting, Cluster Manager treats “NOTFOUND” as an invalid status and removes the job on the Cluster Manager side. This action leads to LSF job synchronization issues.

The typical value for this setting is a few seconds more than the value of `MBD_REFRESH_TIME` if you are using multi-threading `mbatchd` mode. If you do not encounter the “NOTFOUND” problem, Zero value for this setting means the “NOTFOUND” status is treated as an invalid status and the job is removed from the Cluster Manager immediately.

```
HM_LSF_NOTFOUND_KILL_TIME=15
```

Agent Connect Script Max Pending Jobs

Use this option to limit the number of jobs the Cluster Manager will leave in a pending state. This option has a distinct impact on your overall build throughput. While builds are running and could use more Accelerator agents for parallel processing, Cluster Manager leaves jobs in the LSF queue in case more hosts become available. The typical setting is 2. If software builds are important and hosts become available quickly, a setting of 5 might be appropriate.

```
HM_LSF_MAX_PENDING_REQUESTS=2
```

Setting Up LSF

There must be a queue that allows Cluster Manager to submit host exclusive jobs or a queue that specifies `HJOB_LIMIT=1`.

Create a queue that limits jobs to the host group

Edit the `conf/lsbatch/xxxx/configdir/lsb.queues` file.

Example queue definition that allows exclusive jobs and limits those jobs to Agent/EFS hosts:

```
Begin Queue
QUEUE_NAME = eaQueue
EXCLUSIVE = Y
HOSTS = eaGroup
DESCRIPTION = Queue to control ElectricAccelerator agents.
End Queue
```

Note: If you use the exclusive method, the `-x` option after the `bsub` command is required in `submitJob.tcl`. The `-x` option is not required when using `HJOB_LIMIT = 1`.

Example queue definition that is NOT exclusive:

```
Begin Queue
QUEUE_NAME = eaQueue
EXCLUSIVE = N
HJOB_LIMIT = 1
HOSTS = eaGroup
DESCRIPTION = Queue to control ElectricAccelerator agents.
End Queue
```

If `EXCLUSIVE=N`, jobs from another queue may also run on the same host.

You may want to consider imposing limits on ElectricAccelerator’s use of the grid. LSF queue definitions provide a variety of controlling features. For example, you can limit how many hosts are available to ElectricAccelerator during certain time periods. See your LSF Administrator’s Guide for more information.

Additional Information

Ensure you have a standard way to construct the LSF resource query string. If not, you may encounter an issue illustrated by the following example:

If the LSF query condition is CPU > 8 and OS = windows:

User1 may put `--emake-resource="cpu>8&&OS==windows"` while User2 may put `--emake-resource="OS==windows&&cpu>8"` (which is the same resource defined in a different order).

The Cluster Manager handles these as two different resources, but LSF handles them as the same resource.

The issue manifests in this manner: If you submit two builds with the same LSF resource query string (the same CPU arch OS ...), the builds do not appear to share the started agents and they act as totally different builds with two separate resources. In the UI, you can identify the issue by hovering the cursor over the Builds page agent bars and seeing a different "Allocated" number for these two builds. If they are on the same resource, the "Allocated" value for both builds would be the same and they would share the agents fairly.

Chapter 7: Windows-Specific Information

This chapter provides additional Windows-specific information.

Agent-Side Application Setup

It is important that applications running on agents are properly initialized for the users that run the agent processes. By default, the users ECloudInternalUser1, ECloudInternalUser2, and so on, own the processes run by agents. Agents can be run as any user, but each agent must have a unique user, which is a requirement for agents to function properly (contact technical support for additional information).

This requirement imposes additional setup steps for agent machines because some applications require per user setup (WinZip, Visual Studio 2005 and 2008, Microsoft Office, and so on). Setup is particularly important for applications that display a dialog if they are not properly initialized because this may result in stalled jobs (the application remains in interactive mode until the agent times it out for lack of activity) during the use of eMake.

To initialize applications, do one of the following:

- Use the psexec tool available from Microsoft (<http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx>) to run the application as different users. This requires that you have the password for the account (for ECloudInternalUser1, and so on, contact technical support for the user password).
- Log in as each of the configured agent users and run the relevant application to initialize it.
- Identify which files/registry keys must be set for a user by using Microsoft's procmon tool (<http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>), creating a .reg file, and copying the user files to create those entities for all users.

Visual Studio

Initializing Visual Studio

Use the psexec method to initialize Visual Studio as shown:

```
psexec -u ECloudInternalUser1 "C:\Program Files\Microsoft Visual Studio 8\Common7\IDE\devenv.exe"
```

As an alternative, disable profiles for Visual Studio by running this regedit script:

```
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VisualStudio\8.0\Profile]
"AppidSupportsProfiles"="0"
```

Microsoft Office

You must run Microsoft Office by using psexec (or logging in directly) because there is no registry setting to initialize Microsoft Office easily.

Ensure that the Visual Basic setting for Security is set to Medium or Lower (assuming the build tries to run VB scripts). Find this under Tools > Options > Security > Macro Security.

WinZip

You must run WinZip by using psexec (or logging in directly) because there is no registry setting to initialize WinZip easily.

MSBuild

Specify configuration information for MSBuild under `C:\Program Files\MSBuild`. This information must be present on all agents, by either syncing that directory among all machines or by adding that directory to the eMake root.

Antivirus Software

Avoid real-time scans on agent machines. Real-time scans can slow down builds and can lead to unexpected failures due to issues with the antivirus software's dll injection. Generally, scans do not find anything relevant because all results are also checked on the eMake machine.

Cygwin

Electric Cloud strongly recommends that the install location be the same on all agents. The install location must also be the same on all eMake machines. The mount mode must be the same for all machines.

Additional Information

Make the following changes to agent/EFS hosts:

- Disable the Windows error reporting service. This avoids popup windows for crashed applications.

For Windows Vista and Windows Server 2008 R2, the installer ***automatically*** does the following:

- Disables the Windows error reporting service.
- Sets `HKLM\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsDisableLastAccessUpdate` to 0. The default value for Windows Vista and Windows Server 2008 R2 is 1.
- Disables User Account Control (UAC) for 64-bit versions. Disabling UAC avoids popup windows for applications that require administrator privileges. If UAC is enabled, application registry access is redirected to each user's virtual store, even if it runs under the Administrator account.

FileInfo and Superfetch services

Because the FileInfo (used by Superfetch) filter driver issues a couple of calls for each file operation in the EFS driver, it has the potential to slow down the EFS driver.

The FileInfo and Superfetch services run on Windows Vista and Windows 7 only (Microsoft officially turned them off in Windows Server 2008 R2).

ElectricAccelerator turns the two services off by default. You can choose to leave them running by removing the following two lines from `runagent` (located in `<ECloud install>\<arch>\bin`):

```
catch {service stop sysmain}
catch {service stop fileinfo}
```

and rebooting the machine.

Terminating stale processes

Certain processes may continue to run on Windows Agent machines. You can choose to terminate all “stale” processes by adding the following line to `runagent` (located in `<ECloud install>\<arch>\bin`):

```
[efs connect] set terminateStaleProcess 1
```

Registry-Specific Information

To allow parallel building of Windows code, ElectricAccelerator virtualizes the registry and the file system. The following sections discuss important registry information.

Registry Use Under Windows

There are two relevant areas of registry use during an ElectricAccelerator build. By default, ElectricAccelerator virtualizes HKEY_CLASSES_ROOT (except HKEY_CLASSES_ROOT\Installer and HKEY_CLASSES_ROOT\Licenses).

- HKEY_CLASSES_ROOT
- All other keys

HKEY_CLASSES_ROOT

This key contains file name extensions and the COM class registration (<http://msdn.microsoft.com/en-us/library/ms724475.aspx> and <http://technet2.microsoft.com/windowsserver/en/library/dd670c1d-2501-4f32-885b-0c6a1ae662f41033.mspx?mfr=true>). Configuration data is stored under the program ids, CLSID, Interface, TypeLib, Appld, and so on.

For entities created during the build, this information must be virtualized to all involved agents.

The following information is registered for a type library:

[http://msdn2.microsoft.com/en-us/library/ms221610\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms221610(VS.85).aspx)

```
\TypeLib\{libUUID}
\TypeLib\{libUUID}\major.minor = human_readable_string
\TypeLib\{libUUID}\major.minor\HELPDIR = [helpfile_path]
\TypeLib\{libUUID}\major.minor\Flags = typelib_flags
\TypeLib\{libUUID}\major.minor\lcid\platform = localized_typelib_filename
```

Other entities that are registered by UUID are registered in different places:

[http://msdn2.microsoft.com/en-us/library/ms221150\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms221150(VS.85).aspx)

A ProgID("ApplicationName") maps to and from a CLSID(GUID). The CLSID maps to the actual ActiveX component ("APP.EXE"). The type library is available from the CLSID:

```
\CLSID\TypeLib = {UUID of type library}
\CLSID\{UUID} = human_readable_string
\CLSID\{UUID}\ProgID = AppName.ObjectName.VersionNumber
\CLSID\{UUID}\VersionIndependentProgID = AppName.ObjectName
\CLSID\{UUID}\LocalServer[32] = filepath[/Automation]
\CLSID\{UUID}\InProcServer[32] = filepath[/Automation]
```

[http://msdn2.microsoft.com/en-us/library/ms221645\(VS.85\).aspx](http://msdn2.microsoft.com/en-us/library/ms221645(VS.85).aspx)

Applications that add interfaces must register the interfaces so OLE can find the appropriate remoting code for interprocess communication. By default, Automation registers dispinterfaces that appear in the .odl file. It also registers remote Automation-compatible interfaces that are not registered elsewhere in the system registry under the label ProxyStubClsid32 (or ProxyStubClsid on 16-bit systems).

The syntax of the information registered for an interface is as follows:

```
\Interface\{UUID} = InterfaceName
\Interface\{UUID}\Typelib = LIBID
\Interface\{UUID}\ProxyStubClsid[32] = CLSID
```

All Other Keys

Other keys are likely not relevant to the build. HKEY_LOCAL_MACHINE, HKEY_CURRENT_USER, HKEY_USERS, and HKEY_CURRENT_USER are machine specific. If other areas must be virtualized, it is recommended that you add them to the `emake-reg-root` option.

Registry Underlay

When a process in the build requests information from the registry, the EFS first checks if the requested key is already present in its cache. If the key is not present, the EFS relays the request to the agent, which in turn sends the request to eMake. After receiving the response from eMake, the agent loads the key into the EFS cache, subject to the following conditions:

- If the key does not exist at all in the local registry on the agent host, the value from the eMake response is used unconditionally.
- If the key exists in the local registry, the value from the local registry is given precedence over the initial value from eMake, but not any value set by prior commands in the build. That is, if the key is modified during the course of the build, the modified value is used in preference of any value from the local registry.

The order of precedence is (lowest to highest):

- value from eMake host registry prior to the start of the build
- value from the agent host registry, if any
- value set by an earlier job in the build

The additional checking of precedence enables ElectricAccelerator to interoperate with tools that store host-specific licensing information in the registry. If the agent simply used the value from eMake unconditionally in all cases, such tools would fail to operate correctly.

Electric Cloud STRONGLY RECOMMENDS AGAINST running builds locally on agent host machines. Running builds locally on agent machines may add relevant keys to the local machine, which take precedence over the eMake machine's keys. If a key that should come from the eMake machine (such as the typelib information for a lib generated during the build) is already present on the agent due to a locally performed build, the wrong information is used, possibly causing a broken build.

If an agent machine has locally created keys, remove the typelibs that are created during the build from the registry. Any typelib that has an invalid path name associated with it is a likely candidate for an "underlayed" lookup.

Ideally, typelibs created by a build are known. At this point, it is recommended to check for their existence on the cluster. If an error occurs that indicates the direction of this problem (for example, a library/typelib cannot be found), investigate the failing agent's registry.

ExcludeProcessList Registry Entry

You can add a multi-string registry value to the agent host inside HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ElectricFS to exclude processes from interfering with EFS and causing a crash. The ExcludeProcessList entry can list processes from McAfee antivirus (for example, Mcshield.exe and mfevtps.exe) or other antivirus software.

Note: Make these registry changes only if the system crashed previously.

Windows Kernel Non-paged Pool Memory

Under high-volume network traffic, the non-paged pool memory in the Windows kernel has the potential to become depleted. This issue in the Windows kernel can be triggered by applications such as MySQL, Java server applications, and so on. Over a period of time, this results in a machine crash.

The workaround is to use the 64-bit version of Windows. Though this does not completely resolve the issue, it increases the available memory to a point where crashes are unlikely and infrequent.

Support for Managed Code/.NET

There are no known limitations with respect to building managed code or .NET code. There are, however, areas to keep in mind:

- Agents must have the same version of .NET installed.
- Agents must be on the same service pack level and have the same OS and tool hotfixes installed.
- The language bar must be enabled on all agent machines, or disabled on all agent machines.
- Including the Global Assembly Cache in the eMake root is not recommended. Contact technical support for more details.

Glossary of Terms

.TLB file – A type library is a binary file containing all type information that is needed to use procedures or classes in DLLs ([http://msdn.microsoft.com/en-us/library/ms221060\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms221060(VS.85).aspx) and <http://vb.mvps.org/hardcore/html/whatistypelibrary.htm>). It corresponds to a registry key hierarchy under HKEY_CLASSES_ROOT, which maps a GUID to an actual file name and defines additional configuration information.

.PCH file – A precompiled header file is used by the Visual Studio tool chain. It allows the compiler to load just a memory dump of the already-parsed header information, thus reducing compile time drastically. A potential issue exists when using pch files on different machines because the files are sensitive to the program code size of the cl process. This means that when certain system libraries change (due to a hotfix) or additional libraries are dynamically loaded (such as antivirus scanners or the language bar), the pch file may become unusable.

.PDB file – A program database file contains debugging symbols for object files built using the Visual Studio tool chain. All compiles accessing the same pdb file must be serialized during an ElectricAccelerator build to guarantee correctness, which may slow down builds.

.DLL file – A dynamically loaded library under Windows is code that is shared among applications (prime examples are the c runtime library and other system libraries) and loaded dynamically when the application starts.

