

ElectricFlow 7.1 Installation Guide

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

ElectricFlow 7.1

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

Published 7/27/2017

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

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

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

Copyright protection includes all forms and matters of copyrightable material and information now allowed by statutory or judicial law or hereinafter granted, including without limitation, material generated from software programs displayed on the screen such as icons, 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, ElectricFlow, and ElectricInsight are registered trademarks or trademarks of Electric Cloud, Incorporated.

Electric Cloud products—ElectricAccelerator, ElectricCommander, ElectricInsight, ElectricFlow Deploy, ElectricFlow Release, and Electric Make—are commonly referred to by their “short names”—Accelerator, Commander, Deploy, eMake, Flow, Insight, and Release—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: Introduction to ElectricFlow	1-1
Unique Functionality	1-2
Challenges Solved by ElectricFlow	1-4
Architecture	1-4
Local Configuration	1-4
Remote Web Server Configuration	1-5
Clustered Configuration	1-7
Chapter 2: System Requirements and Supported Platforms	2-1
Supported Server Platforms	2-1
Supported Agent Platforms	2-10
Server and Agent Compatibility	2-10
ectool/Perl API Incompatibility	2-11
Diffie-Hellman Key Size Incompatibility	2-11
Hardware Requirements	2-11
Browser Requirements	2-11
Port Usage	2-12
Default Server Ports	2-12
Default Database Ports	2-12
Default Ports for ElectricFlow Services	2-12
Avoiding Port Conflicts	2-12
Database Requirements	2-13
Built-In Database	2-13
Supported Alternate Databases	2-13
Database Sizing	2-13
Disk Usage	2-13
Memory Usage	2-15
Modifying the Memory Settings for an ElectricFlow Server	2-15
Modifying the Memory Settings for an ElectricFlow Agent	2-16
Checksum Utility	2-16
Software Licenses	2-16
Standard License	2-16
Enterprise License	2-17
Registered Host License	2-17
Legacy Software Licenses	2-18
Chapter 3: Installing ElectricFlow	3-1
ElectricFlow Installer Files	3-1
Choosing the Correct Installation Process	3-3

User Interface Installations	3-3
Interactive Command-Line Installation	3-4
Silent Unattended Installation	3-5
Non-Server Platform Agent Installation	3-5
Before You Install ElectricFlow	3-5
Linux and Windows Installations	3-5
Linux Installations	3-6
Default Installation Directories	3-8
Graphical User Interface Installation Methods	3-8
Running an Express Server Graphical User Interface Installation	3-8
Running an Advanced Graphical User Interface Installation	3-13
Running an Express Agent Graphical User Interface Installation	3-21
Running an Express Agent Graphical User Interface Installation (Agent-Only Installer)	3-26
Running an Advanced Agent Graphical User Interface Installation (Agent-Only Installer)	3-31
Interactive Command-Line Installation Methods	3-38
Running an Express Server Command-Line Installation	3-38
Running an Advanced Command-Line Installation	3-40
Running an Express Agent Command-Line Installation	3-43
Running an Express Agent Command-Line Installation (Agent-Only Installer)	3-46
Running an Advanced Agent Command-Line Installation (Agent-Only Installer)	3-48
Running an Express Agent Command-Line Installation (Agent-Only Installer) When the Server Uses Registered and Concurrent Licenses	3-51
Silent Unattended Installation Method	3-53
Running a Silent Install	3-53
Silent Install Arguments	3-54
Linux Silent Installation Examples	3-58
Windows Silent Installation Examples	3-61
Linux Repository Server Silent Installation	3-63
Windows Repository Server Silent Installation Example	3-64
Non-Server Platform Agent Installation Method	3-64
Installing Agents with the UNIX Installer	3-65
UNIX Agent Silent Installation	3-66
Installing or Upgrading Remote Agents	3-68
Prerequisites	3-68
Permissions for Installing or Upgrading Remote Agents	3-69
Installing Remote Agents Using the Web Interface	3-70
Installing Remote Agents Using the API	3-81
Upgrading Remote Agents Using the Web Interface	3-90
Upgrading Remote Agents Using the API	3-96
Moving the Artifact Repository in Linux	3-98
Moving the Artifact Repository in Windows	3-100
Connecting ElectricFlow to a Microsoft SQL Server Named Instance	3-102
Installing the MySQL JDBC Driver	3-103
Logging Into the ElectricFlow Web Interface	3-104
Chapter 4: Clustering	4-1
Benefits from Clustering	4-1
Architecture of an ElectricFlow Cluster	4-1

Resource, Agent, and Procedure Configuration Considerations	4-2
Default Local Resource Use	4-2
Unsupported Host	4-3
Separate Local Agents For Improved Performance	4-3
Pool Local Agents For Improved Reliability	4-3
Procedure Strategies	4-3
Agent Resource Strategies	4-3
Database Restriction	4-3
broker-data Directory Restriction	4-4
Software for Clustering	4-4
Apache Zookeeper	4-4
Load Balancer	4-4
Dependencies for Clustering	4-4
Configuring Clustering	4-5
Separating Agents from ElectricFlow Servers	4-5
Preparing Your Cluster Resources	4-6
Installing and Configuring a Load Balancer	4-6
Installing ZooKeeper	4-7
Running ZooKeeper as a Service on Linux	4-8
Running ZooKeeper as a Service on Windows	4-9
Ensuring ZooKeeper Can Locate Java	4-10
Verifying that ZooKeeper is Running	4-10
Exhibitor Software	4-10
Configuring a Multi-ZooKeeper Cluster	4-11
ZooKeeper Requires a Majority of Nodes to Be Up	4-11
Installing ElectricFlow Software	4-11
New ElectricFlow Installation for Reliability	4-12
New ElectricFlow Installation for Performance	4-12
Converting an Existing ElectricFlow Installation for Reliability	4-13
Converting an Existing ElectricFlow Installation for Performance	4-14
Duplicating Repository Contents	4-14
Running a Cluster in Clustered Mode	4-15
Running a Cluster in Single-Server Mode	4-17
Adding the Configuration to ZooKeeper	4-18
Uploading Configuration Files to ZooKeeper	4-18
Prerequisites	4-18
Location	4-18
Command	4-18
Import Files	4-19
Next Steps	4-20
Getting information on the ElectricFlow Server Cluster from ZooKeeper	4-20
Prerequisites	4-21
Locations	4-21
Command	4-21
Output	4-21
Configuring Web Server Properties	4-22
Configuring Repository Server Properties	4-22

Configuring ElectricFlow Agents	4-23
Configuring the Cluster Workspace	4-23
Configuring ElectricFlow Repositories	4-24
Adding Trusted Agents to Clusters	4-24
Preparing Your Cluster Environment	4-24
Method 1	4-25
Method 2	4-26
Separating Agents from ElectricFlow Servers	4-26
Verifying ElectricFlow Services	4-26
Accessing ElectricFlow with Clustering	4-27
Health Check for the ElectricFlow Cluster	4-27
Additional Ways to Improve an ElectricFlow Cluster	4-27
Third-Party Software	4-27
ElectricFlow Components	4-28
Chapter 5: Configuration	5-1
Applying an Enterprise License Key	5-1
External Database Configuration	5-2
Database Interactions	5-2
Database User	5-2
Recommended MySQL Settings	5-3
Configuring ElectricFlow to Use an Alternate Database	5-3
Setting the Database with the Web Interface	5-3
Setting the Database from a Command Line	5-4
Universal Access to the Plugins Directory	5-10
Configuring Universal Access for a Network Location	5-10
Replicating the Plugin Directory on Remote Systems	5-13
Environment Proxy Server Configuration	5-13
Configuring Proxy Settings for Servers	5-14
Testing Server Proxy Settings	5-14
Configuring Proxy Agents	5-14
Increasing File Descriptors on Linux	5-15
Adjusting Swappiness on Linux	5-16
Chapter 6: Roadmap to Upgrade ElectricFlow	6-1
Single-Site Architecture	6-4
Remote Web Server Configuration	6-5
Clustered Configuration	6-6
Server Components in ElectricFlow	6-7
Chapter 7: Upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x (Use Cases 1 to 6)	7-1
Upgrade Process Overview	7-3
Properties Changed to UUIDs	7-5
Use Cases	7-10
Use Case 1: Large Database Upgrade on Existing Hardware	7-11
Step 1.1: Download and Extract the Upgrade Bundle	7-12
Step 1.2: Run the Migration Tool to Convert \$[jobID] and \$[workflowId] References	7-13
Step 1.3: Prepare for the Database Upgrade	7-15
Step 1.4: Run the Background Upgrader	7-18

Step 1.5: Troubleshooting the Database Upgrade	7-26
Step 1.6: Shut Down the Source (Original) Server	7-34
Step 1.7: Run the Installer	7-34
Use Case 2: Large Database Upgrade on New Hardware	7-43
Step 2.1: Download and Extract the Upgrade Bundle	7-44
Step 2.2: Run the Migration Tool to Convert \${jobID} and \${workflowId} References	7-45
Step 2.3: Prepare for the Upgrade	7-47
Step 2.4: Run the Background Upgrader	7-51
Step 2.5: Troubleshooting the Database Upgrade	7-59
Step 2.6: Run the Installer	7-67
Step 2.7: Shut Down the Existing Server	7-75
Step 2.8: Complete the Upgrade	7-75
Use Case 3: Small Database Upgrade on Existing Hardware	7-75
Step 3.1: Download and Extract the Upgrade Bundle	7-75
Step 3.2: Run the Migration Tool to Convert \${jobID} and \${workflowId} References	7-77
Step 3.3: Prepare for the Upgrade	7-79
Step 3.4: Shut Down the Source (Original) Server	7-80
Step 3.5: Run the Installer	7-80
Use Case 4: Small Database Upgrade on New Hardware	7-89
Step 4.1: Download and Extract the Upgrade Bundle	7-89
Step 4.2: Run the Migration Tool to Convert \${jobID} and \${workflowId} References	7-91
Step 4.3: Prepare for the Upgrade	7-93
Step 4.4: Run the Installer on the New Server	7-94
Step 4.5: Shut Down the Existing Server	7-103
Step 4.6: Complete the Upgrade	7-103
Use Case 5: Built-In Database Upgrade on Existing Hardware	7-103
Step 5.1: Download and Extract the Upgrade Bundle	7-103
Step 5.2: Run the Migration Tool to Convert \${jobID} and \${workflowId} References	7-105
Step 5.3: Create a Database Backup of the ElectricCommander 4.2.x Database	7-107
Step 5.4: Shut Down the Source (Original) Server	7-107
Step 5.5: Run the Installer	7-107
Step 5.6: Import the ElectricCommander 4.2.x Database into ElectricFlow 7.x	7-116
Use Case 6: Built-In Database Upgrade on New Hardware	7-116
Step 6.1: Download and Extract the Upgrade Bundle to the Existing Server	7-116
Step 6.2: Run the Migration Tool to Convert \${jobID} and \${workflowId} References	7-118
Step 6.3: Create a Database Backup of the ElectricCommander 4.2.x Database	7-122
Step 6.4: Install ElectricFlow 6.x on the New Server	7-122
Step 6.5: Shut Down the Existing Server	7-131
Step 6.6: Import the ElectricCommander 4.2.x Database into ElectricFlow 6.x	7-131
Chapter 8: Use Case 7: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version in a Clustered Environment	8-1
Agent Configuration Settings Preserved After an Upgrade	8-1
Properties in <data_dir>/conf/agent.conf	8-2
Properties in <data_dir>/conf/agent/wrapper.conf	8-2
Properties in <data_dir>/conf/agent/agent.properties	8-3
Use Cases	8-4

Upgrading Clusters	8-4
Preparing for Your Upgrade	8-5
Upgrade Testing	8-5
Backing Up Your Existing ElectricFlow Data	8-5
Upgrade Installer Preservation	8-7
MySQL Upgrades	8-7
Choosing the Correct Upgrade Method	8-7
User Interface Upgrade Method	8-9
Interactive Command-Line Upgrade Method	8-12
Silent (Unattended) Upgrade Method	8-13
Copying Repository Contents	8-13
Uploading Configuration Files to ZooKeeper if Needed	8-14
Chapter 9: Use Case 8: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version	9-1
Agent Configuration Settings Preserved After an Upgrade	9-1
Properties in <data_dir>/conf/agent.conf	9-2
Properties in <data_dir>/conf/agent/wrapper.conf	9-2
Properties in <data_dir>/conf/agent/agent.properties	9-3
Use Cases	9-4
Preparing for Your Upgrade	9-4
Upgrade Testing	9-4
Backing Up Your Existing ElectricFlow Data	9-4
Upgrade Installer Preservation	9-5
MySQL Upgrades	9-5
Choosing the Correct Upgrade Method	9-5
User Interface Upgrade Method	9-7
Interactive Command-Line Upgrade Method	9-10
Silent (Unattended) Upgrade Method	9-11
Copying Repository Contents	9-11
Chapter 10: Uninstalling ElectricFlow	10-1
Uninstalling ElectricFlow on Windows	10-1
Uninstalling on Windows 2008 or Windows 7	10-1
Uninstalling ElectricFlow on UNIX	10-1
Disaster Recovery	10-2
Disaster Recovery Environment Setup	10-2
Steps to Perform During a Disaster Recovery Failover	10-6
Server Maintenance	10-7
Chapter 11: Maintenance	11-1
ElectricFlow Server Backups	11-1
Data Backup Methods	11-1
Preparing for a Backup	11-2
Backing Up an ElectricFlow Server	11-2
ElectricFlow Server Restores	11-3
Preparing for a Restore	11-3
Restoring Your ElectricFlow Server	11-3
Switching to an Alternate Database	11-10
Preventing Database Changes During the Export	11-10

Exporting and Importing Your Data	11-10
Apache Web Server or Agent Certificates	11-10
Generating a CA Request	11-11
Sending the CA Request	11-11
Installing the Signed Certificate	11-11
Using chkconfig	11-11
Starting and Stopping Servers and Agents Manually	11-12
Stop All ElectricFlow Agent Services	11-12
Stop All ElectricFlow Server Services	11-12
Start All ElectricFlow Agent Services	11-13
Start All ElectricFlow Server Services	11-13
Log File Locations	11-14
Agent Logs	11-14
Server Logs	11-14
Web Server Logs	11-14
Repository Server Logs	11-15
Installer Logs	11-15
Web Interface Online Help System	11-15
Chapter 12: Troubleshooting	12-1
ElectricFlow Server is Unresponsive and Displays an OutOfMemory Error	12-1
Windows PHP Does not Handle Time Zones Correctly	12-2
ElectricFlow Server Certificate Fails Security Scan	12-2
Linux Upgrade Breaks Symbolic Links	12-4
Chapter 13: Installing ElectricFlow Agents	13-1
Graphical User Interface Installation Methods	13-1
Running an Express Agent Graphical User Interface Installation (Agent-Only Installer)	13-1
Running an Advanced Agent Graphical User Interface Installation (Agent-Only Installer)	13-6
Interactive Command-Line Installation Methods	13-13
Running an Express Agent Command-Line Installation	13-13
Running an Express Agent Command-Line Installation (Agent-Only Installer)	13-16
Running an Advanced Agent Command-Line Installation (Agent-Only Installer)	13-18
Running an Express Agent Command-Line Installation (Agent-Only Installer) When the Server Uses Registered and Concurrent Licenses	13-21
Non-Server Platform Agent Installation Method	13-24
Installing Agents with the UNIX Installer	13-24
UNIX Agent Silent Installation	13-25
Index	14-1

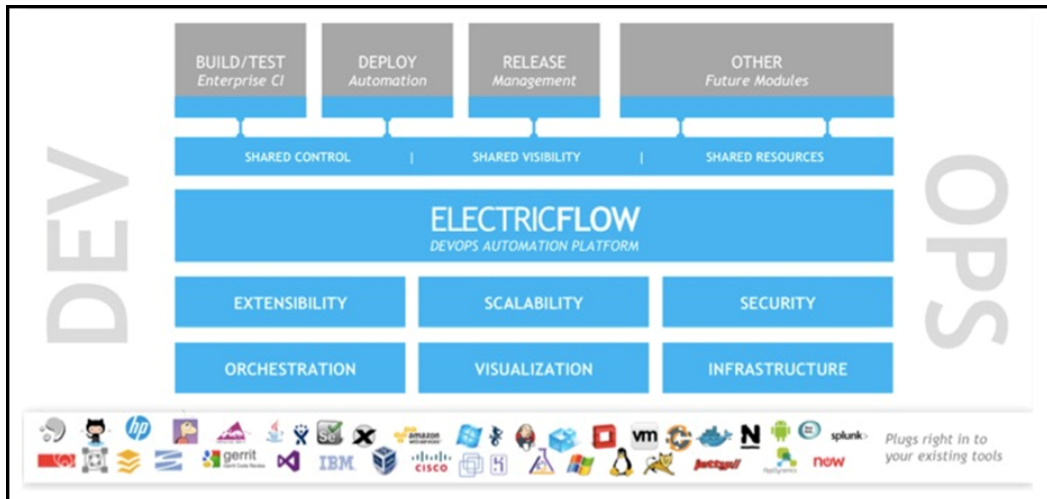
Chapter 1: Introduction to ElectricFlow

ElectricFlow™ (including the ElectricFlow Platform, formerly known as ElectricCommander) accelerates the continuous delivery of software and makes software delivery processes more repeatable, visible, scalable, and efficient. It provides domain-specific capabilities to automate the build, test, package, deploy, and release processes across many delivery pipelines.

ElectricFlow is built on a powerful proven automation platform that natively integrates domain-specific capabilities for Enterprise-level continuous delivery. The automation platform gives distributed DevOps teams shared control and visibility into infrastructure, tool chains, and processes. It accelerates and automates the software delivery process and enables agility, availability, predictability, and security across many build-test, deployment, and release pipelines.

The following diagram shows how ElectricFlow provides build/test, deploy, and release automation.

- ElectricFlow provides automation, management, and visibility of the build, test, deploy, and release processes by
 - Automating any workflows and pipelines.
 - Modeling and deploying one application for more than one use case.
 - Deploying all, some, or specific versions of artifacts in an application.
 - Keeping track of changes to tracked objects including applications, artifacts, jobs, resources, and workflows, referred to as Change Tracking.
 - Optimizing how resources are used in dynamic environments.
- ElectricFlow uses a process model to connect applications to environments.
- You create and manage resources, artifacts, projects, workflows, and procedures to support deployment and pipeline automation.



Unique Functionality

ElectricFlow is the most scalable solution on the market. Only ElectricFlow provides enterprise-class scalability for build and release management. It is easy to install and use on a simple build, yet scales to support the largest and most complex build and test processes. The ElectricFlow multi-threaded Java server provides efficient synchronization even under high job volume.

Facilities provided by ElectricFlow:

- **Complete end-to-end software deployment solution**

Automates standard deployment processes across your enterprise. You can select the components of the working applications in your software environment.

- **Workflow functionality**

Use Workflows to design and manage processes at a higher level than individual jobs. Workflows allow you to combine procedures into processes to create build-test-deploy lifecycles (for example). A workflow contains states and transitions you define to provide complete control over your workflow process. The ElectricFlow Workflow feature allows you to define an unlimited range of large or small lifecycle combinations to meet your needs.

- **Continuous Integration Manager (CI Manager)**

This feature provides a front-end user interface for creating, managing, and monitoring continuous integration builds. The CI Manager dashboard provides:

- Visually see your running builds, build progress, build status, and historical build outcomes.
- Easily accessed "Actions" to configure a continuous integration build.
- Quick configuration of your preferred SCM system.
- A project can contain any number of continuous integration builds, depending on the work you have already set up for your procedures/steps to perform.

- **Resource management**

If a resource is over committed, ElectricFlow delays some jobs until others have finished with the resource. You can define pools of equivalent resources and ElectricFlow balances the load across the pool.

- **Access control**

Users log into the system and ElectricFlow uses their information to control activities. Privileges can be set for individuals or groups to ensure the security you need.

- **Preflight Build functionality**

Developers can build and test code changes in isolation on their local machines before those changes are committed to a production build.

- **Search, sort, and filter functions**

Minimize the display of information that is of no interest to you, and quickly retrieve the information you need.

- **Detailed job information**

ElectricFlow records a variety of information about each job. You can view jobs and see step run times, successes, and failures.

- **Email notifications**

Get important information or data to individuals or groups immediately and on a regular basis for a particular job or a specific job aspect.

- **Powerful and flexible reporting facilities**

Various statistics such as the number of compiles or test errors are collected after each step and recorded in the ElectricFlow database. A variety of reports can be generated from this information.

- **Artifact management**

Use artifacts to improve performance across builds, provide better reusability of components, and improve cross-team collaboration with greater traceability. For example, instead of each developer repeatedly downloading third-party packages from external source, these components can be published and versioned as an artifact. A developer then simply retrieves a specific artifact version from a local repository, guaranteeing a consistent package from build to build.

- **ElectricFlow command-line tool**

All ElectricFlow features are available from a command-line tool (ectool), a Perl API (ec-perl), and a web interface.

- **Plugin capability**

ElectricFlow has an extensible UI which enables easy development of plugins. You can integrate with other tools, use custom dashboards, and create unique user experiences based on roles.

- **Workspaces**

ElectricFlow creates a workspace for each job. A workspace is a disk area jobs can use for storage.

- **Data models based on properties**

Properties are used to store job input data such as the source code branch to use for the build, to collect data during a job (such as number of errors or warnings), and to annotate the job after it completes (for example, a build has passed QA).

- **Zones and gateways**

A zone (or top-level network) is a way to partition a collection of agents to secure them from use by other groups. A gateway is a secured connection between two zones when you want to share or transfer information to another zone. For example, you might want a developers zone and a test zone. The ElectricFlow server is a member of the default zone, created during ElectricFlow installation.

Challenges Solved by ElectricFlow

Traditional software build processes face the following challenges:

- Wasted time on script-intensive, manual, home-grown systems

These systems are error prone, do not scale well, and have little or no management visibility or reporting.

- Multiple, disconnected build and test systems across locations

Disconnected build and test systems result in redundant work and the inability to share/reuse code files across teams, making it painful to manage build and test data.

- Slow overall build and release cycles

Slow cycle times directly impact release predictability and time-to-market.

ElectricFlow addresses these problems with a three-tier architecture, AJAX-powered web interface, and first-of-its-kind build and release analytic capabilities for reporting and compliance. With this solution, your developers, release engineers, build managers, QA teams, and managers gain:

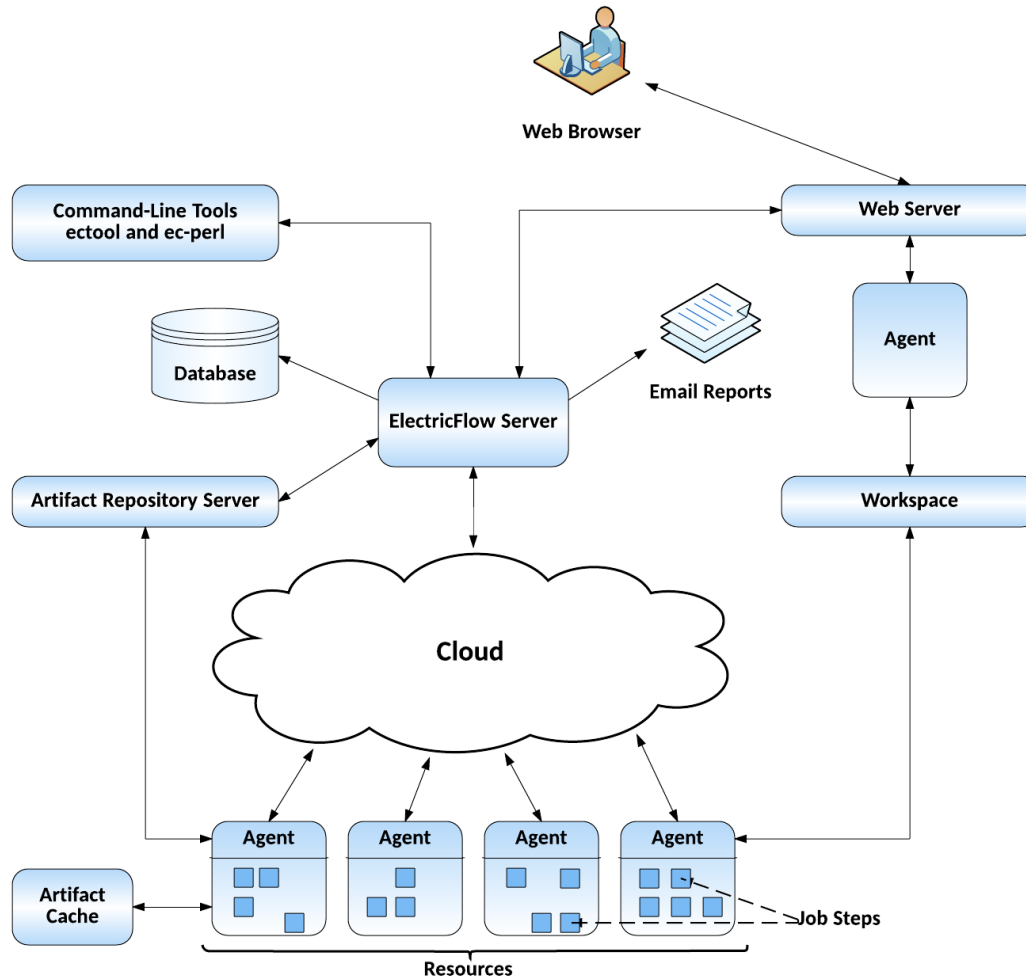
- A shared platform for disseminating best practices and reusing common procedures
- The ability to support geographically distributed teams
- Continuous integration and greater agility
- Faster throughput and more efficient hardware utilization
- Visibility and reporting for better project predictability
- Better software quality by integrating and validating against all target platforms and configurations

Architecture

ElectricFlow was architected from the ground up to support enterprise scale software production. Based on a three-tier architecture, ElectricFlow scales to handle large, complex environments. ElectricFlow's multi-threaded Java server provides efficient synchronization even under high job volume.

Local Configuration

The following diagram shows an ElectricFlow architecture configuration at a single site.



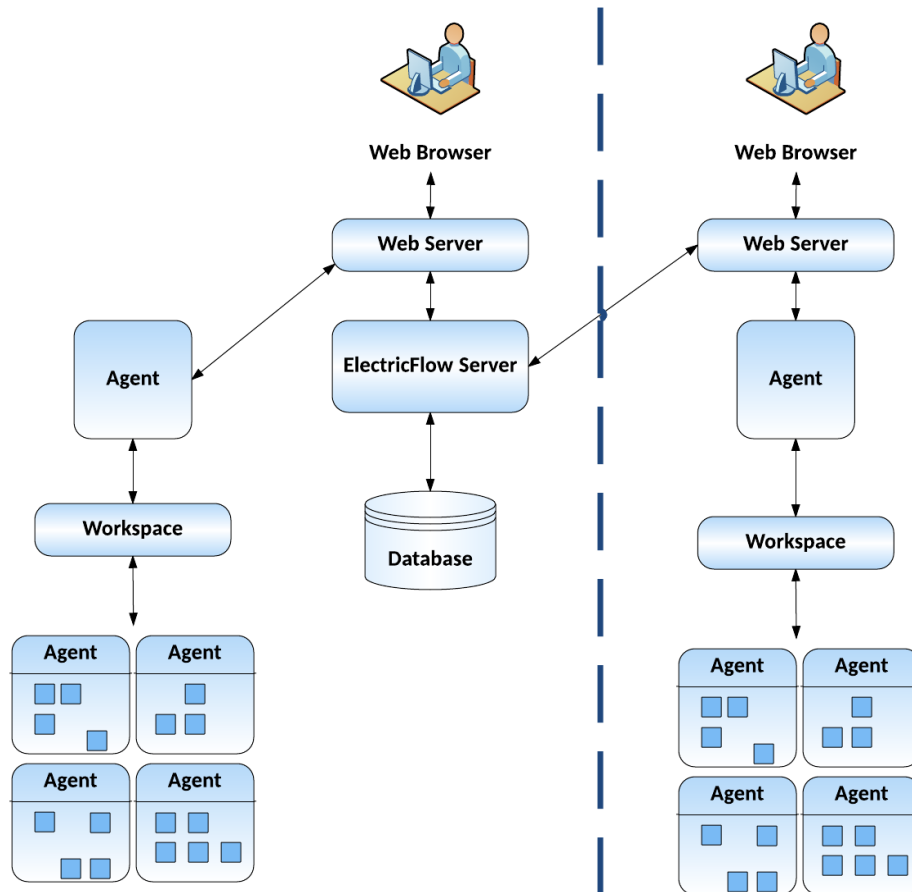
In the local configuration:

- The ElectricFlow server manages resources, issues commands, generates reports.
- An underlying database stores commands and metadata.
- Agents execute commands, monitor status, and collect results, in parallel across a cluster of servers for rapid throughput.

If you are only evaluating the ElectricFlow software, the database, ElectricFlow server, web server, and repository server can reside on the same machine. In a production environment, the database should reside on a separate machine from the ElectricFlow server to prevent performance issues. It is acceptable for the ElectricFlow server, web server, and repository server to reside on the same machine in a local configuration, but not required.

Remote Web Server Configuration

The following diagram shows an example of a remote web server architecture configuration.



In the example remote web server configuration:

- There are web servers at each site
- The database and ElectricFlow server is located at your headquarters
- Proxy resources exist at each site

This remote web server configuration helps prevent network latency. If you have multiple sites, ElectricFlow can be configured in numerous ways to help you work more efficiently.

ElectricFlow supports multiple workspaces, including those co-located on agents that use them. In this architecture, step log files are created locally so even the largest log files can be captured without a performance penalty. You can view the step log files remotely from the web UI, but there will be a performance penalty if the files must be retrieved across the WAN. This means that remote users will experience the performance penalty when the web server retrieves the step log file contents, and when the contents are sent back across the WAN to the browser.

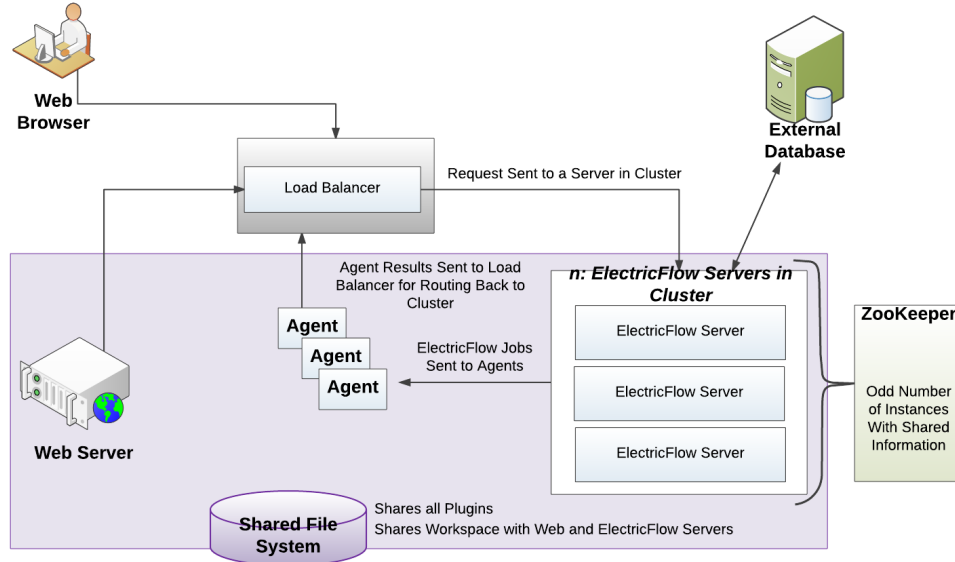
To minimize these performance issues, install one central ElectricFlow server, and then install an ElectricFlow web server at each remote site. The remote ElectricFlow web servers should be co-located with the remote agents and workspaces so remote users to log in through their local web server. Any operations initiated from the remote location, including running a job, are completed by the central ElectricFlow server. Job data is retrieved from the central server when a remote user views the Job Details page. If the job is using a workspace at the remote user's site, the links to all step log files will refer to local paths. The log files are only accessed by the remote web server's agent, and not the ElectricFlow server. This eliminates both trips across the WAN which

improves performance. The ElectricFlow web server reads the log file locally (via its agent), and then displays the page to the user whose browser is also on the same side of the WAN.

Note: You can install ElectricFlow web servers on any Windows or Linux platform suitable for installing the ElectricFlow server.

Clustered Configuration

The following diagram shows an ElectricFlow clustered configuration.



You can also add horizontal scalability and high availability to your ElectricFlow environment by adding additional machines to create a clustered ElectricFlow configuration.

A clustered ElectricFlow configuration has the following benefits:

- Add fault tolerance by re-routing jobs to running ElectricFlow servers
- Increase the supported number of simultaneous jobs and corresponding API requests
- Expand capacity over time by adding additional ElectricFlow servers
- Distribute API requests across multiple ElectricFlow servers
- Distribute ElectricFlow requests across multiple web servers

A clustered ElectricFlow configuration requires two additional software components:

- A centralized service for maintaining and synchronizing group services in cluster
- A load balancer for routing work to machines in the cluster

See [Clustering](#) on page 4-1 for additional details and clustered configuration set up procedures.

Chapter 2: System Requirements and Supported Platforms

This section describes hardware and software specifications and configurations for installing and running ElectricFlow on Windows or UNIX systems. All version requirements for operating systems and databases are routinely tested and fully supported by Electric Cloud. Contact Electric Cloud technical support if you have any questions regarding newer software versions.

Supported Server Platforms

This section describes the supported platforms for the ElectricFlow, web, or repository servers.

Windows Platforms

The following table lists all supported Microsoft Windows server platforms.

Platform	Notes
Windows 10 (64-bit)	—
Windows 8.1 (64-bit)	—
Windows 7 (64-bit)	<ul style="list-style-type: none">• Service Pack 1 is recommended.• An administrator might need to disable User Account Control (UAC). If the installer runs under account x, but services will run under account y, installation directories (both program and data) will probably have permissions that prevent y's access. This applies particularly to data directories.
Windows Server 2012 R2 (64-bit) Windows Server 2012 (64-bit) Windows Server 2008 R2 (64-bit) Windows Server 2008 (64-bit)	<ul style="list-style-type: none">• An administrator might need to disable User Account Control (UAC). If the installer runs under account x, but services will run under account y, installation directories (both program and data) will probably have permissions that prevent y's access. This applies particularly to data directories.

Linux Platforms

The following table lists all supported Linux server platforms.

Platform	Notes
CentOS 7 (64-bit)	<p>The following installation prerequisites apply to all ElectricFlow installers.</p> <p>Choosing the CentOS User</p> <p>Do not choose “nobody” for the CentOS user. CentOS does not allow a command such as <code>su - nobody -c foo.sh</code>, because it is not a shell account.</p> <p>Installing 32-Bit Libraries</p> <p>The ElectricFlow installation executable requires 32-bit libraries, which Ubuntu has omitted from its distribution. The installer checks whether the libraries are installed. If any are missing, the installer asks whether to automatically install them. You can also install the libraries before starting the installer by following the procedures below.</p> <p>For the full installer, 32-bit agent-only installer, or 64-bit agent-only installer on unsupported platforms, complete the following steps.</p> <ol style="list-style-type: none"> 1. Enter the following command: <pre>yum install libstdc++.i686</pre> <p>If you do not run this command, the ElectricFlow Apache server will not start, and the installer will fail silently.</p> 2. Enter the following command: <pre>yum install libuuid.i686</pre> <p>If you are installing ElectricFlow agents only (without a web server) you do not need to run this command on each agent machine.</p> 3. If you are using an LDAP account for ownership of the server, web, and repository services, install the 32-bit NSS packages by entering the following command: <pre>yum install nss-pam-ldapd*.i686</pre> <p>If you do not run this command, the ElectricFlow Apache server will not start.</p> <p>For the installer on a target machine that does not have internet access, complete the following steps.</p> <ol style="list-style-type: none"> 1. Download the packages to a machine that has internet access by entering the following commands: <pre>sudo su - cd /tmp yum install --downloadonly --downloadaddir=/tmp libstdc++.i686 yum install --downloadonly --downloadaddir=/tmp libuuid.i686 yum install --downloadonly --downloadaddir=/tmp nss-pam-ldapd*.i686</pre> 2. Copy the packages to the target machine.

Platform	Notes
	<p data-bbox="456 275 1354 306">3. Install the packages on the target machine by entering the following commands:</p> <pre data-bbox="537 331 1203 447">rpm -ivh libstdc++-4.4.7-16.el6.i686.rpm rpm -ivh libuuid-2.17.2-12.18.el6.i686.rpm rpm -ivh nss-pam-ldapd-0.7.5-20.el6_6.3.i686.rpm</pre>

Platform	Notes
Red Hat Enterprise Linux 7 (64-bit) Red Hat Enterprise Linux 6 (64-bit)	<p>The following installation prerequisites apply to all ElectricFlow installers.</p> <p>Choosing the RHEL User</p> <p>Do not choose “nobody” for the RHEL user. RHEL does not allow a command such as <code>su - nobody -c foo.sh</code>, because it is not a shell account.</p> <p>Installing 32-Bit Libraries</p> <p>The ElectricFlow installation executable requires 32-bit libraries, which Ubuntu has omitted from its distribution. The installer checks whether the libraries are installed. If any are missing, the installer asks whether to automatically install them. You can also install the libraries before starting the installer by following the procedures below.</p> <p>For the full installer, 32-bit agent-only installer, or 64-bit agent-only installer on unsupported platforms, complete the following steps.</p> <ol style="list-style-type: none"> 1. Enter the following command: <pre>yum install libstdc++.i686</pre> <p>If you do not run this command, the ElectricFlow Apache server will not start, and the installer will fail silently.</p> 2. Enter the following command: <pre>yum install libuuid.i686</pre> <p>If you are installing ElectricFlow agents only (without a web server) you do not need to run this command on each agent machine.</p> 3. If you are using an LDAP account for ownership of the server, web, and repository services, install the 32-bit NSS packages by entering the following command: <pre>yum install nss-pam-ldapd*.i686</pre> <p>If you do not run this command, the ElectricFlow Apache server will not start.</p> <p>For the installer on a target machine that does not have internet access, complete the following steps.</p> <ol style="list-style-type: none"> 1. Download the packages to a machine that has internet access by entering the following commands: <pre>sudo su - cd /tmp yum install --downloadonly --downloadaddir=/tmp libstdc++.i686 yum install --downloadonly --downloadaddir=/tmp libuuid.i686 yum install --downloadonly --downloadaddir=/tmp nss-pam-ldapd*.i686</pre> 2. Copy the packages to the target machine.

Platform	Notes
	<p>3. Install the packages on the target machine by entering the following commands:</p> <pre>rpm -ivh libstdc++-4.4.7-16.el6.i686.rpm rpm -ivh libuuid-2.17.2-12.18.el6.i686.rpm rpm -ivh nss-pam-ldapd-0.7.5-20.el6_6.3.i686.rpm</pre>
Red Hat Enterprise Linux 5 (64-bit)	<p>The following installation prerequisites apply to all ElectricFlow installers.</p> <ul style="list-style-type: none">• Do not choose “nobody” for the RHEL user. RHEL does not allow a command such as <code>su - nobody -c foo.sh</code>, because it is not a shell account.• To use ElectricFlow on RHEL5.x, the supported installation method requires the SELinux configuration to be less restrictive. Either run the <code># setenforce 0</code> command as the root user to temporarily set the SELinux operating mode to “permissive” (does not require a reboot) , or edit the <code>/etc/selinux/config</code> file and change line <code>SELINUX=enforcing</code> line to <code>SELINUX=permissive</code> (this change requires rebooting the Linux server to have the changes take effect).

Platform	Notes
Ubuntu 16.04 (64-bit)	The following installation prerequisites apply to all ElectricFlow installers.
Ubuntu 14.04 (64-bit)	<p>Choosing the Ubuntu User</p> <p>Do not choose “nobody” for the Ubuntu user. Ubuntu does not allow a command such as <code>su - nobody -c foo.sh</code>, because it is not a shell account.</p>
Ubuntu 12.04 (64-bit)	<p>Adding the bin Directory to the PATH Environment Variable</p> <p>Update <code>/etc/environment</code> to include the ElectricCommander <code>bin</code> directory in the PATH environment variable. Steps running with impersonation on Ubuntu use PATH that is set in <code>/etc/environment</code>. As a side-effect, the ElectricCommander <code>bin</code> directory is not in PATH in the impersonation context, so calls to tools such as <code>ectool</code> and <code>postp</code> fail with a “not found” error.</p>
Ubuntu 10.04 (64-bit)	<p>Fixing the "raise ValueError, 'need a file or string" Error</p> <p>If you receive an error during installation similar to the following:</p> <pre>File "/usr/lib/lsb/install_initd", line 3, in <module> import sys, re, os, initdutils File "/usr/lib/lsb/initdutils.py", line 18 raise ValueError, 'need a file or string' ^ SyntaxError: invalid syntax</pre> <p>run the following command:</p> <pre>sudo sed -i "s/python3/python/" /usr/lib/lsb/install_initd</pre> <p>This error is a known Ubuntu bug.</p> <p>Installing 32-Bit Libraries</p> <p>The ElectricFlow installation executable requires 32-bit libraries, which Ubuntu has omitted from its distribution. The installer checks whether the libraries are installed. If any are missing, the installer asks whether to automatically install them. You can also install the libraries before starting the installer by following the procedures below.</p> <p>For the full installer, 32-bit agent-only installer, or 64-bit agent-only installer on unsupported platforms, complete the following steps.</p>

Platform	Notes
	<ol style="list-style-type: none"> 1. Install the Ubuntu 32-bit compatibility layer by entering the following commands: <pre> sudo dpkg --add-architecture i386 sudo apt-get update sudo apt-get install lib32bz2-1.0 (Ubuntu 14.04, 12.04, or 10.04) or s udo apt-get install libbz2-1.0:i386 (Ubuntu 16.04) sudo apt-get install libuuid1:i386 </pre> 2. If you plan to use an LDAP account for ownership of the server, web, or repository services, install the 32-bit NSS packages by entering the following command: <pre> sudo apt-get update && sudo apt-get install libnss-ldap:i386 </pre> <p>If you do not run this command, the ElectricFlow Apache server will not start.</p> <p>For the installer on a target machine that does not have internet access, complete the following steps.</p> <ol style="list-style-type: none"> 1. Download the packages to a machine that has internet access by entering the following commands: <pre> sudo su - dpkg --add-architecture i386 apt-get update aptitude download lib32bz2-1.0 (Ubuntu 14.04, 12.04, or 10.04) or aptitude download lib32bz2-1.0:i386 (Ubuntu 16.04) aptitude download libuuid1:i386 aptitude download libnss-ldap:i386 </pre> 2. Copy the packages to the target machine. 3. Install the packages on the target machine by entering the following commands: <pre> dpkg -i --force-depends lib32bz2-1.0_1.0.6-5_amd64.deb (Ubuntu 14.04, 12.04, or 10.04) or dpkg -i --force-depends lib32bz2-1.0:i386_ 1.0.6-5_amd64.deb (Ubuntu 16.04) dpkg -i --force-depends libuuid1_2.20.1-5.1ubuntu20.7_i386. deb dpkg -i --force-depends libnss-ldap_264-2.2ubuntu4.14.04.1_ i386.deb </pre> <p>The <code>--force-depends</code> option is required to prevent the system from trying to install dependencies for packages.</p>

Supported Agent Platforms

The following table lists all supported agent platforms.

Platform	Notes
32-bit and 64-bit server machines specified for the ElectricFlow server	Go to Supported Server Platforms on page 2-1.
HP-UX 11i v1 (11.11) or later (Architecture: PA-RISC 2.0)	<ul style="list-style-type: none"> • Make sure that patches PHKL_29243 and PHSS_39077 (or patches superceding these patches) are installed. • HP-UX Secure Shell requires a random number generator on the system. It searches for <code>/dev/urandom</code> and then <code>/dev/random</code> and uses the first device it finds. If it fails to find them, it uses its own internal random number generator. By default, HP-UX 11i v2 systems includes these random number devices. They can also be obtained for HP-UX 11i v1 by downloading and installing the HP-UX Strong Random Number Generator from http://software.hp.com. HP recommends that Secure Shell users on HP-UX 11i v1 systems install the Strong Random Number Generator product, as it significantly speeds up program initialization and execution for some commands.
Mac OS X 10.4 (Tiger) or later (Architecture: Intel)	—
Sun Solaris 10 (Architecture: Sparc and Intel x86)	—
Sun Solaris 9 (Architecture: Sparc only)	—
SUSE Linux Enterprise Server 12.1 (32 and 64-bit)	Run <code>zypper install libstdc++6-32bit</code> before installing ElectricFlow agents on a 64-bit machine. This command installs the SUSE 32-bit libraries required for the ElectricFlow executable file to work.
SUSE Linux Enterprise Server 11.4 (32 and 64-bit)	—
SUSE Linux Enterprise Server 10.4 and 10.3 (32 and 64-bit)	—

Server and Agent Compatibility

Not all combinations of server version and agent version are supported. This is because of an ectool/Perl API communication incompatibility as well as a Diffie-Hellman key size incompatibility.

ectool/Perl API Incompatibility

Servers running ElectricFlow 5.0 or later no longer support agents running with versions 4.1 or earlier. The pre-4.2 ElectricCommander ectool/Perl API no longer communicates with the ElectricFlow server. You must upgrade any pre-4.2 Commander agents before you upgrade the ElectricFlow server.

Note: ElectricCommander 4.1 and earlier are no longer supported. Contact Electric Cloud technical support for manual instructions to upgrade your installation.

Diffie-Hellman Key Size Incompatibility

To enable ElectricFlow server versions 7.0 or newer to configure Diffie-Hellman cipher suites properly, ElectricFlow uses OpenSSL-1.0.1T with SSLv2 enabled. Because of OpenSSL and JRE changes, the minimum Diffie-Hellman key size requirement is increased to 1024 bits (from 768 bits) as of version 7.0.

Server versions 7.0 or newer use Jetty (a Java HTTP server), which listens on the 8000 (unsecure) and 8443 (secure) ports. Server versions 7.0 or newer use Java 1.8.0_66, in which the ephemeral DH key size defaults to 1024 bits during SSL/TLS handshaking in the SunJSSE provider.

For details on the increase of the key size requirement as of Java 1.6-u101, see the Java release note at <http://www.oracle.com/technetwork/java/javase/overview-156328.html#6u101-b31>. For details as of Java 1.7-u85, see the Java release note at <http://www.oracle.com/technetwork/java/javase/7u85-relnotes-2587591.html>.

Because their minimum key size is 1024 bits, agent versions 7.0 or newer can connect only to:

- Server versions 5.4, 6.0.1, or 6.5 or higher via ectool
- External applications that require SSL with a minimum key size of 1024 bits

However, ElectricCommander agents of versions 5.0.6, 5.3, or 5.4 and ElectricFlow agent versions 6.0.1 or 6.5 or newer can connect to all ElectricFlow server versions (including 7.0 or newer) via ectool and ec-perl.

ElectricCommander server versions 5.0.6 or 5.3 or newer can run jobs using all agent versions (including 7.0 or newer). ElectricFlow server versions 7.0 or newer can run jobs using ElectricCommander agent versions 5.0.6 or 5.3 or newer.

Hardware Requirements

This section lists the minimum requirements for any Windows or Linux machines installed with the ElectricFlow server software.

- Processor clock rate: 1.5 GHz or higher
- Memory: 4 GB available RAM or more (8 GB recommended for small to medium deployments)
- Processors: 2 or more (4 processors recommended for small to medium deployments)

Browser Requirements

ElectricFlow supports the following web browsers:

- Microsoft Internet Explorer 10.0 or later
- Mozilla Firefox 10.0 or later
- Chrome 13 or later

Note: Web browser extensions such as AdblockPlus for Google Chrome can interfere with the display of ElectricFlow web pages. We recommend disabling any ad blocking browser extensions or add an exclusion for ElectricFlow web pages.

Port Usage

ElectricFlow is pre-configured to use certain ports by default. This section contains the default port values and actions you can take to avoid port conflicts.

Note: Transport Layer Security (TLS) has replaced Secure Sockets Layer version 3.0 (SSLv3) on the ElectricFlow web server and the ElectricFlow server.

Default Server Ports

ElectricFlow servers use the following ports:

- 8000 - ElectricFlow Server
- 8443 - ElectricFlow Server (SSL port)
- 80 - ElectricFlow Web Server
- 443 - ElectricFlow Web Server (SSL port)
- 6800 - Port used by the ElectricFlow agent for HTTP communication on the localhost network interface.
- 7800 - ElectricFlow Agents (by default, this is an HTTPS port)
- 61613 - Preflight file transfer port, other file transfer, event notifications, or other messaging
- 8200 - Artifact repository server (by default, this is an HTTPS port)

Default Database Ports

Alternate databases use the following ports:

- 1521 - Oracle 11g Release 1, and R2; Oracle 12c
- 1433 - Microsoft SQL Server
- 3306 - MySQL

Default Ports for ElectricFlow Services

The Java Service Wrapper uses the following ports to communicate with a Java virtual machine (JVM):

- 127.0.0.1:32000—ElectricFlow Agents
- 127.0.0.1:32001—ElectricFlow Server
- 127.0.0.1:32002—Artifact Repository

Avoiding Port Conflicts

If you are installing an ElectricFlow server and you have a web server or other application that uses the same ports as the ElectricFlow host, you can choose one of the following actions:

- Select different web server or application ports
- Uninstall the existing web server or application
- Disable the existing web server or application
- Reconfigure the existing web server or application to use another port

IMPORTANT: Make sure your firewall is open for the ports used by ElectricFlow.

Database Requirements

You cannot log into ElectricFlow until a database is configured. During the ElectricFlow server installation, you can select the built-in (default) ElectricFlow database or an alternate database.

Note: If you are using two ElectricFlow servers, they cannot point to the same database.

Built-In Database

The built-in database is not supported for production systems, so you should use an alternate database in your production environment. Using an alternate database requires an ElectricFlow enterprise license.

You can switch to an alternate database at any time. For more information, see [Configuration](#) on page 5-1.

Supported Alternate Databases

Alternate databases must be configured to use UTF-8 encoding and configured to allow up to 200 open connections. ElectricFlow supports the following alternate databases:

- MySQL 5.5.12, 5.6, 5.7 or later (5.5.12 or later is recommended)
 - Clean installations of the ElectricFlow server version 4.2.3 or newer require the MySQL JDBC driver. See [Installing the MySQL JDBC Driver](#) on page 3-103.
 - For upgrades, additions to my.cnf/my.ini are required. See [Installing the MySQL JDBC Driver](#) on page 3-103.
- MS SQL Server 2008 and 2008 R2
- MS SQL Server 2012, 2014, and 2016
- Oracle 11g Release 2
- Oracle 12c

Note: In the Oracle database, set the OPEN_CURSORS parameter to at least 1000 to prevent ElectricFlow from running out of open cursors.

Database Sizing

Expected database growth over time can be correlated with the number of job steps created. Database growth is NOT correlated with build log or build artifact sizes.

To create a reasonable database growth estimate per period:

1. Estimate the number of jobs per period.
2. Multiply the "estimated number of jobs" by the number of steps estimated per job. This will determine the estimated number of steps per period.
3. Multiply the "estimated number of steps per period" by 10 to determine the disk size (in Kbytes) required per period.

For example: If you run 500 jobs per day with an average of 200 steps per job, you would run 100K steps per day. This means your database would grow about 1 GB per day or 90 GB per quarter. Using this example, if you prune jobs older than 30 days, database size could be maintained at about 30 GB.

Disk Usage

Disk space usage varies and depends on the quantity and size of the jobs you run. We recommend starting with the following free space recommendations:

- **Server**—10 GB
- **Agents**—5 GB each

Sizing artifact cache directory space on resources

By default, artifacts are retrieved into the `<DATA_DIRECTORY>/artifact-cache` directory of the agent installation. You can modify the `agent.conf` file to change the location, or you can specify the cache directory location on each resource known to ElectricFlow. Determining how much free space the cache partition needs to accommodate all of your artifact versions can be difficult. Here is one approach to approximate the disk/partition size you need:

For each artifact, estimate how large you think each version will be and how many versions you plan to keep. Compute the total required space to be the sum of `version-size * numVersions` for each artifact. Add a buffer of 50%. Using your end result, allocate a disk/partition that size and configure the cache as a directory on that disk/partition.

- **Repository server**—If using Artifact Management functionality, you could need 20-30 GB, or more disk space for your repository server.

Although a server install includes an artifact repository, We recommend that production repository servers be installed on different machines than the ElectricFlow server. The repository server might do a very large amount of disk and network I/O when transferring artifact versions to and from requesters and this might adversely affect ElectricFlow server performance.

Sizing the repository backingstore

For a repository installation, by default, the repository backingstore is the `<DATA_DIRECTORY>/repository-data` directory. You can modify the `<DATA_DIRECTORY>/conf/repository/server.properties` file or use `ecconfigure` to update the backingstore location. Determining exactly how much free space the backingstore disk/partition needs to accommodate your artifact versions can be difficult. Here is one approach to approximate the disk size you need:

For each artifact, estimate how large you think each version will be and how many versions you plan to keep. Compute the total required space to be the sum of `version-size * numVersions` for each artifact. Add a buffer of 50%. Using your end result, allocate a disk/partition that size and configure the repository backingstore as a directory on that disk/partition.

- **Logs**

You can set the following properties as Java system properties in `wrapper.conf`:

- `ec.logRoot` controls the location of the log output. The default location is the `logs/commander.log` directory.
- `ec.logHistory` controls the number of days of log history that is kept. The default is 30 days.
- `ec.logSize` controls the size of each log file before it is zipped up and a new log file started. The default is 100 MB, but each log rotation will zip the file, so that only about 6-7 MB of space are being taken.

Production systems generate multiple log files per day – an average system can generate 50-100 log files. This means that the daily requirement for space (under this type of load) is 300-700 MB. Retaining 1 months' worth of logs requires 9-21 GB of space, so adjusting the `ec.logHistory` value to something lower might be appropriate, if you want to allot less space for this logging.

To limit the amount of disk space for logging, the most effective approach is to use a lower `ec.logHistory` value.

Memory Usage

Memory usage varies depending on whether or not the ElectricFlow server is a dedicated machine.

- An ElectricFlow server running on a dedicated machine has a default minimum heap memory allocation of 20% and a maximum heap memory allocation of 40%. This applies to either a 32 or 64-bit system.
- In general, an ElectricFlow agent has a default minimum memory usage of 16 MB and a maximum memory usage of 64 MB. However, agents for REPO-server, Web-Server and Proxy agents needing higher settings; for details, see the [KBEC-00248 - Agent Memory Configuration](#) Knowledge Base article.

Modifying the Memory Settings for an ElectricFlow Server

There are two ways you can adjust the amount of memory for the ElectricFlow server.

- Modify the `wrapper.java.initmemory.percent` and `wrapper.java.maxmemory.percent` lines in `wrapper.conf`

Use the following table to determine the correct directory path.

Server Type	System	Path
Non-repository	Windows 2008	c:\ProgramData\Electric Cloud\ElectricCommander\conf\wrapper.conf
	Windows 7	
	Linux	/opt/electriccloud/electriccommander/conf/wrapper.conf
Repository	Windows 2008	c:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\wrapper.conf
	Windows 7	
	Linux	/opt/electriccloud/electriccommander/conf/repository/wrapper.conf

- Use `ecconfigure` to set the initial and maximum memory settings.

For example, to set the ElectricFlow Server initial memory percentage to 21% and the maximum memory percentage to 31%, enter the following command:

```
ecconfigure --serverInitMemory 21 --serverMaxMemory 31
```

Modifying the Memory Settings for an ElectricFlow Agent

To adjust the amount of memory for the ElectricFlow agent, modify the `wrapper.java.initmemory.percent` and `wrapper.java.maxmemory.percent` lines in `wrapper.conf` for the agent. Use the appropriate directory path:

- Windows: `C:\ProgramData\Electric Cloud\ElectricCommander\conf\agent\wrapper.conf`
- Linux: `/opt/electriccloud/electriccommander/conf/agent/wrapper.conf`

Note: For agents only, on a 32-bit Physical Address Extension (PAE) OS, Java allocates percentage of memory based on the physical memory, not the extended memory. The maximum memory for a 32-bit system is 4 GB, so a 20% allocation is about 1 GB.

Checksum Utility

An MD5 checksum file is available on the Electric Cloud FTP site. If you choose to verify that ElectricFlow 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 Windows, Linux, and Mac operating systems.

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

To use the MD5 checksum utility on a Mac:

1. In Finder, browse to `/Applications/Utilities`.
2. Double-click the Terminal icon. A Terminal window appears.
3. In the Terminal window, type: `"md5"` (followed by a space).
4. Drag the downloaded file from the Finder into the Terminal window.
5. Click in the Terminal window, press the Return key, and compare the checksum displayed on the screen to the one on the download page.

Software Licenses

The Licenses page in the ElectricFlow web interface shows you the current software use entitlements. You can view your license information by going to **Administration > Licenses** in the platform UI.

Standard License

The standard license allows you to use only the built-in database. In ElectricCommander 4.2 and later, *standard license* is packaged with the server and is automatically installed if no other license is present.

This is an example of an ElectricFlow server with a standard license.

Licenses

All Licenses

Import License

Company	Feature	Expiration	Grace Period (Days)	Actions
Electric Cloud, Inc.	Server (standard)		0	<div><div></div><div></div></div>

Current Usage

Maximum Concurrent Steps: 2

Concurrent Steps: 0

Maximum Concurrent Hosts: unlimited

Concurrent Hosts In Use: 0

Maximum Provided Hosts: unlimited

Provided Hosts In Use: 0

Maximum Concurrent Users: unlimited

Active Users: 0

The standard license restricts you to running only two concurrent steps.

Enterprise License



The enterprise license allows you to use an external (alternate) database. In ElectricCommander 4.2 and later, you must apply an enterprise license key when you want to configure an alternate database.

This is an example of an ElectricFlow server with an enterprise license.

Licenses

All Licenses

Import License

Company	Feature	Expiration	Grace Period (Days)	Actions	
Electric Cloud	Server (enterprise)	2016-01-07	0		

Current Usage

Maximum Concurrent Steps: unlimited

Concurrent Steps: 0

Maximum Concurrent Hosts: 60

Concurrent Hosts In Use: 0

Maximum Provided Hosts: 60

Provided Hosts In Use: 0

Maximum Concurrent Users: unlimited

Active Users: 0

Registered Host License

Registered hosts are dedicated hosts on which a procedure or step can always run. This license controls the number of unique registered host names.

Note: ElectricFlow can have a mixed license with concurrent and registered hosts.

Licenses

All Licenses

Import License

Company	Feature	Expiration	Grace Period (Days)	Actions
Electric Cloud	Server (freemium)	2015-10-31	0	<div><div></div><div></div></div>

Current Usage

Maximum Concurrent Steps: unlimited

Concurrent Steps: 0

Maximum Registered Hosts: 5

Registered Hosts In Use: 0

Maximum Concurrent Users: unlimited

Active Users: 0

Maximum Applications: 3

Applications In Use: 1

Legacy Software Licenses

Before ElectricCommander 4.2, *all licenses were enterprise licenses*. If you are a legacy customer, and you perform a clean server installation, you can install your original (pre-ElectricCommander 4.2) license (which is an enterprise license) on the new server and configure the server to talk to the production database. This approach is possible only when *no other server* is running and connected to the production database.

Note: If you want to install a new server (such as for testing), do *not* install your original license on the new server. For simple testing, you might be able to use the standard license on the test server, if your needs fit within the constraints of this license. However, if you need a production license for a test server, contact Electric Cloud Technical Support.

Chapter 3: Installing ElectricFlow

This section describes the methods for installing ElectricFlow in a new environment.

IMPORTANT: Product Name Change and Deprecation Notice

To bring focus to the ElectricFlow brand, the name “ElectricCommander” is changing to “ElectricFlow.” All ElectricCommander functionality is still available. All changes involving the new name are backward compatible. URLs and shortcuts and custom scripts will work without changes. For details, see the [FAQ](#).

The installation guide refers to the installer using the ElectricFlow name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, the installer file name will not contain “ElectricCommander” in the file name starting with ElectricFlow version 6.0.

IMPORTANT: To upgrade ElectricCommander or ElectricFlow, see [Roadmap to Upgrade ElectricFlow](#) on page 6-1.

IMPORTANT: The following situation might occur when the workspace files are in a directory other than the default *workspace* directory and the ElectricFlow configuration links to this workspace directory. When you install a new version, ElectricFlow creates a workspace directory in the default location. It does not recognize the preconfigured workspace link in the previous configuration.

When configuring ElectricFlow after upgrading it, you cannot use `ecconfigure` to move the workspace directory to the preconfigured network location. You must manually specify the link to the workspace directory in the new configuration.

Note: Although the ElectricFlow installer is in `$INSTALL_DIRECTORY/src`, do not launch it from there.

ElectricFlow Installer Files

The following installer files are available for the Windows and Linux platforms.

Type	What the Installer Does	Filename
Full	Installs all components, including the ElectricFlow server, web server, repository server, agents, and ElectricFlow tools. Only 32-bit agents are installed.	ElectricFlow-<version> Example: ElectricFlow-6.0.0.12345
32-bit agent only	Installs only the 32-bit agent.	ElectricFlowAgent-x86-<version> Example: ElectricFlowAgent-x86-6.0.0.12345
64-bit agent only	Installs only the 64-bit agent.	ElectricFlowAgent-x64-<version> Example: ElectricFlowAgent-x64-6.0.0.12345

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
```

```
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Choosing the Correct Installation Process

This section describes the various installation methods and available options for specific platform types.

For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms](#) on page 2-1 and [Supported Agent Platforms](#) on page 2-10.

User Interface Installations

These methods provide an installation Wizard you use to install the ElectricFlow software on a supported server platform. These installation methods are generally preferred by Windows users, but are supported on Linux platforms with X installed.

Installation options:

- **Express Server**

This option installs the ElectricFlow server, web server, repository server, built-in database, one agent (for running jobs), and ElectricFlow tools on one machine with the default ElectricFlow server settings. This option is available via the "full" installer file (see [ElectricFlow Installer Files](#) on page 3-1). This installation method is most useful for quickly installing the ElectricFlow software for evaluation purposes.

IMPORTANT:

Running ElectricFlow on a single machine with the built-in database is generally not recommended for a production environment. ElectricFlow will connect to an external database in a typical production configuration.

If ElectricFlow was installed with the built-in database, you can reconfigure it to use an external database. Go to [External Database Configuration](#) on page 5-2 for configuration instructions.

- **Advanced**

This installation option allows you to select individual components, directories, or ports. This option is available via the "full" installer file (see [ElectricFlow Installer Files](#) on page 3-1). You can use this option to install the ElectricFlow server, web server, and repository server on a machine without the built-in database.

- **Express Agent**

This option quickly installs an ElectricFlow agent and ElectricFlow tools. This option is available via the "full" installer file (see [ElectricFlow Installer Files](#) on page 3-1). Use this option for managed hosts where you intend to run job steps. This installation method is useful if you need to install an individual agent. If you need to install multiple agents, you might prefer to use the [Silent Unattended Installation](#) on page 3-5.

- **32-Bit Agent-Only**

This option quickly installs a 32-bit ElectricFlow agent. This option is available via the 32-bit agent-only installer file (see [ElectricFlow Installer Files](#) on page 3-1). Use this option for managed hosts where you intend to run job steps. This installation method is useful if you need to install an individual agent. If you need to install multiple agents, you might prefer to use the [Silent Unattended Installation](#) on page 3-5.

- **64-Bit Agent-Only**

This option quickly installs a 64-bit ElectricFlow agent. This option is available via the 64-bit agent-only installer file (see [ElectricFlow Installer Files](#) on page 3-1). Use this option for managed hosts where you intend to run job steps. This installation method is useful if you need to install an individual agent. If you need to install multiple agents, you might prefer to use the [Silent Unattended Installation](#) on page 3-5.

Interactive Command-Line Installation

These methods provide an interactive command line for installing the ElectricFlow software on a supported server platform. These installation methods are only available for Linux platforms.

Installation options:

- **Express Server**

This option installs the ElectricFlow server, web server, repository server, built-in database, one agent (for running jobs), and ElectricFlow tools on one machine with the default ElectricFlow server settings. This option is available via the "full" installer file (see [ElectricFlow Installer Files](#) on page 3-1). This installation method is most useful for quickly installing the ElectricFlow software for evaluation purposes.

IMPORTANT:

Running ElectricFlow on a single machine with the built-in database is generally not recommended for a production environment. ElectricFlow will connect to an external database in a typical production configuration.

If ElectricFlow was installed with the built-in database, you can reconfigure it to use an external database. Go to [External Database Configuration](#) for the configuration instructions.

- **Advanced**

This installation option allows you to select individual components, directories, or ports. This option is available via the "full" installer file (see [ElectricFlow Installer Files](#) on page 3-1). You can use this option to install the ElectricFlow server, web server, and repository server on a machine without the built-in database.

- **Express Agent**

This option quickly installs an ElectricFlow agent and ElectricFlow tools. This option is available via the "full" installer file (see [ElectricFlow Installer Files](#) on page 3-1). Use this option for managed hosts where you intend to run job steps. This installation method is useful if you need to install an individual agent. If you need to install multiple agents, you might prefer to use the method called [Silent Unattended Installation](#) on page 3-5.

- **32-Bit Agent-Only**

This option quickly installs a 32-bit ElectricFlow agent. This option is available via the 32-bit agent-only installer file (see [ElectricFlow Installer Files](#) on page 3-1). Use this option for managed hosts where you intend to run job steps. This installation method is useful if you need to install an individual agent. If you need to install multiple agents, you might prefer to use the [Silent Unattended Installation](#) on page 3-5.

- **64-Bit Agent-Only**

This option quickly installs a 64-bit ElectricFlow agent. This option is available via the 64-bit agent-only installer file (see [ElectricFlow Installer Files](#) on page 3-1). Use this option for managed hosts where you intend to run job steps. This installation method is useful if you need to install an individual agent. If you need to install multiple agents, you might prefer to use the [Silent Unattended Installation](#) on page 3-5.

Silent Unattended Installation

This method provides a non-interactive command-line installation for supported server platforms. This option is available via the "full" installer file, the 32-bit installer file, or the 64-bit installer file (see [ElectricFlow Installer Files](#) on page 3-1). You might find this installation method preferable for installing multiple remote agents and servers.

- **Windows**

This option is only for Windows platforms.

- **Linux**

This option is only for Linux platforms.

Non-Server Platform Agent Installation

This method provides a command-line for installing the ElectricFlow agent and tool software on supported non-server platforms. This installation method is only supported on non-server platforms.

Installation options:

- **Command-Line Agent**

This option allows you to install an agent from a UNIX command-line installer.

- **Silent Agent**

This option allows you to run unattended (silent) installations with the UNIX installer.

Before You Install ElectricFlow

Review the following information before attempting to install any ElectricFlow software.

Linux and Windows Installations

- You must install ElectricFlow on a local drive. Electric Cloud does not support installing the ElectricFlow server on a network volume.
- We recommend installing the ElectricFlow server first, before installing remote agents or web servers.
- Make sure you have completed any prerequisite platform setup.

For more information, see [Supported Server Platforms](#) on page 2-1 and [Supported Agent Platforms](#) on page 2-10.

- The ElectricFlow-provided built-in database is **not** recommended for production systems. We strongly recommend using an alternate database in your production ElectricFlow environment. Additionally, the built-in database is **not** supported in a clustered ElectricFlow configuration. See [Supported Alternate Databases](#) on page 2-13 for alternate databases supported by ElectricFlow.
- If the X Window System is not running or not available, the Linux user interface installer will run in interactive command-line mode.
- When installing a 64-bit machine, the 64-bit version of the Java Runtime Environment is installed automatically.
- When installing an agent, repository server, and/or web server, you can enter information for a remote ElectricFlow server. That information can then be used to discover the server's plugins directory and set it accordingly so the local install is in sync with the remote ElectricFlow server. During an agent installation, you can create a resource object on the server automatically. Similarly, during a repository installation, you can create a repository object on the server automatically.
- If you plan to use a clustered ElectricFlow configuration, see [Clustering](#) on page 4-1 to read about additional requirements and considerations.

Linux Installations

Review the following information before installing ElectricFlow on a Linux machine.

64-Bit Linux Platforms

The Linux installer checks for and installs the required 32-bit compatibility libraries on supported 64-bit Linux platforms. When you (as the root user) are installing ElectricFlow on a 64-bit machine running a supported Linux platform, the installer checks if the required 32-bit libraries are installed on it during the installation process. If the 32-bit libraries are already installed, the installation process continues to the next step. However, if the 32-bit libraries are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686
nss-pam-ldapd.*i686. They are required for correct work of ElectricFlow.
Install them? [n/Y]
```

- If you enter `y`, the software installs the missing 32-bit libraries, and then continues to the next step.
- If you enter `n`, the installation process stops (aborts).

The installer does not check for the required 32-bit libraries in these situations:

- The machines run on unsupported Linux platforms such as Debian, CentOS, or Fedora.
- The system is not connected to the internet.

Full Installer and 32-Bit Agent Installer

The ElectricFlow full and 32-bit installers are 32-bit executables that do not check if the machine has the required 32-bit compatibility libraries during the installation process.

Before installing ElectricFlow agents on a RHEL machine, you may need to complete this prerequisite:

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the

ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.
- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.
- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

Before installing Electric Flow agents on a Ubuntu machine, you may need to complete this prerequisite:

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Unsupported Linux Platforms

For platforms such as Debian, CentOS, or Fedora, install the following 32-bit libraries before installing ElectricFlow. They are required for the ElectricFlow installation executable file to work. We recommend installing *all* of these libraries on your 64-bit machines.

- `libstdc++.i686`: If you do not install this, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.
- `libuuid.i686`: Install this if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.
- `nss-pam-ldapd*.i686`: Install the 32-bit NSS packages if you are using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

No Internet Connection

These are alternate ways to download and install the 32-bit libraries:

- Ubuntu: Create an offline copy of the repository and manage it offline as described in <https://help.ubuntu.com/community/AptGet/Offline/Repository/>
- Red Hat Enterprise Linux (RHEL): Manage the yum repositories as described in https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/sec-Configuring_Yum_and_Yum_Repositories.html

Default Installation Directories

ElectricFlow uses the following default installation directories:

Platform	Data Type	Default Path
Windows	Program files	C:\Program Files\Electric Cloud\ElectricCommander
	Data (database, logs, configuration files)	Windows 2008 or Windows 7: C:\ProgramData\Electric Cloud\ElectricCommander
UNIX/Mac	All program files and data	/opt/electriccloud/electriccommander

Note: You can change the installation directories when you install the ElectricFlow software.

Graphical User Interface Installation Methods

The graphical user interface installation methods are supported by Windows platforms and Linux platforms running the X Window System.

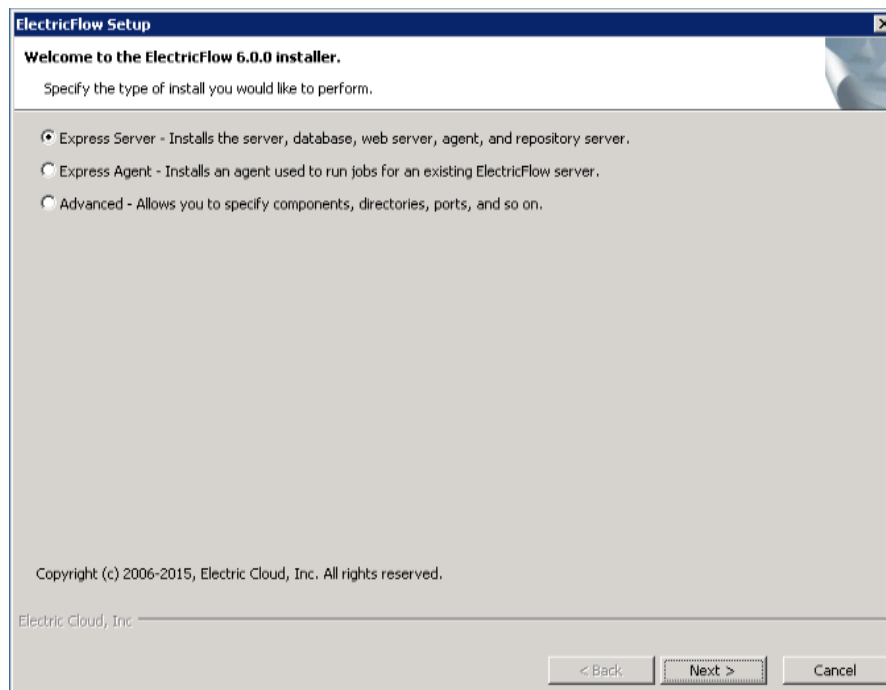
Running an Express Server Graphical User Interface Installation

The express server installation installs the ElectricFlow server, including the web server, built-in database, agent for running jobs, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

Note: The built-in database is not supported in a clustered ElectricFlow configuration.

1. Double-click the `ElectricFlow-<version>` file to run the full installer.

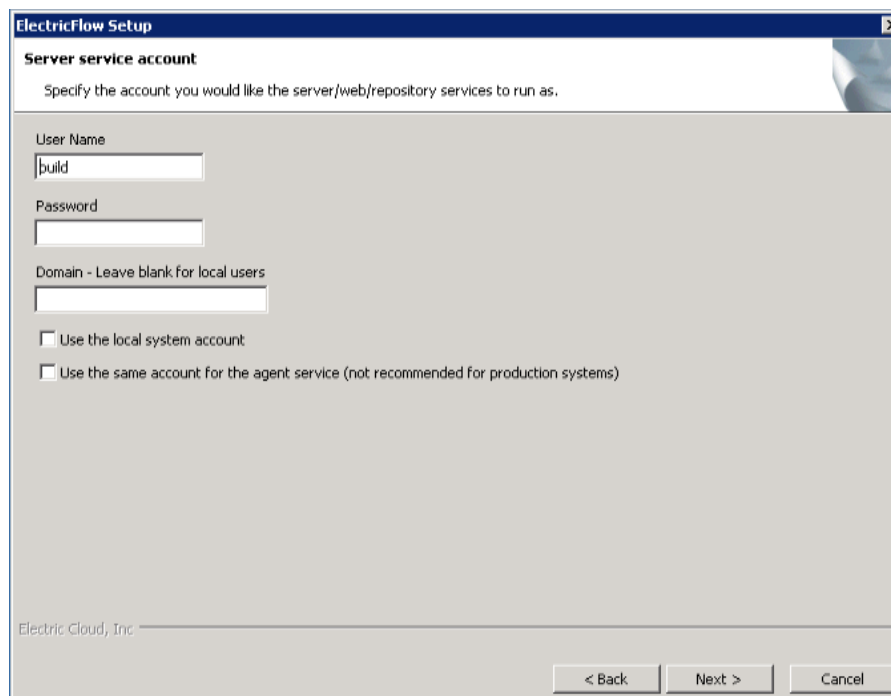
The Welcome to the ElectricFlow Installer screen appears.



Note: The screen examples in this procedure are from a Windows system. Different options will appear in some windows on a Linux system.

2. Select the **Express Server** installation option, and then click **Next** to continue.

The Server Service Account screen appears.



3. Select the appropriate step for your platform and complete the information for the server service account.

○ If you have a Windows system:

- **User Name** - Use this field to enter the name of the user who will run the ElectricFlow server, web server, and repository server services.
- **Password** - Use this field to enter the password of the user who will run the ElectricFlow server, web server, and repository server services.
- **Domain** - Use this field to enter the domain name information for the user. For example, electric-cloud.com. Leave this field blank if this is a local user.
- **Use the local system account** - Select this check box if you want the ElectricFlow server, repository server, and web server services to run as the local Windows system account.

Note: The local system account cannot access network resources such as shared filesystems used for plugins or workspaces.

- **Use the same account for the agent service** - Select this check box if you want the agent on the ElectricFlow server machine to run as the same account.

For security reasons in production environments, you may want to use a separate account for the agent service because the server account has permission to read the key file (/opt/electriccloud/electriccommander/conf/passkey in Linux or C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey in Windows). The key file is used to decrypt passwords stored in ElectricFlow. Using a different account for the agent service ensures that a process running on the agent cannot gain access to the key file.

○ If you have a Linux system:

- **User Name** - Use this field to enter the name of the user who owns the ElectricFlow server, repository server, and web server processes.
- **Group Name** - Use this field to enter the name of the group who owns the ElectricFlow server, repository server, and web server processes.
- **Use the same account for the agent service** - Select this check box if you want the same user and group to own the agent process on the ElectricFlow server machine.

For security reasons in production environments, you might want to use a separate user and group for the agent service because the server service has permission to read the key file (/opt/electriccloud/electriccommander/conf/passkey in Linux or C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey in Windows). The key file is used to decrypt passwords stored in ElectricFlow. Using a different user and group for the agent service ensures that a process running on the agent cannot gain access to the key file.

4. Click **Next** to continue.

The Agent Service Account screen appears.

IMPORTANT: If you selected the **Use the same account for the agent service** check box on the previous screen, you will not see this screen.

5. Select the appropriate step for your platform and complete the information for the agent service account.

- If you have a Windows system:

- **User Name** - Use this field to enter the name of the user who will run the ElectricFlow agent service.

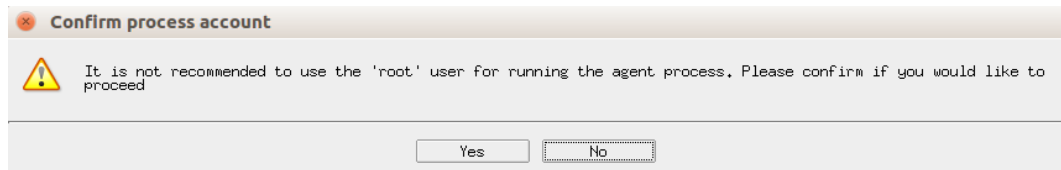
The user that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory.

- **Password** - Use this field to enter the password of the user who will run the ElectricFlow agent service.
- **Domain** - Use this field to enter the domain name information for the user. For example, `electric-cloud.com`. Leave this field blank if this is a local user.
- **Use the local system account** - Select this check box if you want the ElectricFlow agent service to run as the local Windows system account.

Note: The local system account does not have access to network shares.

- If you have a Linux system:
 - **User Name** - Use this field to enter the name of the user who owns the ElectricFlow agent process.

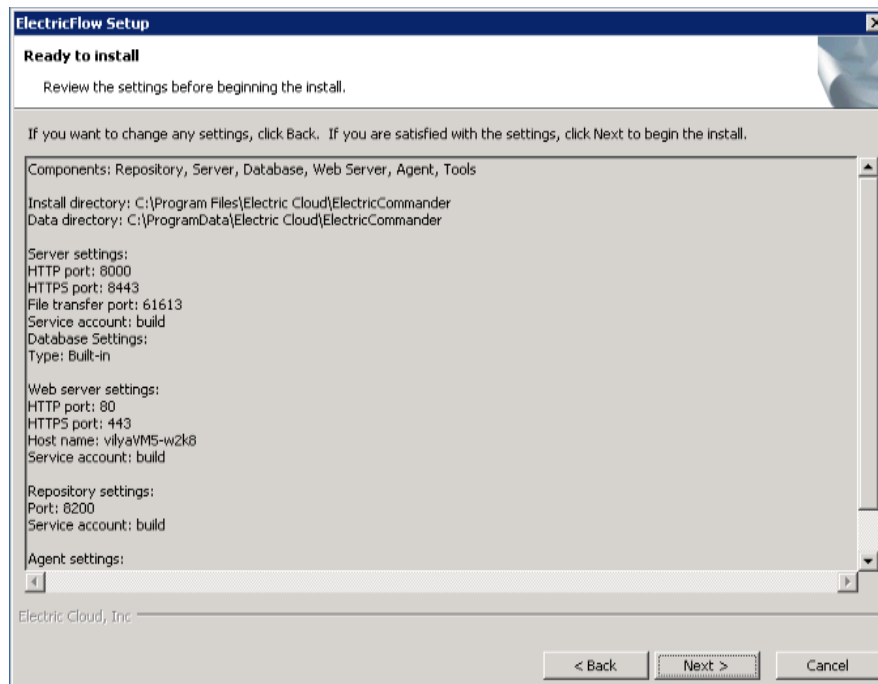
The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, click **Yes** when the following confirmation appears:



- **Group Name** - Use this field to enter the name of the group that owns the ElectricFlow agent process.

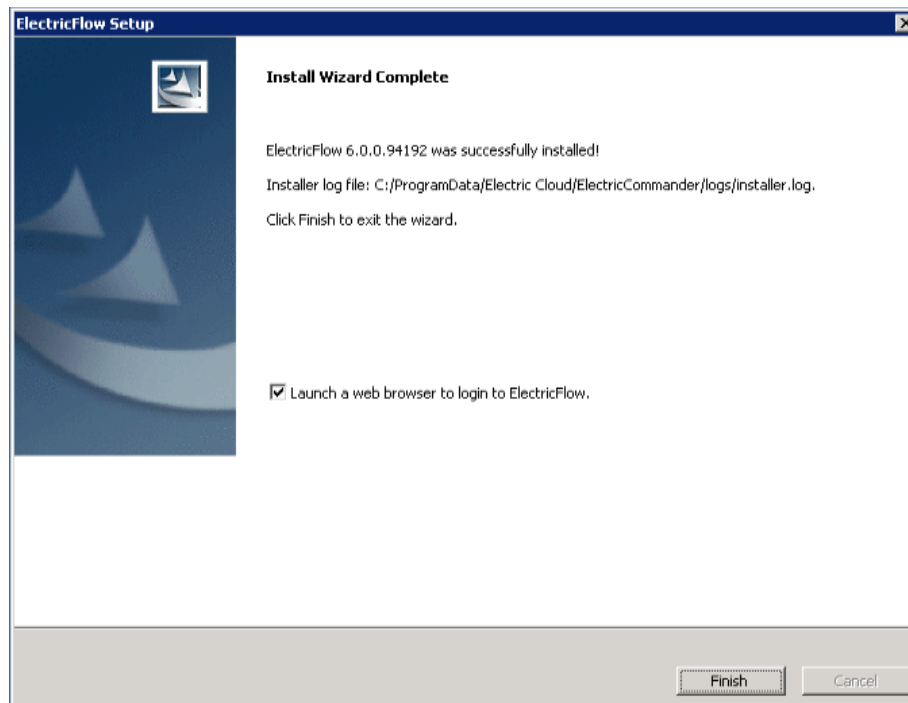
6. Click **Next** to continue.

The Ready to Install screen appears.



7. Review the default settings and your service account selections. Use the **Back** button to change your service account selections if necessary.
8. Click **Next** to continue.

The installer displays a status bar to show the progress of the installation process which can take fifteen minutes. When the install process is complete, the Install Wizard Complete screen appears.



Note: The ElectricFlow server will automatically start when installation is complete.

9. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
10. Click **Finish** to close the wizard.

Running an Advanced Graphical User Interface Installation

The advanced graphical user interface installation allows you to install individual ElectricFlow components such as an ElectricFlow server, web server, repository server, or ElectricFlow tools on specific machines. You can also change the default installation settings to accommodate your environment. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

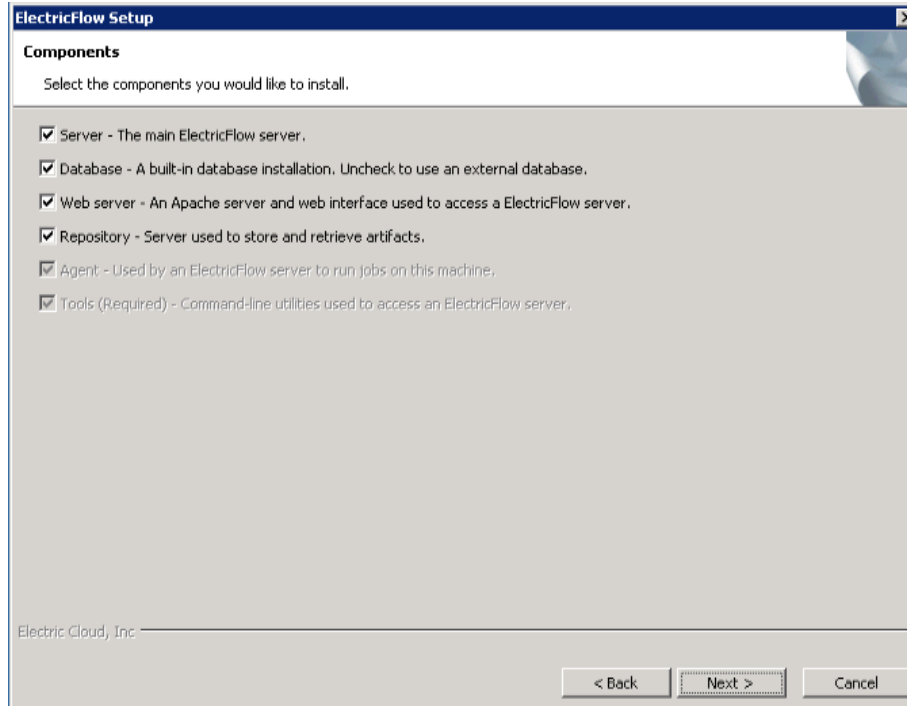
1. Double-click the `ElectricFlow-<version>` file to run the installer.

The Welcome to the ElectricFlow Installer screen appears.

Note: The screen examples in this procedure are from a Windows system. Different options will appear in some windows on a Linux system.

2. Select the **Advanced** installation option, and then click **Next** to continue.

The Components screen appears. All options are selected by default.



3. Clear the check boxes for any servers you do **NOT** want to install on the current machine. For more information, see [Architecture](#) on page 1-4.

Available options are:

Server - Select this check box if you want to install an ElectricFlow server.

Database - Only select this check box if you want to install the built-in database. This is not recommended for production systems. Additionally, the built-in database is **not** supported in a clustered ElectricFlow configuration. Clear this check box if you plan to use an external database. If you plan to use MySQL, see [Installing the MySQL JDBC Driver](#) on page 3-103.

Web server - Select this check box if you want to install an Apache web server.

Repository - Select this check box if you want to install an ElectricFlow repository server. If you select this option, a single agent is automatically installed on the machine to run jobs.

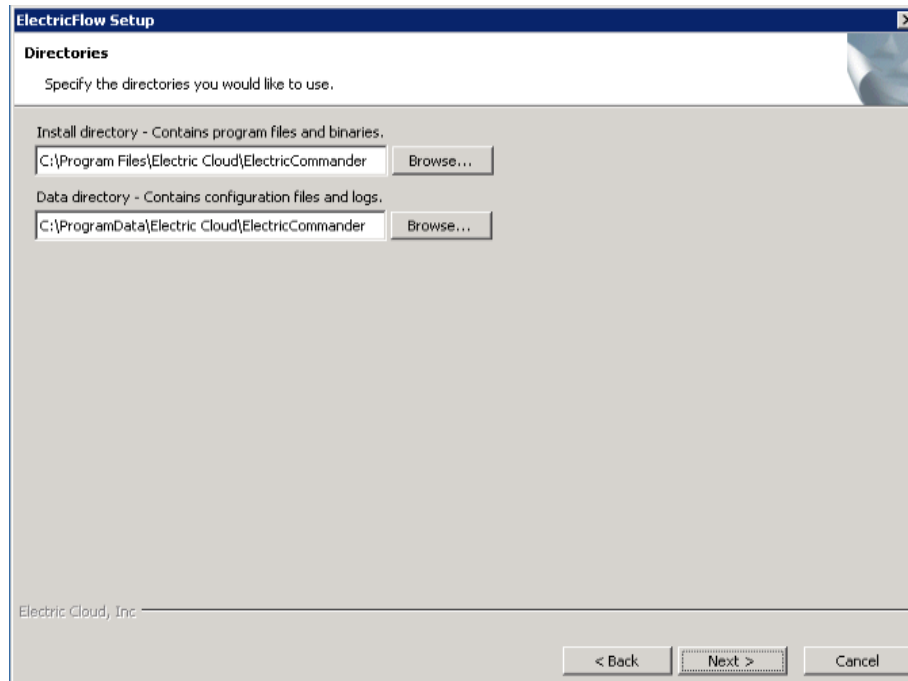
Agent - Select this check box if you want to install ElectricFlow agent software.

Tools - Select this check box if you want to install ElectricFlow tools. If you only want to install the ElectricFlow tools, clear all the check boxes. This option does not automatically install an ElectricFlow agent like the other options.

IMPORTANT: The following installation screens will vary depending on which servers you choose to install. **You might see all, some, or none of the screens in the following steps.**

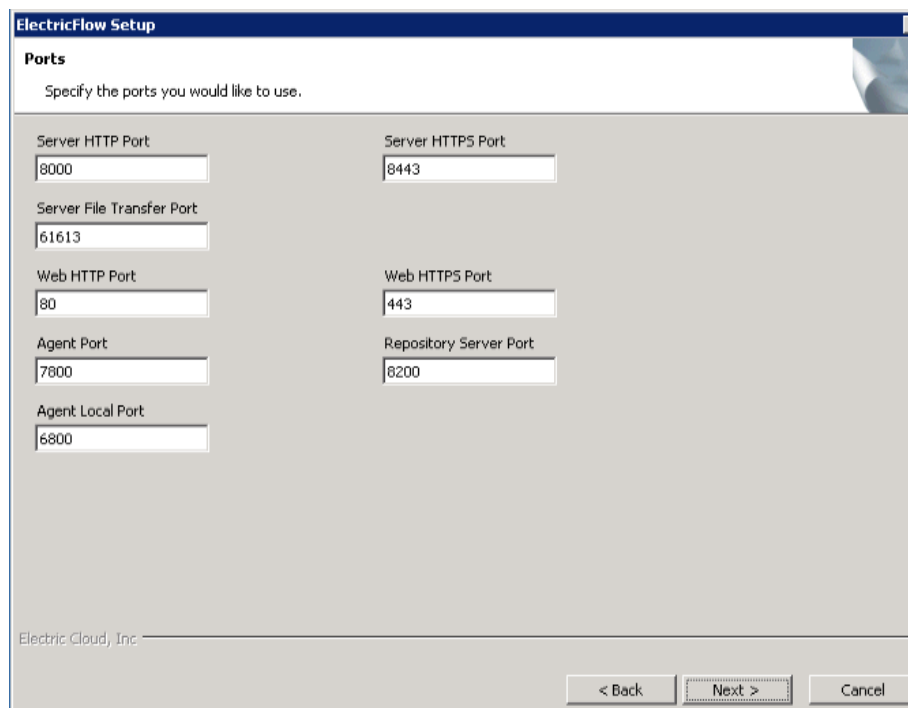
4. Click **Next** to accept the component selections.

The Directories screen appears. ElectricFlow uses the default directories to install files and components.



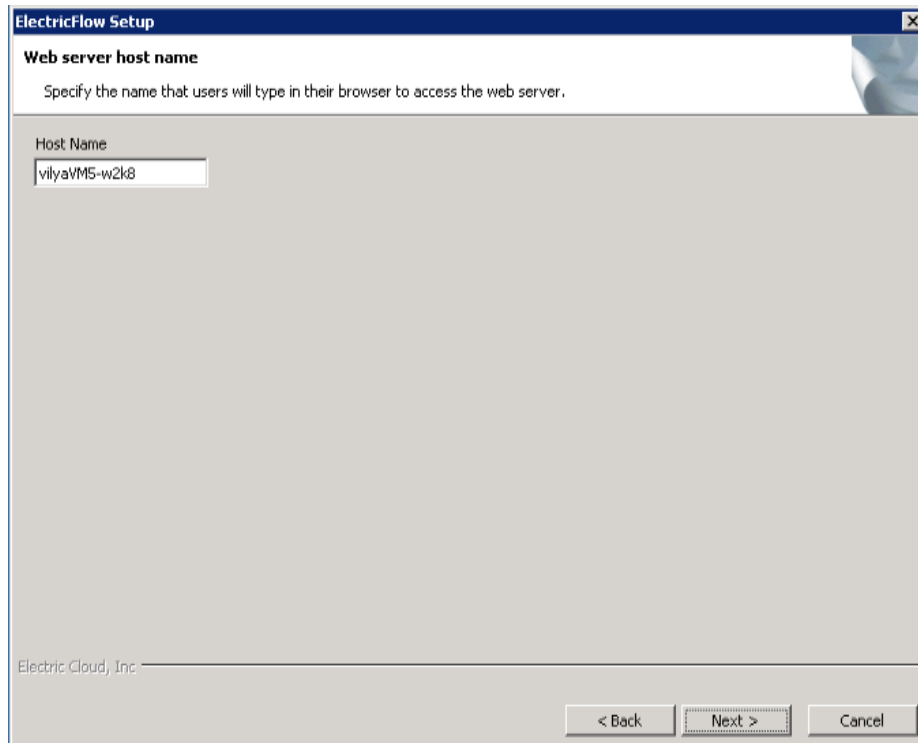
5. Click **Next** to continue, or click **Browse** to specify different directory locations.

The Ports screen with the default ElectricFlow port values appears if you are installing an ElectricFlow, web, or repository server.



6. Complete the information for the Ports screen, and click **Next** to continue. You can enter alternate port numbers if you need to specify different port values.

The Web Server Host Name screen appears if you are installing a web server.



ElectricFlow Setup

Web server host name

Specify the name that users will type in their browser to access the web server.

Host Name

vilyaVMS-w2k8

Electric Cloud, Inc.

< Back Next > Cancel

7. Complete the information for the Web Server Host Name screen, and click **Next** to continue. The host name is the name users must enter in their browser to access the ElectricFlow web server.

The Server Service Account screen appears if you are installing an ElectricFlow, web, or repository server.

8. Complete the information on the Server Service Account screen, and click **Next** to continue.

- If you have a Windows system:

- **User Name** - Use this field to enter the name of the user who will run the ElectricFlow server, web server, and repository server services.
- **Password** - Use this field to enter the password of the user who will run the ElectricFlow server, web server, and repository server services.
- **Domain** - Use this field to enter the domain name information for the user. For example, electric-cloud.com. Leave this field blank if this is a local user.
- **Use the local system account** - Select this check box if you want the ElectricFlow server, repository server, and web server services to run as the local Windows system account.

Note: The local system account cannot access network resources such as shared filesystems used for plugins or workspaces.

- **Use the same account for the agent service** - Select this check box if you want the agent on the ElectricFlow server machine to run as the same account.

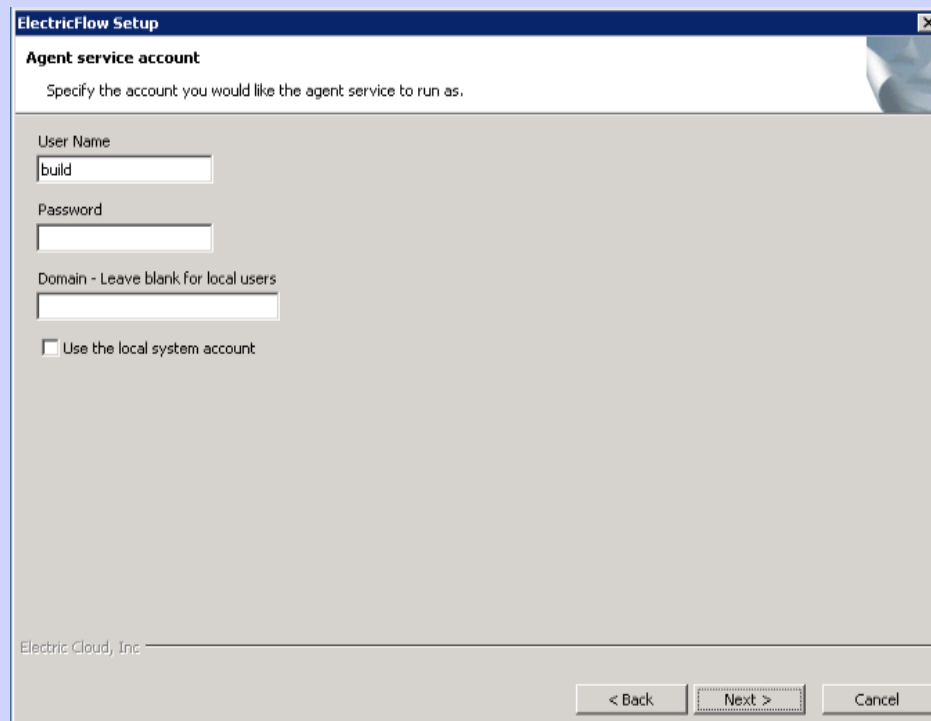
For security reasons in production environments, you might want to use a separate account for the agent service because the server account has permission to read the key file (/opt/electriccloud/electriccommander/conf/passkey in Linux or C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey in Windows). The key file is used to decrypt passwords stored in ElectricFlow. Using a different account for the agent service ensures that a process running on the agent cannot gain access to the key file.

- If you have a Linux system:
 - **User Name** - Use this field to enter the name of the user who owns the ElectricFlow server, repository server, and web server processes.
 - **Group Name** - Use this field to enter the name of the group who owns the ElectricFlow server, repository server, and web server processes.
 - **Use the same account for the agent service** - Select this check box if you want the same user and group to own the agent process on the ElectricFlow server machine.

For security reasons in production environments, you might want to use a separate user and group for the agent service because the server service has permission to read the key file (`/opt/electriccloud/electriccommander/conf/passkey` in Linux or `C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey` in Windows). The key file is used to decrypt passwords stored in ElectricFlow. Using a different user and group for the agent service ensures that a process running on the agent cannot gain access to the key file.

The Agent Service Account screen appears if you are installing an agent. An agent is automatically installed on the machine to run jobs if you are installing a web or repository server.

IMPORTANT: If you selected the **Use the same account for the agent service** check box on the previous screen, you will not see the fields to supply your agent service account information.



9. Complete the information on the Agent Service Account screen, and click **Next** to continue.

- If you have a Windows system:

- **User Name** - Use this field to enter the name of the user who will run the ElectricFlow agent service.

The user that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory.

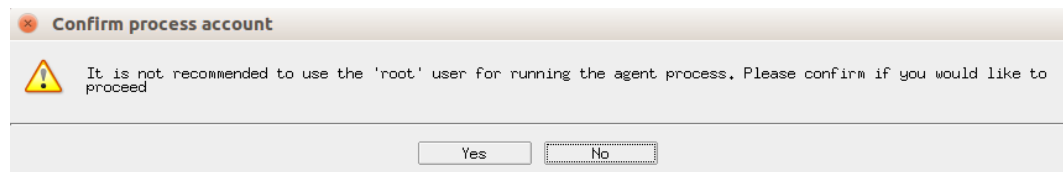
- **Password** - Use this field to enter the password of the user who will run the ElectricFlow agent service.
- **Domain** - Use this field to enter the domain name information for the user. For example, `electric-cloud.com`. Leave this field blank if this is a local user.
- **Use the local system account** - Select this check box if you want the ElectricFlow agent service to run as the local Windows system account.

Note: The local system account does not have access to network shares.

- If you have a Linux system:

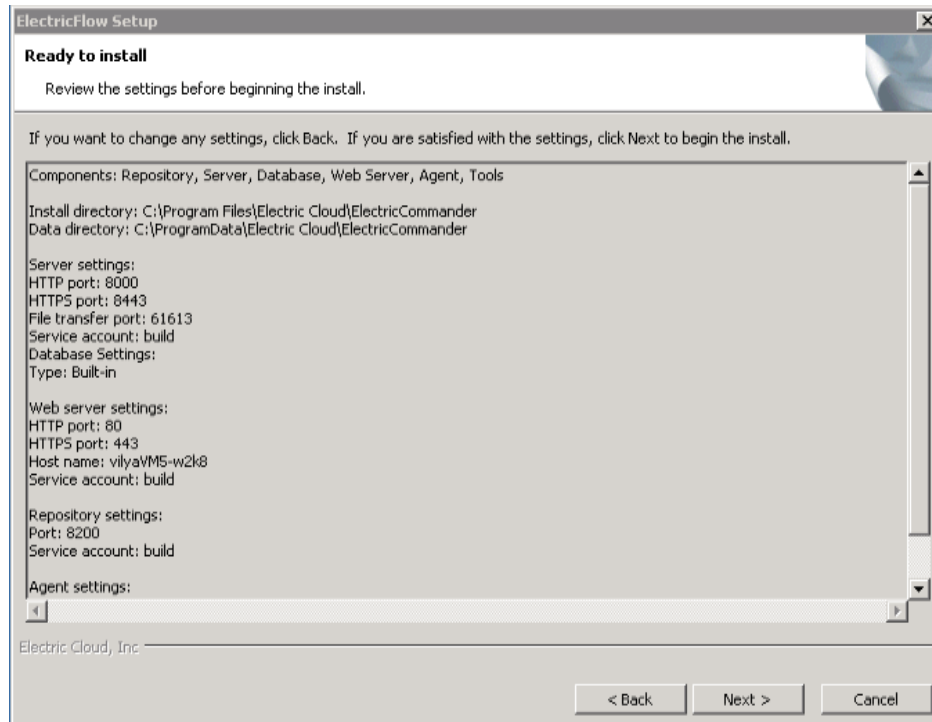
- **User Name** - Use this field to enter the name of the user who owns the ElectricFlow agent process.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, click **Yes** when the following confirmation appears:



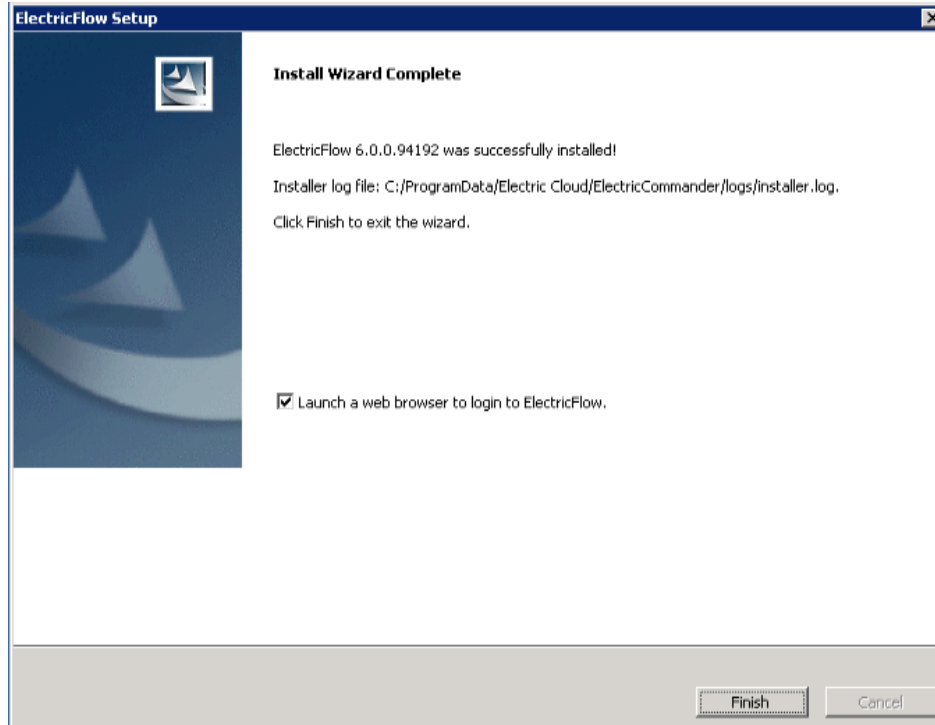
- **Group Name** - Use this field to enter the name of the group that owns the ElectricFlow agent process.

The Ready to Install screen appears.



10. Review your installation settings. Use the **Back** button to modify any information if necessary.
11. Click **Next** to continue.

The installer displays a status bar to show the progress of the installation process which can take fifteen minutes. When the install process is complete, the Install Wizard Complete screen appears.



Note: The ElectricFlow server will automatically start when installation is complete.

11. Select the **Launch a web browser to login to ElectricFlow** check box if you want the ElectricFlow login screen to open.
12. Click **Finish** to close the wizard.

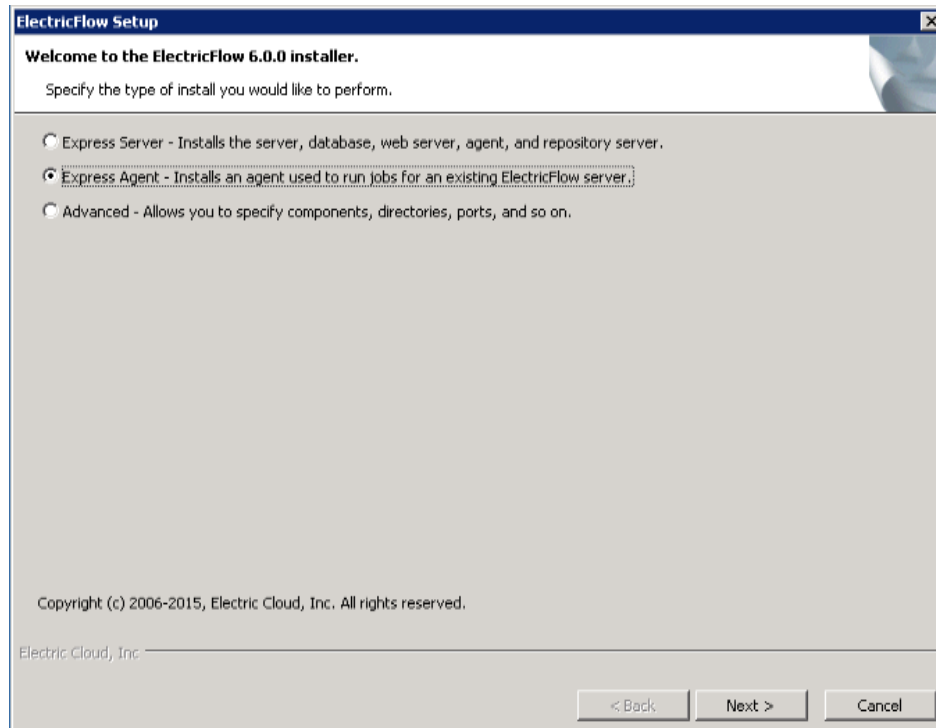
Running an Express Agent Graphical User Interface Installation

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

Note: You might install ElectricFlow agent software on Windows or Linux with this installation method. For Solaris, HP-UX, Mac, or other supported UNIX agent machines, see [Non-Server Platform Agent Installation Method](#) on page 13-24.

1. Double-click the `ElectricFlow-<version>` file.

The Welcome to the ElectricFlow Installer screen appears.



Note: The screen examples in this procedure are from a Windows system. Different options might appear on a Linux system.

2. Select the **Express Agent** installation option, and then click **Next** to continue.

The Remote ElectricFlow Server screen appears.

ElectricFlow Setup

Remote ElectricFlow server

Specify an existing ElectricFlow server and what you would like to retrieve from it.

Server Host Name:

ElectricFlow User Name:

Password:

☒ Discover the plugins directory

☒ Create a resource

☐ Trusted (restrict to one server)

Resource Name:

☐ Create a repository

Repository Name:

☒ Create in default zone

Agent Gateway URL:

Zone Name:

Electric Cloud, Inc.

< Back Next > Cancel

3. Complete the following information on the Remote ElectricFlow Server screen.

- **Server Host Name** - Use this field to enter the name of the ElectricFlow server that will communicate with this agent. If the remote server is using a non-default HTTP port, you must specify the Server Host Name as `host:port`.
- **ElectricFlow User Name** - Use this field to enter the name of an ElectricFlow user on the ElectricFlow server who has sufficient privileges to create a resource. This field defaults to the ElectricFlow-supplied `admin` user.
- **Password** - Use this field to enter the password for the ElectricFlow user. The default password for the `admin` user is `changeme`.
- **Discover the plugins directory** - Select this check box if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

Note: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

- **Create a resource** - Select this check box if you want to create a resource on the remote ElectricFlow server for the agent you are installing.
- **Trusted** - Select this check box to restrict this agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.
- **Resource Name** - Use this field to enter the name of the resource you would like to use for the agent. This field is available for use when the Create a resource check box is selected.
- **Create in default zone** - Select this check box if you want to create the agent in the default zone.

- **Agent Gateway URL** - Use this field to enter the URL of the gateway used to communicate with the ElectricFlow server. This field is available for use when the Create in default zone check box is cleared.
 - **Zone Name** - Use this field to enter the name of the zone used during remote agent and/or remote repository creation. This field is available for use when the Create in default zone check box is cleared.
4. Click **Next** to continue.

The Agent Service Account screen appears.

The screenshot shows a Windows-style dialog box titled "ElectricFlow Setup". Inside, the section "Agent service account" is active, with the instruction "Specify the account you would like the agent service to run as." Below this are three text input fields: "User Name" (containing "build"), "Password", and "Domain - Leave blank for local users". There is also an unchecked checkbox labeled "Use the local system account". At the bottom left is the text "Electric Cloud, Inc." and at the bottom right are three buttons: "< Back", "Next >", and "Cancel".

5. Select the appropriate steps for your platform and complete the following information on the screen.
- If you have a Windows system:
 - **User Name** - Use this field to enter the name of the user who will run the ElectricFlow agent service.

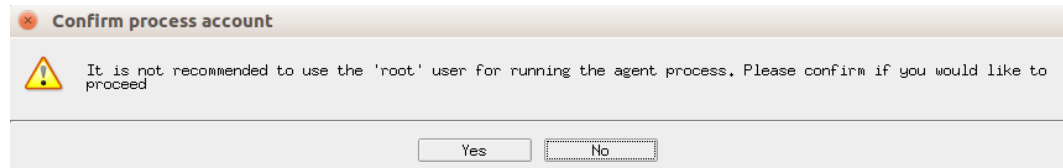
The user that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory.
 - **Password** - Use this field to enter the password of the user who will run the ElectricFlow agent service.
 - **Domain** - Use this field to enter the domain name information for the user. For example, `electric-cloud.com`. Leave this field blank if this is a local user.
 - **Use the local system account** - Select this check box if you want the ElectricFlow agent service to run as the local Windows system account.

Note: The local system account cannot access network resources such as shared filesystems used for plugins or workspaces.

- If you have a Linux system:

- **User Name** - Use this field to enter the name of the user who owns the ElectricFlow agent process.

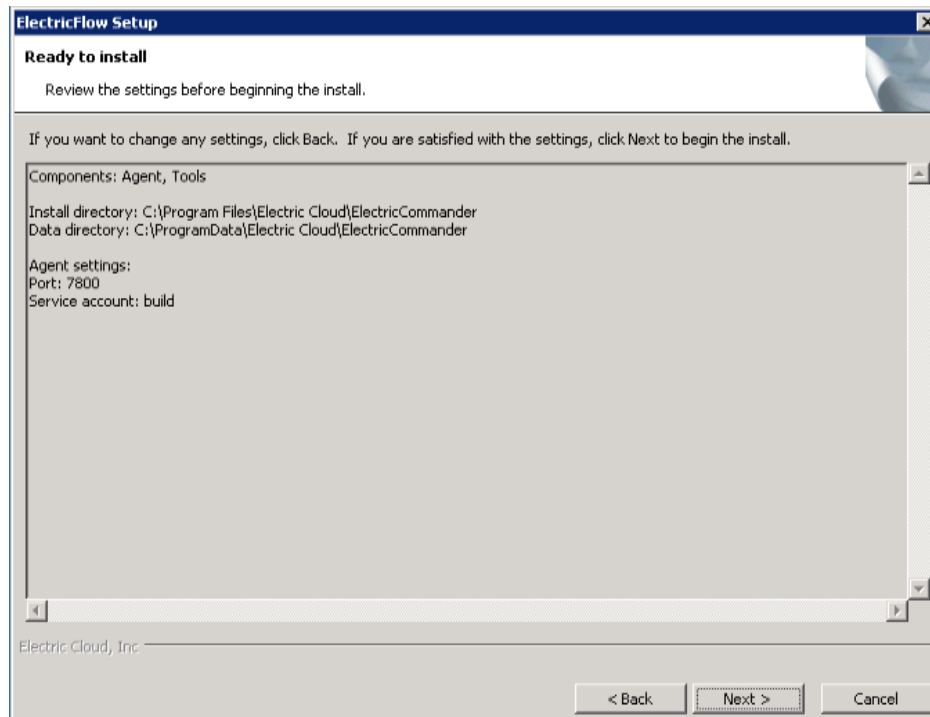
The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, click **Yes** when the following confirmation appears:



- **Group Name** - Use this field to enter the name of the group who owns the ElectricFlow agent process.

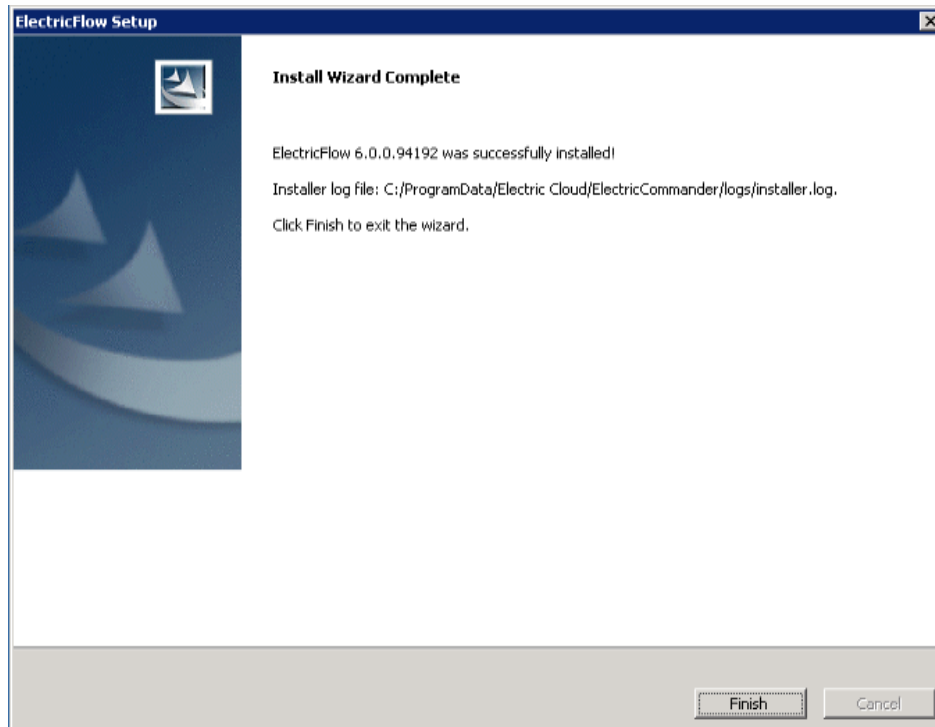
6. Click **Next** to continue.

The Ready to Install Screen appears.



7. Review this screen to verify your selections. Use the **Back** button to change any of your settings if necessary.
8. Click **Next** to continue.

ElectricFlow installs the agent and tools components. This process can take a few minutes.



9. Click **Finish** to complete the installation.

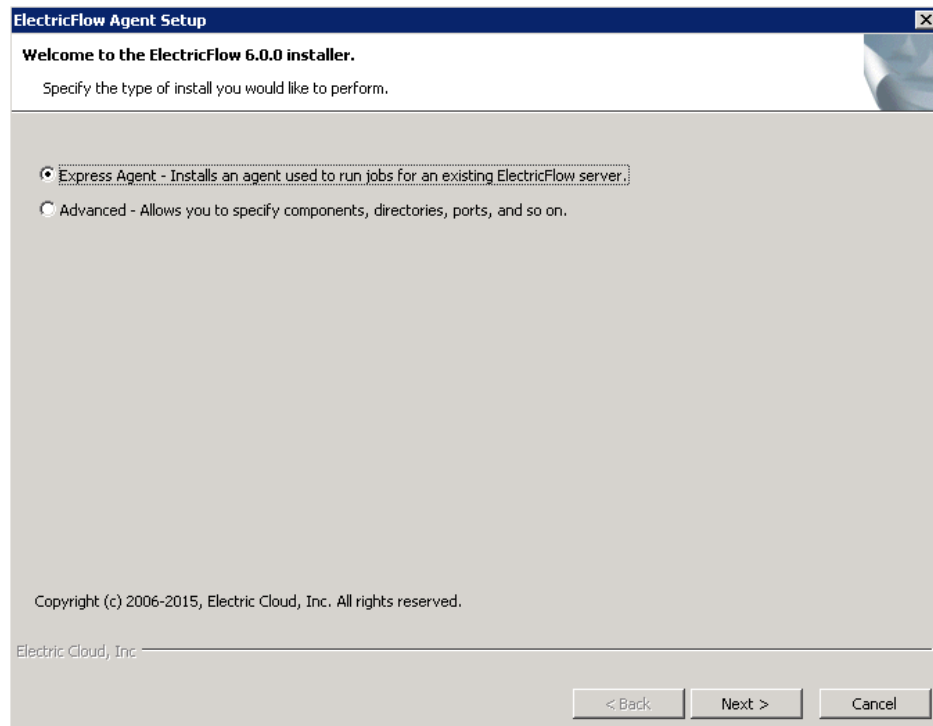
Running an Express Agent Graphical User Interface Installation (Agent-Only Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. You can install a 32-bit agent or a 64-bit agent. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

Note: You might install ElectricFlow agent software on Windows or Linux with this installation method. For Solaris, HP-UX, Mac, or other supported UNIX agent machines, see [Non-Server Platform Agent Installation Method](#) on page 13-24.

1. Double-click the `ElectricFlowAgent-x86-<version>` file (for a 32-bit agent) or the `ElectricFlowAgent-x64-<version>` file (for a 64-bit agent).

The Welcome to the ElectricFlow Installer screen appears.



Note: The screen examples in this procedure are from a Windows system. Different options might appear on a Linux system.

2. Select the **Express Agent** installation option, and then click **Next** to continue.

The Remote ElectricFlow Server screen appears.

ElectricFlow Setup

Remote ElectricFlow server

Specify an existing ElectricFlow server and what you would like to retrieve from it.

Server Host Name

ElectricFlow User Name

Password

☒ Discover the plugins directory

☒ Create a resource

☐ Trusted (restrict to one server)

Resource Name

☐ Create a repository

Repository Name

☒ Create in default zone

Agent Gateway URL

Zone Name

Electric Cloud, Inc.

< Back Next > Cancel

3. Complete the following information on the Remote ElectricFlow Server screen.

- **Server Host Name** - Use this field to enter the name of the ElectricFlow server that will communicate with this agent. If the remote server is using a non-default HTTP port, you must specify the Server Host Name as `host:port`.
- **ElectricFlow User Name** - Use this field to enter the name of an ElectricFlow user on the ElectricFlow server who has sufficient privileges to create a resource. This field defaults to the ElectricFlow-supplied `admin` user.
- **Password** - Use this field to enter the password for the ElectricFlow user. The default password for the `admin` user is `changeme`.
- **Discover the plugins directory** - Select this check box if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

Note: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10

- **Create a resource** - Select this check box if you want to create a resource on the remote ElectricFlow server for the agent you are installing.
- **Trusted** - Select this check box to restrict this agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.
- **Resource Name** - Use this field to enter the name of the resource you would like to use for the agent. This field is available for use when the Create a resource check box is selected.
- **Create in default zone** - Select this check box if you want to create the agent in the default zone.

- **Agent Gateway URL** - Use this field to enter the URL of the gateway used to communicate with the ElectricFlow server. This field is available for use when the Create in default zone check box is cleared.
 - **Zone Name** - Use this field to enter the name of the zone used during remote agent and/or remote repository creation. This field is available for use when the Create in default zone check box is cleared.
4. Click **Next** to continue.

The Agent Service Account screen appears.

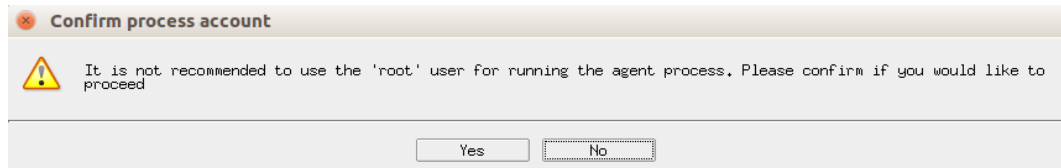
5. Select the appropriate steps for your platform and complete the following information on the screen.
- If you have a Windows system:
 - **User Name** - Use this field to enter the name of the user who will run the ElectricFlow agent service.
The user that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory.
 - **Password** - Use this field to enter the password of the user who will run the ElectricFlow agent service.
 - **Domain** - Use this field to enter the domain name information for the user. For example, `electric-cloud.com`. Leave this field blank if this is a local user.
 - **Use the local system account** - Select this check box if you want the ElectricFlow agent service to run as the local Windows system account.

Note: The local system account cannot access network resources such as shared filesystems used for plugins or workspaces.

- If you have a Linux system:

- **User Name** - Use this field to enter the name of the user who owns the ElectricFlow agent process.

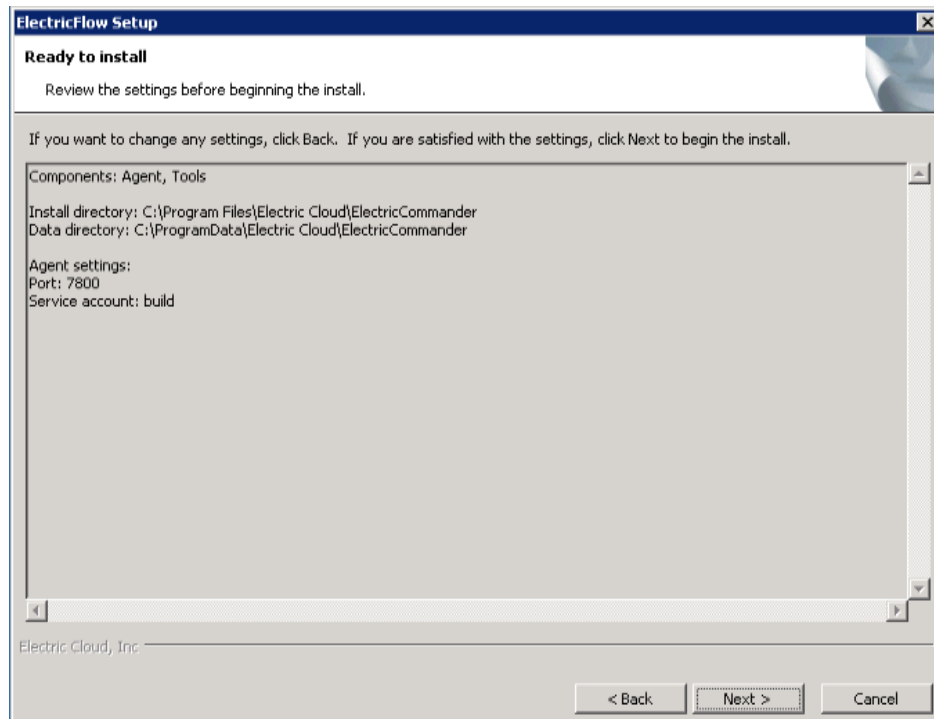
The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, click **Yes** when the following confirmation appears:



- **Group Name** - Use this field to enter the name of the group who owns the ElectricFlow agent process.

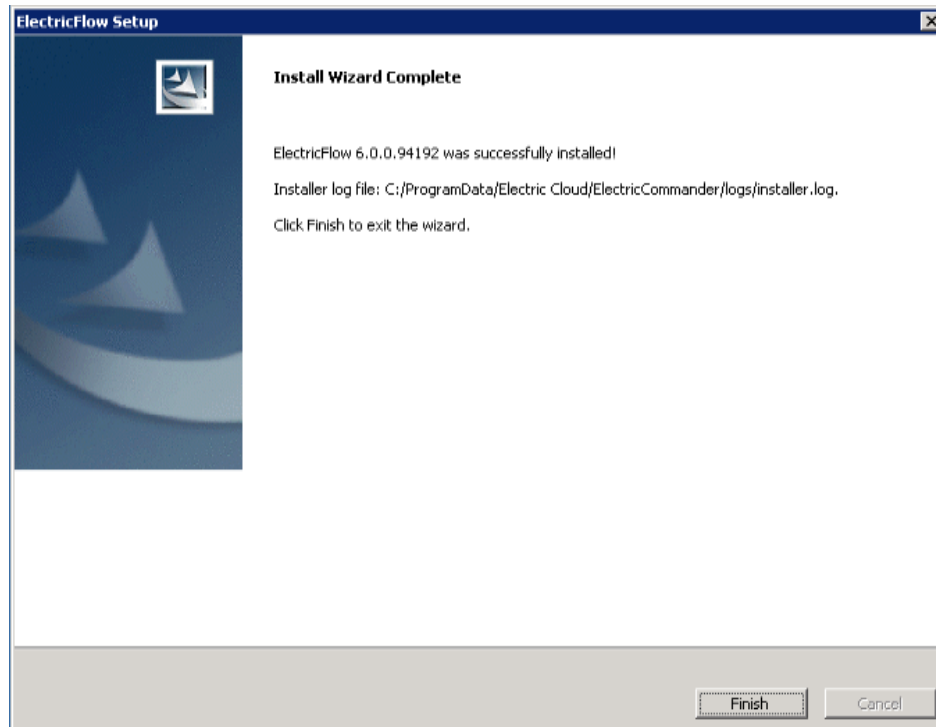
6. Click **Next** to continue.

The Ready to Install Screen appears.



7. Review this screen to verify your selections. Use the **Back** button to change any of your settings if necessary.
8. Click **Next** to continue.

ElectricFlow installs the agent and tools components. This process can take a few minutes.



9. Click **Finish** to complete the installation.

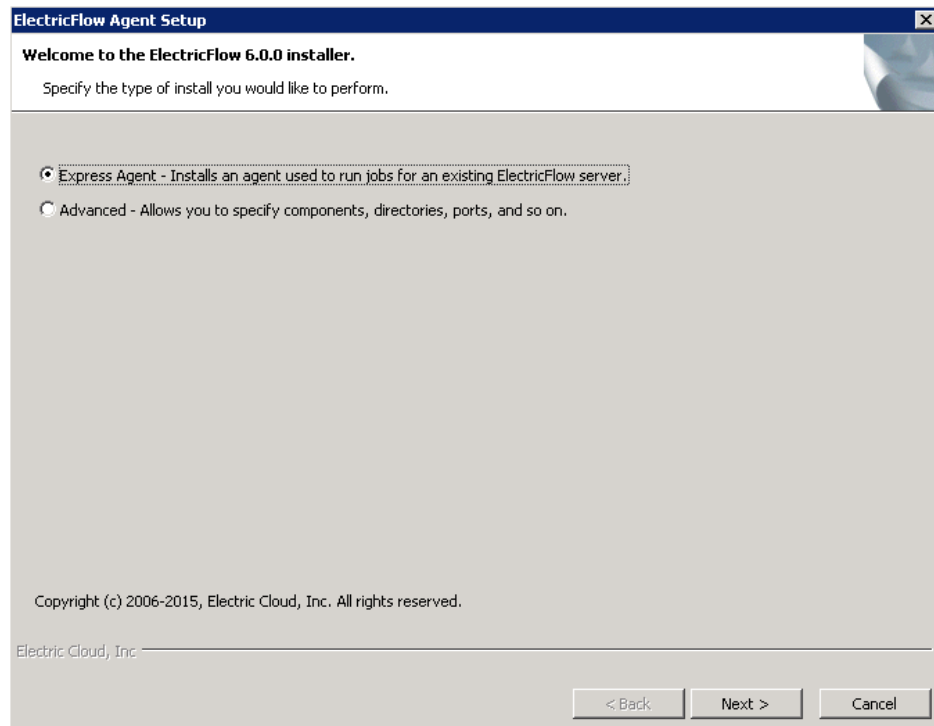
Running an Advanced Agent Graphical User Interface Installation (Agent-Only Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. You can install a 32-bit agent or a 64-bit agent. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

Note: You might install ElectricFlow agent software on Windows or Linux with this installation method. For Solaris, HP-UX, Mac, or other supported UNIX agent machines, see [Non-Server Platform Agent Installation Method](#) on page 13-24.

1. Double-click the `ElectricFlowAgent-x86-<version>` file (for a 32-bit agent) or the `ElectricFlowAgent-x64-<version>` file (for a 64-bit agent).

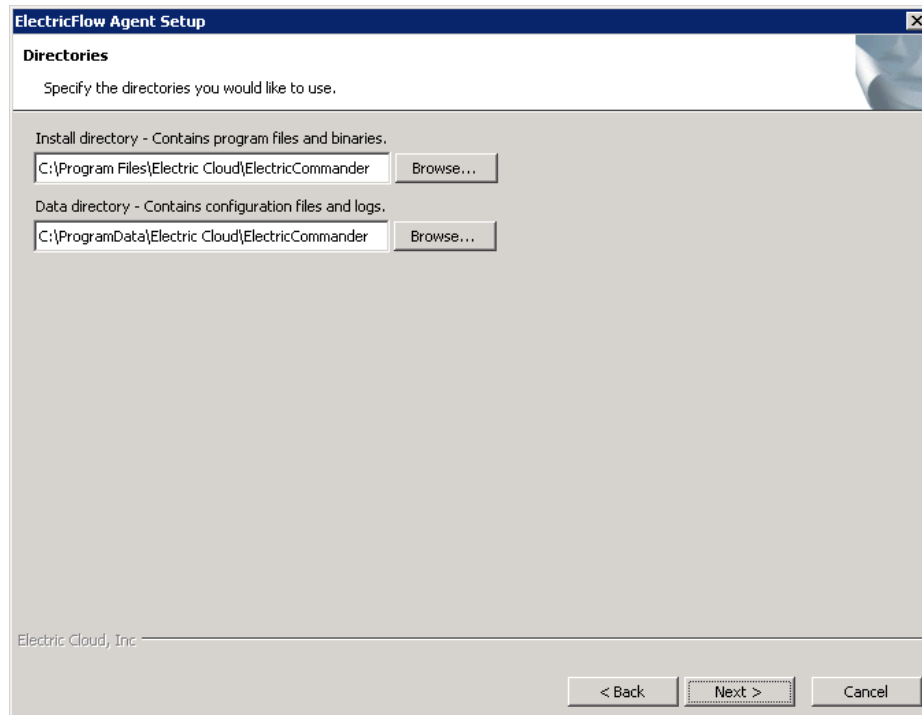
The Welcome to the ElectricFlow Installer screen appears.



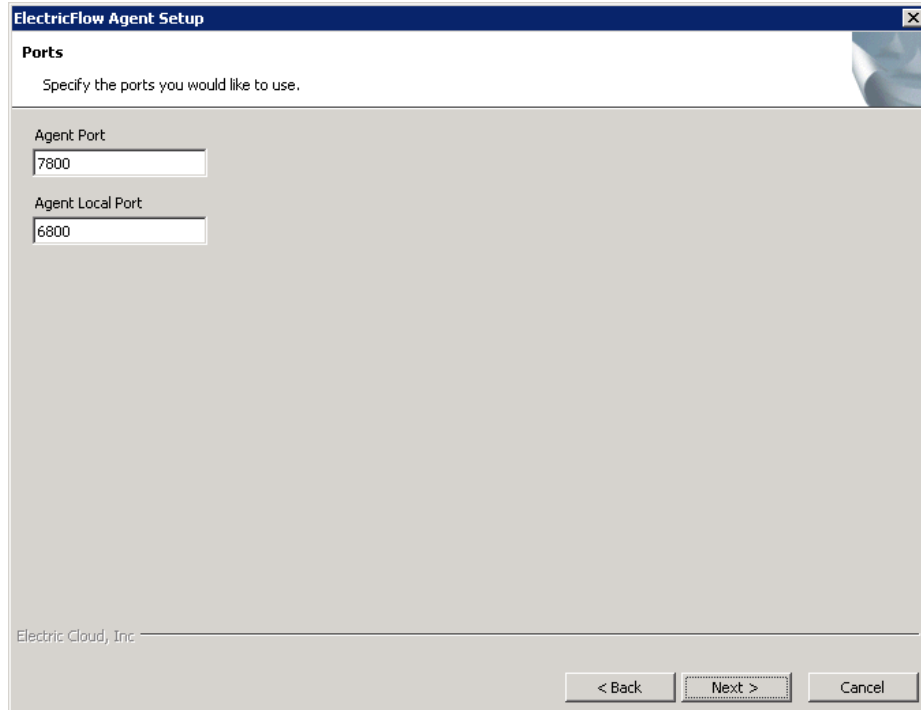
Note: The screen examples in this procedure are from a Windows system. Different options might appear on a Linux system.

2. Select the **Advanced Agent** installation option, and then click **Next** to continue.

The Directories screen appears.



3. Complete the following information on the Directories screen.
 - **Install directory** - Use this field to enter a new installation directory path for program files and binaries.
 - **Data directory** - Use this field to enter a new installation directory path for configuration files and logs.
 4. Click **Next** to continue.
- The Ports screen appears.

The screenshot shows a window titled "ElectricFlow Agent Setup" with a "Ports" tab. The instruction "Specify the ports you would like to use." is at the top. Below it are two input fields: "Agent Port" with the value "7800" and "Agent Local Port" with the value "6800". At the bottom left is the text "Electric Cloud, Inc.". At the bottom right are three buttons: "< Back", "Next >" (which is highlighted with a dashed border), and "Cancel".

ElectricFlow Agent Setup

Ports

Specify the ports you would like to use.

Agent Port
7800

Agent Local Port
6800

Electric Cloud, Inc.

< Back Next > Cancel

5. Complete the following information on the Ports screen.
 - **Agent port** - Use this field to specify a different port to eliminate any conflicts with your existing system configuration.
 - **Agent local port** - Use this field to specify a different port to be used by the agent for HTTP communication on the localhost network interface.
6. Click **Next** to continue.

The Remote ElectricFlow Server screen appears.

ElectricFlow Setup

Remote ElectricFlow server

Specify an existing ElectricFlow server and what you would like to retrieve from it.

Server Host Name:

ElectricFlow User Name:

Password:

☒ Discover the plugins directory

☒ Create a resource

☐ Trusted (restrict to one server)

Resource Name:

☐ Create a repository

Repository Name:

☒ Create in default zone

Agent Gateway URL:

Zone Name:

Electric Cloud, Inc.

< Back Next > Cancel

7. Complete the following information on the Remote ElectricFlow Server screen.

- **Server Host Name** - Use this field to enter the name of the ElectricFlow server that will communicate with this agent. If the remote server is using a non-default HTTP port, you must specify the Server Host Name as `host:port`.
- **ElectricFlow User Name** - Use this field to enter the name of an ElectricFlow user on the ElectricFlow server who has sufficient privileges to create a resource. This field defaults to the ElectricFlow-supplied `admin` user.
- **Password** - Use this field to enter the password for the ElectricFlow user. The default password for the `admin` user is `changeme`.
- **Discover the plugins directory** - Select this check box if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

Note: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10

- **Create a resource** - Select this check box if you want to create a resource on the remote ElectricFlow server for the agent you are installing.
- **Trusted** - Select this check box to restrict this agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.
- **Resource Name** - Use this field to enter the name of the resource you would like to use for the agent. This field is available for use when the Create a resource check box is selected.
- **Create in default zone** - Select this check box if you want to create the agent in the default zone.

- **Agent Gateway URL** - Use this field to enter the URL of the gateway used to communicate with the ElectricFlow server. This field is available for use when the Create in default zone check box is cleared.
 - **Zone Name** - Use this field to enter the name of the zone used during remote agent and/or remote repository creation. This field is available for use when the Create in default zone check box is cleared.
8. Click **Next** to continue.

The Agent Service Account screen appears.

The screenshot shows a Windows-style dialog box titled "ElectricFlow Setup". Inside, the section "Agent service account" is active, with the instruction "Specify the account you would like the agent service to run as." Below this are three text input fields: "User Name" (containing "build"), "Password", and "Domain - Leave blank for local users". There is also an unchecked checkbox labeled "Use the local system account". At the bottom left, it says "Electric Cloud, Inc.". At the bottom right are three buttons: "< Back", "Next >", and "Cancel".

9. Select the appropriate steps for your platform and complete the following information on the screen.
- If you have a Windows system:
 - **User Name** - Use this field to enter the name of the user who will run the ElectricFlow agent service.

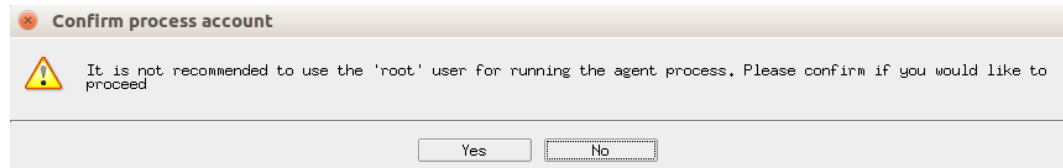
The user that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory.
 - **Password** - Use this field to enter the password of the user who will run the ElectricFlow agent service.
 - **Domain** - Use this field to enter the domain name information for the user. For example, `electric-cloud.com`. Leave this field blank if this is a local user.
 - **Use the local system account** - Select this check box if you want the ElectricFlow agent service to run as the local Windows system account.

Note: The local system account cannot access network resources such as shared filesystems used for plugins or workspaces.

- If you have a Linux system:

- **User Name** - Use this field to enter the name of the user who owns the ElectricFlow agent process.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, click **Yes** when the following confirmation appears:

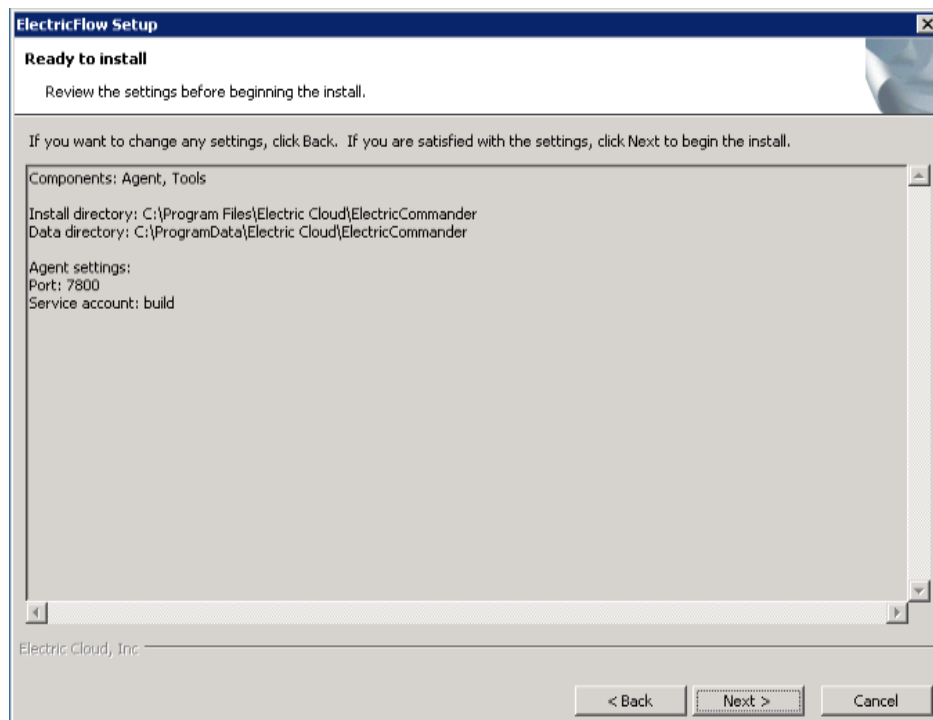


- **Group Name** - Use this field to enter the name of the group who owns the ElectricFlow agent process.

10. Select the appropriate steps for your platform and complete the following information on the screen.

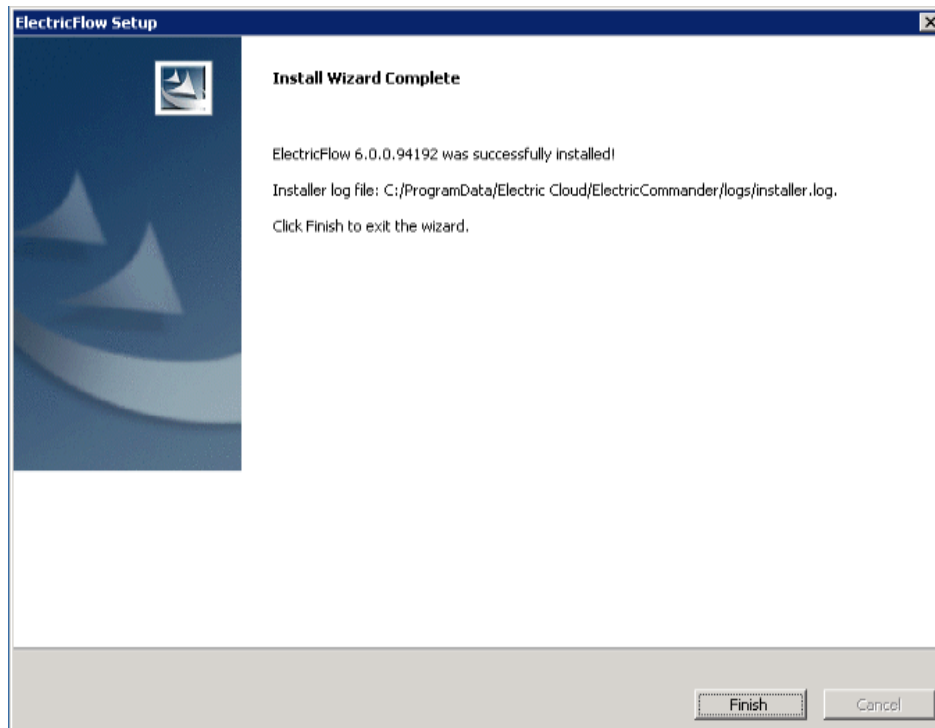
11. Click **Next** to continue.

The Ready to Install Screen appears.



12. Review this screen to verify your selections. Use the **Back** button to change any of your settings if necessary.
13. Click **Next** to continue.

ElectricFlow installs the agent and tools components. This process can take a few minutes.



14. Click **Finish** to complete the installation.

Interactive Command-Line Installation Methods

The interactive command-line installation methods are only supported for Linux-only installs on a local Linux volume. Electric Cloud does not support installing the ElectricFlow server on a network volume.

Note: You might install ElectricFlow agent software on Linux with this installation method. For Solaris, HP-UX, Mac, or other supported UNIX agent machines, see [Non-Server Platform Agent Installation Method](#) on page 13-24.

Running an Express Server Command-Line Installation

The express server command-line installation installs the ElectricFlow server, web server, built-in database, agent for running jobs, and ElectricFlow tools. This installation uses the default ElectricFlow server settings. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

Note: The built-in database is not supported in a clustered ElectricFlow configuration.

1. Choose one of the following commands to start the full installer:

- If you have a Linux platform, enter: `./ElectricFlow-<version>`
- If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface. To override this behavior, enter: `./ElectricFlow-<version> --mode console`

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow on your computer. Continue? [n/Y]
```

2. Continue the installation by entering `y`.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressServer,
expressAgent, or advanced. [expressServer]
```

3. Enter: `expressServer`.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Specify the user the server, web, and/or repository will run as. []
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.
Install them? [n/Y]
```

4. Enter `y` to continue the installation.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Specify the user the server, web, and/or repository will run as. []
```

5. Enter a user name.

This is the user who owns the ElectricFlow server, repository server, and web server processes. For example, you might enter `build`.

The software displays the following message.

```
Specify the group the server, web, and/or repository will run as. []
```

6. Enter a group name.

This is the group who owns the ElectricFlow server, repository server, and web server processes. For example, you might enter `build`.

The software displays the following message.

```
Use the same service account for the agent (not recommended for production
systems)? [y/N]
```

Note: For security reasons in production environments, you might want to use a separate user and group for the agent service because the server service has permission to read the key file

(`/opt/electriccloud/electriccommander/conf/passkey` in Linux or `C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey` in Windows). The key file is used to decrypt passwords stored in ElectricFlow. Using a different user and group for the agent service ensures that a process running on the agent cannot gain access to the key file.

7. Choose one of the following options:

- Enter `y` to use the same user and group for the agent service. This is not recommended for production systems.
- Enter `n` to use a separate user and group for the agent service.

The software displays the following message.

Specify the user the agent will run as. []

a. Enter a User Name. This is the user who owns the ElectricFlow agent process.

For example, you might enter `build`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

It is not recommended to use the 'root' user for running the agent process. Please confirm if you would like to proceed [y/N]

The software displays the following message.

Specify the group the agent will run as. []

b. Enter a Group Name. This is the group that owns the ElectricFlow agent process.

For example, you might enter `build`.

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line "ElectricFlow <version> was successfully installed!" appears.

Running an Advanced Command-Line Installation

The advanced command-line installation allows you to install individual ElectricFlow components such as an ElectricFlow server, web server, repository server, or ElectricFlow tools on specific machines. You can also change the default installation settings to accommodate your environment. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Run the following command to start the installer.

- If you have a Linux platform, enter: `./ElectricFlow-<version>`
- If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface. To override this behavior, enter: `./ElectricFlow-<version> --mode console`

The software displays the following message.

Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.

This will install ElectricFlow on your computer. Continue? [n/Y]

2. Enter `y` to continue the installation.

The software displays the following message.

Specify the type of setup you would like to perform: `expressServer`, `expressAgent`, or `advanced`. [`expressServer`]

3. Enter: `advanced`.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Install an ElectricFlow server? [n/Y]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.
Install them? [n/Y]
```

4. Enter `y` to continue the installation.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Install an ElectricFlow server? [n/Y]
```

5. Select the servers you want to install on the current machine. For more information, see [Architecture](#) on page 1-4.

Note: If you only want to install the ElectricFlow tools, enter `n` for all of the server options. The ElectricFlow tools are automatically installed even if you choose not to install any server software.

a. Enter `y` to install an ElectricFlow server.

The software displays the following message.

```
Install a built-in database? [n/Y]
```

b. Choose one of the following options:

- Enter `y` to install a built-in database.

This is not recommended for production systems. Additionally, the built-in database is **not** supported in a clustered ElectricFlow configuration.

- Enter `n` if you plan to use an external database.

For more information, see [External Database Configuration](#) on page 5-2. If you plan to use MySQL, see [Installing the MySQL JDBC Driver](#) on page 3-103.

The software displays the following message.

```
Install an Apache web server? [n/Y]
```

c. Enter `y` to install an Apache web server.

The software displays the following message.

```
Install an ElectricFlow repository server? [n/Y]
```

d. Enter `y` to install an ElectricFlow repository server.

The software displays the following message.

```
Specify the install directory (for program files and binaries).  
[/opt/electriccloud/electriccommander]
```

IMPORTANT: The following installation options will vary depending on which servers you choose to install. You might see all, some, or none of the prompts in the following steps.

5. Press **Enter** to accept the default installation directory, or enter a new installation directory path for program files and binaries.

The software displays the following message.

```
Specify the data directory (for configuration files and logs).  
[/opt/electriccloud/electriccommander]
```

6. Press **Enter** to accept the default installation directory, or enter a new installation directory path for configuration files and logs.

The software displays prompts for server port values. The prompts that appear will vary depending on the sever software you previously selected to install.

7. Press **Enter** to accept the default port values, or enter alternate port numbers if you need to specify a different port value.

The following message only appears if you are installing an Apache web server. If you are not installing a web server, you will see a prompt to enter a user name.

```
Specify the host name that users will type in their browser to access the web  
server. [hostName]
```

8. Enter a web server host name if you are installing an Apache web server.
9. This is the host name users need to type into their browser to access the ElectricFlow web server.

The software displays the following message.

```
Specify the user the server, web, and/or repository will run as. []
```

10. Enter a user name if you are installing an ElectricFlow, web, or repository server.

This is the user who owns the ElectricFlow server, repository server, and web server processes. For example, you might enter `build`.

The software displays the following message.

```
Specify the group the server, web, and/or web repository will run as. []
```

11. Enter a group name if you are installing an ElectricFlow, web, or repository server.

This is the group who owns the ElectricFlow server, repository server, and web server processes. For example, you might enter `build`.

The following message only appears if an agent is installed on this machine. An agent is automatically installed with a web or repository server.

```
Use the same service account for the agent (not recommended for production  
systems)? [y/N]
```

Note: For security reasons in production environments, you might want to use a separate user and group for the agent service because the server service has permission to read the key file (`/opt/electriccloud/electriccommander/conf/passkey` in Linux or `C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey` in Windows). The key file

is used to decrypt passwords stored in ElectricFlow. Using a different user and group for the agent service ensures that a process running on the agent cannot gain access to the key file.

12. Choose one of the following options if an agent is automatically installed with the server:
 - Enter `y` to use the same user and group for the agent service. This is not recommended for production systems.
 - Enter `n` to use a separate user and group for the agent service.

The software displays the following message.

```
Specify the user the agent will run as. []
```

- a. Enter a User Name. This is the user who owns the ElectricFlow agent process.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent
process. Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

- b. Enter a Group Name. This is the group that owns the ElectricFlow agent process..

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line "ElectricFlow <version> was successfully installed!" appears.

Running an Express Agent Command-Line Installation

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:
 - If you have a Linux platform, enter: `./ElectricFlowAgent-<version>`
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface. To override this behavior, enter: `./ElectricFlow--<version> --mode console`

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow on your computer. Continue? [n/Y]
```

2. Continue the installation by entering `y`.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressServer,
expressAgent, or advanced. [expressServer]
```

3. Enter: `expressAgent`.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686  
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.  
Install them? [n/Y]
```

4. Enter `y`.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

5. Enter `y` if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

```
Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]
```

6. Enter `y` to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for **pre-existing** ElectricFlow servers.

The software displays the following message.

```
Register as trusted agent (required for gateway)? [y/N]
```

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

7. Choose one of the following options:

- If a gateway used to communicate with the ElectricFlow server, you must select `y`. This option allows you to create a trusted network connection between the agent and server under the same certificate authority. This will allow the agent and the ElectricFlow server to communicate across the network.
- If there is no gateway between the agent and ElectricFlow server, enter `n`.

IMPORTANT: If you deviated from the recommended agent options, you will see variations in the installation options that appear on your system.

The software displays the following message.

```
Create repository and/or agent in the default zone? [n/Y]
```

7. Enter `y` to create the agent in the default zone.

The software displays the following message.

```
Specify the host:port of a remote ElectricFlow server the agent, repository
server and/or web server being installed can link to. The port is only required
if it is not the default. [] <hostName:port>
```

8. Enter the Server Host Name of the ElectricFlow server that will communicate with this agent. You must specify the Server Host Name as `hostName:port` if the remote server is using a non-default HTTP port.

The software displays the following message.

```
Specify the user name with which to login to <hostName:port>. [admin]
```

9. Enter the user name of a user on the ElectricFlow server who has sufficient privileges to create a resource. The default is the ElectricFlow-supplied `admin` user.

The software displays the following message.

```
Specify the password for "<electricflow_user>" on <hostName:port>. []
```

10. Enter the password for the ElectricFlow user. The default password for the `admin` user is `changeme`.

The software displays the following message.

```
Specify the name of the resource to create on <hostName:port>. [<resource_name>]
```

11. Enter the following information if the agent must be registered as a trusted agent. These options only appear if you entered `y` for Register as trusted agent (required for gateway)? [`y/N`].

a. Enter a resource name to use on the ElectricFlow server.

The software displays the following message.

```
Specify the agent gateway URL in the form of 'ipOrHostname:port' []
```

b. Enter an agent gateway URL. This is the URL of the gateway used to communicate with the ElectricFlow server.

The software displays the following message.

```
Specify the zone name for the agent and/or repository []
```

c. Enter the Zone Name. This is the zone used during remote agent and/or remote repository creation.

The software displays the following message.

```
Specify the user the agent will run as. []
```


12. Enter a user name. This is the user who owns the ElectricFlow agent process. For example, you might enter `build`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.  
Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

13. Enter a Group Name. This is the group that owns the ElectricFlow agent process. For example, you might enter `build`.

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line "ElectricFlow <version> was successfully installed!" appears.

Running an Express Agent Command-Line Installation (Agent-Only Installer)

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:
 - If you have a Linux platform, enter `./ElectricFlowAgent-x86-<version>` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version>` (for a 64-bit agent).
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface.

To override this behavior, enter `./ElectricFlow-x86-<version> --mode console` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version> --mode console` (for a 64-bit agent).

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow Agent on your computer. Continue? [n/Y]
```

2. Continue the installation by entering `y`.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressAgent or advanced.  
[expressAgent]
```

3. Enter: `expressAgent`.
 - If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.
Install them? [n/Y]
```

4. Enter `y`.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

5. Enter `y` if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

```
Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]
```

6. Enter `y` to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for **pre-existing** ElectricFlow servers.

The software displays the following message.

```
Register as trusted agent? [y/N]
```

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

7. Choose one of the following options:

- If a gateway is used to communicate with the ElectricFlow server, you must select `y`. This option allows you to create a trusted network connection between the agent and server under the same certificate authority. This will allow the agent and the ElectricFlow server to communicate across the

network.

- If there is no gateway between the agent and ElectricFlow server, enter `n`.

IMPORTANT: If you deviated from the recommended agent options, you will see variations in the installation options that appear on your system.

The software displays the following message.

```
Specify the user the agent will run as. []
```

7. Enter a user name. This is the user who owns the ElectricFlow agent process. For example, you might enter `build`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.  
Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

8. Enter a Group Name. This is the group that owns the ElectricFlow agent process. For example, you might enter `build`.

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line "ElectricFlow <version> was successfully installed!" appears.

Running an Advanced Agent Command-Line Installation (Agent-Only Installer)

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:
 - If you have a Linux platform, enter `./ElectricFlowAgent-x86-<version>` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version>` (for a 64-bit agent).
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface.

To override this behavior, enter `./ElectricFlow-x86-<version> --mode console` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version> --mode console` (for a 64-bit agent).

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow on your computer. Continue? [n/Y]
```

2. Continue the installation by entering `y`.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressAgent or advanced.  
[expressAgent]
```

3. Enter: advanced.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Specify the install directory (for program files and binaries).
[/opt/electriccloud/electriccommander]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.
Install them? [n/Y]
```

4. Enter y.

The missing 32-bit libraries are installed.

Note: If you enter n, the installation process stops and the installer exits.

The software displays the following message.

```
Specify the install directory (for program files and binaries).
[/opt/electriccloud/electriccommander]
```

5. Enter a new installation directory path for program files and binaries.

The software displays the following message.

```
Specify the data directory (for configuration files and logs).
[/opt/electriccloud/electriccommander]
```

6. Enter a new installation directory path for configuration files and logs.

The software displays the following message.

```
Specify the agent port. [7800]
```

7. Enter a different port to eliminate any conflicts with your existing system configuration.

The software displays the following message.

```
Specify the agent local port. [6800]
```

8. Enter a different port to be used by the agent for HTTP communication on the localhost network interface.

The software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

9. Enter y if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

```
Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]
```

10. Enter `y` to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for **pre-existing** ElectricFlow servers.

The software displays the following message.

```
Register as trusted agent? [y/N]
```

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development environments.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

11. Choose one of the following options:

- If a gateway is used to communicate with the ElectricFlow server, you must select `y`. This option allows you to create a trusted network connection between the agent and server under the same certificate authority. This will allow the agent and the ElectricFlow server to communicate across the network.
- If there is no gateway between the agent and ElectricFlow server, enter `n`.

IMPORTANT: If you deviated from the recommended agent options, you will see variations in the installation options that appear on your system.

The software displays the following message.

```
Specify the user the agent will run as. []
```

11. Enter a user name. This is the user who owns the ElectricFlow agent process. For example, you might enter `build`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.  
Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

12. Enter a Group Name. This is the group that owns the ElectricFlow agent process. For example, you might enter `build`.

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line "ElectricFlow <version> was successfully installed!" appears.

Running an Express Agent Command-Line Installation (Agent-Only Installer) When the Server Uses Registered and Concurrent Licenses

Use this procedure when the ElectricFlow server uses a mix of registered and concurrent licenses.

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:
 - If you have a Linux platform, enter `./ElectricFlowAgent-x86-<version>` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version>` (for a 64-bit agent).
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface. To override this behavior, enter `./ElectricFlowAgent-x86-<version> --mode console` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version> --mode console` (for a 64-bit agent).

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow Agent on your computer. Continue? [n/Y]
```

2. Enter `y` to continue the installation.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressAgent or advanced.
[expressAgent]
```

3. Enter: `expressAgent`.
 - If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686
nss-pam-ldapd.*i686. They are required for correct work of ElectricFlow.
Install them? [n/Y]
```

4. Enter `y`.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

5. Enter `y` if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

```
Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]
```

6. Enter `y` to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for *pre-existing* ElectricFlow servers.

The software displays the following message.

```
Register as trusted agent? [y/N]
```

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development environments.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

7. Enter `n` if you are installing the ElectricFlow Community Edition.

The software displays the following message.

```
Create repository and/or agent in the default zone? [y/n]
```

8. Enter `y`.

The software displays the following message.

```
Specify the host:port of a remote ElectricFlow server that the agent,  
repository server and/or web server being installed can link to. The  
port is only required if it is not the default. []
```

9. Enter the `<host:port>`.

The software displays the following message.

```
Specify the user name with which to login to "<host:port>". [admin]
```

10. Enter `admin`.

The software displays the following message.

```
Specify the password for "admin" on "<host:port>". []
```

11. Enter a password.

The software displays the following message.

```
Specify the name of the resource to create on "<host:port>". []
```

12. Enter a resource name.

The software displays the following message.

```
Specify resource type for remote server: Registered or Concurrent. []
```

13. Enter Registered.

The software displays the following message.

```
Specify the user the agent will run as. []
```

14. Enter a user name.

This is the user who owns the ElectricFlow agent process. For example, you can enter `deploy`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.
Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

15. Enter a group name.

This is the group that owns the ElectricFlow agent process. For example, you can enter `deploy`.

The ElectricFlow software is installed on the machine.

When the installation completes successfully, a message that contains the line "ElectricFlow `<version>` was successfully installed!" appears.

Silent Unattended Installation Method

You can run the ElectricFlow installers in unattended (silent) mode with no user interface for either Windows or Linux. Use the arguments in the following list to construct the commands you need for the particular installation you need to perform. For example: server, agent, web server, and so on.

Running a Silent Install

- Choose one of the following commands to begin a silent install:
 - If you have a Linux system, go to a command line and enter one of the following commands:


```
Full installer: ./ElectricFlow-<version> --mode silent <--arguments>
```

```
32-bit agent-only installer: ./ElectricFlowAgent-x86-<version> --mode silent <--arguments>
```

```
64-bit agent-only installer: ./ElectricFlowAgent-x64-<version> --mode silent <--arguments>
```


- If you have a Windows system, go to a command line and enter one of the following commands:

Full installer: `ElectricFlow-<version>.exe --mode silent <--arguments>`

32-bit agent-only installer: `ElectricFlowAgent-x86-<version>.exe --mode silent <--arguments>`

64-bit agent-only installer: `ElectricFlowAgent-x64-<version>.exe --mode silent <--arguments>`

Where:

- `<version>` is your ElectricFlow version number
- `<arguments>` represents any additional silent install arguments you add to the command. For more information, see [Silent Install Arguments](#) on page 3-54.

Silent Install Arguments

The following argument table is an excerpt from the installer help text. You can view the full installer help by entering the `ElectricFlow-<version> --help` command, the `ElectricFlowAgent-x86-<version>` command, or the `ElectricFlowAgent-x64-<version>` command.

IMPORTANT: Only limited validity checking is performed on these values during an unattended installation, which means typing errors or other mistakes might manifest themselves in strange ways.

Note:

Although documented in the ElectricFlow Installation Guide for versions 6.4 and earlier and for versions 6.0.4 and earlier, the `response-file` and `save-response-file` arguments are not supported. Although documented in the ElectricCommander 5.x Installation Guide, the `response-file` and `save-response-file` arguments are not supported.

<code>--agentAllowRootUser</code>	(Linux platforms) Lets you specify <code>root</code> as the user to own the ElectricFlow agent process when you use the <code>--unixAgentUser</code> argument. If you do not use this argument and specify <code>root</code> as the agent user, the installation will exit, and the installer log will contain the following error message: It is not recommended to use the 'root' user for running the agent process. Please use the <code>--agentAllowRootUser</code> flag to override this error and proceed with the 'root' user.
<code>--agentArtifactCache [<directory>]</code>	Directory containing cached artifact versions.
<code>--agentGatewayURL [<URL>]</code>	URL of the gateway used to communicate with the ElectricFlow server.
<code>--agentIdleConnectionTimeout [<milliseconds>]</code>	Idle connection timeout for the ElectricFlow agent (in milliseconds). The default is 0 (the connection does not time out).
<code>--agentInitMemory [<percent>]</code>	Initial Java heap size as a percentage of the total system memory for the ElectricFlow agent. This setting has no default and is overridden if you have set <code>agentInitMemoryMB</code> to a nondefault value.

<code>--agentInitMemoryMB [<megabytes>]</code>	Initial Java heap size for the ElectricFlow agent (in MB). The default is 16.
<code>--agentMaxConnections [<number>]</code>	Maximum number of network connections for the ElectricFlow agent. The default is 200.
<code>--agentMaxConnectionsPerRoute [<number>]</code>	Maximum number of network connections per route for the ElectricFlow agent. The default is 20.
<code>--agentMaxMemory [<percent>]</code>	Maximum Java heap size as a percentage of the total system memory. This setting has no default and is overridden if you have set <code>agentMaxMemoryMB</code> to a nondefault value.
<code>--agentMaxMemoryMB [<megabytes>]</code>	Maximum Java heap size for the ElectricFlow agent (in MB). The default is 64.
<code>--agentLocalPort [<port>]</code>	Port used by the ElectricFlow agent for HTTP communication on the localhost network interface.
<code>--agentOutboundConnectTimeout</code>	Timeout for the ElectricFlow agent establishing outbound connections (in milliseconds). The default is 30000.
<code>--agentPort [<port>]</code>	Port used by the installed ElectricFlow agent for HTTPS communication on any network interface.
<code>--agentWindowsServiceStartType [<start_type>]</code>	(Windows only) Start type of the ElectricFlow agent service. Available values are <code>auto_start</code> and <code>delayed_auto_start</code> .
<code>--dataDirectory [<directory>]</code>	Directory used to store configuration files, logs, and database artifacts.
<code>--force32Bit</code>	Force a 32-bit install, even if the machine is 64-bit.
<code>--haveRunMigrationUtility</code>	When you upgrade from ElectricCommander 4.2.x to ElectricFlow 5.0 and later, the database is modified to support modifications to universally unique identifiers (UUIDs). The upgrade process includes using a migration tool to add support for UUIDs. For more information, go to Properties Changed to UUIDs on page 7-5.
<code>--help</code>	Display this information.
<code>--installAgent</code>	Install the ElectricFlow agent. This option is not required when using the 32-bit or 64-bit agent-only installers. It is ignored when used.

<code>--installDatabase</code>	Install a local built-in database to use with the main ElectricFlow server. This database is not recommended for production systems.
<code>--installDirectory [<directory>]</code>	Directory used to store program files and binaries.
<code>--installRepository</code>	Install an ElectricFlow artifact repository server.
<code>--installServer</code>	Install the main ElectricFlow server.
<code>--installWeb</code>	Install the local web server and ElectricFlow web interface.
<code>--mode [<installer_mode>]</code>	Mode in which the installer will run. Available values: console, silent, or standard.
<code>--remoteServer [<host>:<port>]</code>	host:port for the remote ElectricFlow server. The port is optional and can be omitted if the server is using the default HTTP port.
<code>--remoteServerCreateRepository</code>	Create a repository object on the remote ElectricFlow server.
<code>--remoteServerCreateResource</code>	Create a resource for the installed agent on the remote ElectricFlow server.
<code>--remoteServerDiscoverPlugins</code>	Set the plugins directory for the installed agent and/or web server to the shared plugins directory defined on the remote ElectricFlow server.
<code>--remoteServerPassword [<password>]</code>	Password to use when logging in to the remote ElectricFlow server.
<code>--remoteServerRepository [<repository_name>]</code>	Name of the repository to create on the remote ElectricFlow server.
<code>--remoteServerResource [<resource_name>]</code>	Name of the resource to create on the remote ElectricFlow server.
<code>--remoteServerResourceType [<type>]</code>	Type of resource to create on the remote ElectricFlow server. This argument is available only when the ElectricFlow server is using a mixed-mode license (concurrent resources and registered hosts). Valid options for this argument are concurrent or registered.
<code>--remoteServerUser [<user_name>]</code>	User name to use when logging in to the remote ElectricFlow server.
<code>--repositoryPort [<port>]</code>	Port used by the ElectricFlow artifact repository server (the default is 8200).
<code>--serverFileTransferPort [<port>]</code>	File transfer port used by the installed ElectricFlow server.

<code>--serverHttpPort [<port>]</code>	HTTP port used by the installed ElectricFlow server.
<code>--serverHttpsPort [<port>]</code>	HTTPS port used by the installed ElectricFlow server.
<code>--temp [<directory>]</code>	Directory used to store temporary files used by the installer.
<code>--trustedAgent</code>	Restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.
<code>--unixAgentGroup [<group_name>]</code>	Group the installed ElectricFlow agent runs as.
<code>--unixAgentUser [<user_name>]</code>	User the installed ElectricFlow agent runs as. The user and group that the agent runs as must have permission to write to the <code>\$INSTALL_DIRECTORY/log</code> directory. If you specify root, you must also use the <code>--agentAllowRootUser</code> argument (described above).
<code>--unixServerGroup [<group_name>]</code>	Group the installed ElectricFlow, web, or repository server runs as.
<code>--unixServerUser [<user_name>]</code>	User the installed ElectricFlow, web, or repository server runs as.
<code>--useSameServiceAccount</code>	Use the same account for server and agent services. Not recommended for production systems.
<code>--version</code>	Display installer version information.
<code>--webHostName [<name>]</code>	Name users need to type in their browser to access the web server.
<code>--webHttpPort [<port>]</code>	HTTP port used by the installed web server.
<code>--webHttpsPort [<port>]</code>	HTTPS port used by the installed web server.
<code>--windowsAgentDomain [<domain_name>]</code>	Domain of the account the installed ElectricFlow agent runs as.
<code>--windowsAgentLocalSystem</code>	Run the ElectricFlow agent as the local system account.
<code>--windowsAgentPassword [<password>]</code>	Password of the account the installed ElectricFlow agent runs as.

<code>--windowsAgentUser [<user_name>]</code>	User name of the account the installed ElectricFlow agent runs as. User that the agent runs as must have permission to write to the <code>\$INSTALL_DIRECTORY/log</code> directory.
<code>--windowsServerDomain [<domain_name>]</code>	Domain of the account the installed ElectricFlow, web, or repository server runs as.
<code>--windowsServerLocalSystem</code>	Run the ElectricFlow, web, or repository server as the local system account. Note: The local system account cannot access network resources such as shared file systems used for plugins or workspaces.
<code>--windowsServerPassword [<password>]</code>	Password of the account the installed ElectricFlow, web, or repository server runs as.
<code>--windowsServerUser [<user_name>]</code>	User name of the account the installed ElectricFlow, web, or repository server runs as.
<code>--windowsSkipAdminCheck</code>	Do not check that the user running the installer is a direct member of group Administrators.
<code>--zoneName [<zone_name>]</code>	Zone name used during remote agent and/or remote repository creation.

Linux Silent Installation Examples

The following examples are command strings to use for unattended (silent) installations. In many instances, the command text with the associated options wraps to the following lines.

IMPORTANT: You must enter the installation command and all options on a single line.

Complete ElectricFlow Installation

This installation installs the ElectricFlow server, including the web, repository, and database servers, one agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
./ElectricFlow-<version>
--mode silent
--installServer
--installAgent
--installDatabase
--installWeb
--installRepository
--unixServerUser <server user>
--unixServerGroup <server group>
--unixAgentUser <agent user>
--unixAgentGroup <agent group>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<server group>* is the group who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.
- *<agent group>* is the group that owns the ElectricFlow agent process.

Agent Installation

This installation example installs an ElectricFlow agent, a repository server, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
./ElectricFlow-<version>
--mode silent
--installRepository
--installAgent
--unixAgentUser <agent user>
--unixAgentGroup <agent group>
--unixServerUser <server user>
--unixServerGroup <server group>
--remoteServer <your existing ElectricFlow server>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<server group>* is the group who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.
- *<agent group>* is the group that owns the ElectricFlow agent process.

ElectricFlow Agent Installation (Full Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. This installation also installs Tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
./ElectricFlow-<version>
--mode silent --installAgent
--unixAgentUser <agent user>
--unixAgentGroup <agent group>
```

Where:

- *<agent user>* is the user who owns the ElectricFlow agent process.
- *<agent group>* is the group that owns the ElectricFlow agent process.

ElectricFlow Agent Installation (Agent-Only Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure. This example uses the 64-bit agent-only installer file.

- Enter:

```
./ElectricFlowAgent-x64-<version>
--mode silent
--unixAgentUser <agent user>
--unixAgentGroup <agent group>
```

Where:

- *<agent user>* is the user who owns the ElectricFlow agent process.
- *<agent group>* is the group that owns the ElectricFlow agent process.

Web Server and an Agent

This installation example installs an ElectricFlow web server, an agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
./ElectricFlow-<version>
--mode silent --installWeb
--installAgent
--unixAgentUser <agent user>
--unixAgentGroup <agent group>
--remoteServer <your existing ElectricFlow server>
--unixServerUser <server user>
--unixServerGroup <server group>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<server group>* is the group who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.
- *<agent group>* is the group that owns the ElectricFlow agent process.

Repository Server Installation

This installation example installs an ElectricFlow repository server, an agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
./ElectricFlow-<version>
--mode silent
--installRepository
--installAgent
--unixAgentUser <agent user>
--unixAgentGroup <agent group>
```

```
--unixServerUser <server user>
--unixServerGroup <server group>
--remoteServer <your existing ElectricFlow server>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<server group>* is the group who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.
- *<agent group>* is the group that owns the ElectricFlow agent process.

Tools Only Installation

This installation example installs only the ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
./ElectricFlow-<version> --mode silent
```

Windows Silent Installation Examples

The following examples are command strings to use for unattended (silent) installations. In many instances, the command text with the associated options wraps to the following lines.

IMPORTANT: You must enter the installation command and all options on a single line.

Complete ElectricFlow Installation

This installation installs the ElectricFlow server, including the web, repository, and database servers, one agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
ElectricFlow-<version>.exe
--mode silent
--installServer
--installAgent
--installDatabase
--installWeb
--installRepository
--windowsServerUser <server user>
--windowsServerDomain <domain>
--windowsServerPassword <password>
--windowsAgentUser <agent user>
--windowsAgentDomain <domain>
--windowsAgentPassword <password>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.

Agent Installation

This installation example installs an ElectricFlow agent, a repository server, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
ElectricFlow-<version>.exe
--mode silent
--installRepository
--installAgent
--windowsAgentUser <agent user>
--windowsAgentDomain <domain>
--windowsAgentPassword <password>
--windowsServerUser <server user>
--windowsServerDomain <domain>
--windowsServerPassword <password>
--remoteServer <your existing ElectricFlow server>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.

ElectricFlow Agent Installation (Full Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. This installation also installs Tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
ElectricFlow-<version>.exe --mode silent --installAgent --windowsAgentUser
<agent user> --windowsAgentDomain <domain> --windowsAgentPassword <password>
```

Where:

- *<agent user>* is the user who owns the ElectricFlow agent process.

ElectricFlow Agent Installation (Agent-Only Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure. This example uses the 64-bit agent-only installer file.

- Enter:

```
ElectricFlowAgent-x64-<version>.exe --mode silent --windowsAgentUser <agent
user> --windowsAgentDomain <domain> --windowsAgentPassword <password>
```

Where:

- *<agent user>* is the user who owns the ElectricFlow agent process.

Web Server and an Agent

This installation example installs an ElectricFlow web server, an agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
ElectricFlow-<version>.exe
--mode silent
--installAgent
--installWeb
--windowsAgentUser <agent user>
--windowsAgentDomain <domain>
--windowsAgentPassword <password>
--remoteServer <your existing ElectricFlow server>
--windowsServerUser <server user>
--windowsServerDomain <domain>
--windowsServerPassword <password>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.

Repository Server Installation

This installation example installs an ElectricFlow repository server, an agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
ElectricFlow-<version>.exe
--mode silent
--installRepository
--installAgent
--windowsAgentUser <agent user>
--windowsAgentDomain <domain>
--windowsAgentPassword <password>
--windowsServerUser <server user>
--windowsServerDomain <domain>
--windowsServerPassword <password>
--remoteServer <your existing ElectricFlow server>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.

Tools Only Installation

This installation example installs only the ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
ElectricFlow-<version>.exe --mode silent
```

Linux Repository Server Silent Installation

This installation example installs an ElectricFlow repository server, an agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
./ElectricFlow-<version>
--mode silent
--installRepository
--installAgent
--unixAgentUser <agent user>
--unixAgentGroup <agent group>
--unixServerUser <server user>
--unixServerGroup <server group>
--remoteServer <your existing ElectricFlow server>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<server group>* is the group who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.
- *<agent group>* is the group that owns the ElectricFlow agent process.

Windows Repository Server Silent Installation Example

This installation example installs an ElectricFlow repository server, an agent, and ElectricFlow tools. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

- Enter:

```
ElectricFlow-<version>.exe
--mode silent
--installRepository
--installAgent
--windowsAgentUser <agent user>
--windowsAgentDomain <domain>
--windowsAgentPassword <password>
--windowsServerUser <server user>
--windowsServerDomain <domain>
--windowsServerPassword <password>
--remoteServer <your existing ElectricFlow server>
```

Where:

- *<server user>* is the user who owns the ElectricFlow server, repository server, and web server processes.
- *<agent user>* is the user who owns the ElectricFlow agent process.

Non-Server Platform Agent Installation Method

To install agents and tools on machines that are not supported ElectricFlow server platforms, you must use the appropriate UNIX installer instead of the `./ElectricFlow-<version>` installer file. For more information about supported agent platforms, see [Supported Agent Platforms](#) on page 2-10.

You must use the `commander_<OSType>.bin` file to install the ElectricFlow agent software on a Solaris, HP-UX, or Mac OS agent machine. You can find copies of the various agent UNIX installers on the Electric Cloud FTP site.

Installing Agents with the UNIX Installer

Use the `commander_<OSType>.bin` file to install the ElectricFlow agent software on a Solaris, HP-UX, or Mac OS agent machine.

1. Obtain a copy of `commander_<OSType>.bin`
2. Log in as **root**.
3. Run `chmod +x ./commander_<OSType>.bin` to ensure the installer is executable.
4. Run `./commander_<OSType>.bin` to begin the ElectricFlow agent installation on a UNIX machine.

Where `<OSType>` is the agent platform.

The following information is displayed.

```
Checking installer integrity, please wait...
ElectricCommander 6.0.0.94439 for Linux Installer
Copyright 2006-2015 Electric Cloud, Inc. All rights reserved.
```

Press CTRL-C to exit at any time.

Press Enter to accept default settings.

```
log file: /tmp/commander_install_20150713_151337.log
This suite installer can install several different product options.
Note: The default is to install everything.
Which products would you like to install (agent, tools): agent
```

5. Enter: agent

Note: When you enter the “agent” option, an agent and tools will be installed.

6. Press **Enter** to begin the ElectricFlow UNIX agent installation.

The following information is displayed.

Installing agent and tools.

```
Where would you like the software to be installed?
NOTE: The destination should NOT be an nfs filesystem.
If no directory is specified, by default an “electricCloud” directory is created
in /opt.
Enter destination directory (default is /opt): <directory path>
```

7. Enter the destination directory path.

The following information is displayed.

Enter a pre-existing user to own installed agent files and run agent processes:

8. Enter the name of the user who owns the ElectricFlow agent process.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.  
Please confirm if you would like to proceed [y/N]
```

The following information is displayed.

```
Enter a pre-existing group to own installed agent files and run agent processes:
```

9. Enter the name of the group who owns the ElectricFlow agent process.

The following information is displayed.

```
Enter the agent port (default is 7800):
```

10. Accept the default port or specify a different port to eliminate any conflicts with your existing system configuration.
11. Press **Enter** to start the agent software installation.

When the installation is complete, the following information is displayed.

```
OK: Installation successful!
```

UNIX Agent Silent Installation

The UNIX installers also have a command-line interface for unattended (silent) installations. You can specify configuration options on the command line or in a configuration file.

UNIX Silent Installation Command Arguments

The following table lists the arguments for running a silent install with the UNIX installer.

<code>-q</code>	This option causes the installer to operate in silent mode. The default installation options will be used unless overridden in a configuration file or on the command line.
<code>-f</code>	This option forces the installer to remove and replace any existing files in the destination directory. This option does not uninstall the previous version, it completely removes the directory and writes a new one. For upgrades, see Roadmap to Upgrade ElectricFlow on page 6-1.
<code>--config</code>	This option can be used to specify a file containing configuration variables and values.

Running a Silent UNIX Installation

Use the following procedure to run a silent UNIX installation.

- Enter the following command to begin a silent install:

```
commander_<OStype>.bin -q <arguments>
```

Where:

- `<OStype>` is the agent platform.
- `<arguments>` represents any additional silent install arguments you add to the command.

The following example command-line string uses a configuration file to override default values on a silent install:

```
commander_sun4u_SunOS.bin -q -f --config myconfig
```

Example Configuration File Parameters

The following examples are parameter values to use for an unattended (silent) installation configuration file.

Agent Install Parameters

```
EC_INSTALL_TYPE=agent
DESTINATION_DIR="/opt"
AGENT_USER_TO_RUN_AS=(user for agent files and processes)
AGENT_GROUP_TO_RUN_AS=(group for agent)
EC_AGENT_PORT=(agent port number)
```

Tools Install Parameters

```
EC_INSTALL_TYPE=tools
DESTINATION_DIR="/opt"
USER_TO_RUN_AS="user"
GROUP_TO_RUN_AS="group"
```

Installing or Upgrading Remote Agents

You can install or upgrade remote hosts that are running Linux (x86 or x64), Windows (x86 or x64), MacOS, Solaris, Solaris x86, or HP-UX.

Prerequisites

General Prerequisites

- You must have an artifact repository server installed.
- The user that you specify in the **Authentication Options** dialog box must have administrator privileges on the target machines.
- The agent service user of the driving agent must have administrator rights on the target machine.

Linux and UNIX Prerequisites

- The target machines must be running the SSH daemon.
- For SSH, the user account on the target machine must have passwordless sudo configured for running the installer with root privileges.

For example, in the `/etc/sudoers` file, you must add "`<username> ALL= (ALL) NOPASSWD:ALL`".

- For each target machine that you want to upgrade, ensure that `PasswordAuthentication=yes` in the `/etc/ssh/sshd_config` file. For example:

```
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication yes
```

On SUSE platforms, `PasswordAuthentication=no` is the default.

Windows Prerequisites

Windows Driving Resources

- They must be in the same Windows domain as the remote windows hosts where you will install the agents.
- They must be in the same zone as the zone where you will install the agents.
- Powershell 3.0 or newer must be installed on older versions of Windows. To check the Powershell version:
 1. Open Powershell on the host by selecting **Start >All Programs > Accessories > Windows PowerShell > Windows PowerShell**.
 2. Enter the `Get-Host | Select-Object Version` command.
- Windows Remote Management (WinRM) must be installed and configured.
 - WinRM is installed automatically with all currently-supported versions of Windows.
 - The WinRM service starts automatically on Windows Server 2008. On Windows Vista, the service must be started manually.

- Before using a domain user to install agents on remote hosts, you must enable multi-hop support for WinRM on the driving resource system and on each remote host. For details, see the *Enabling Multi-Hop Support for Windows Remote Management Before Installing or Upgrading Remote Agents* KB article at <https://helpcenter.electric-cloud.com/hc/en-us/sections/200516863-ElectricFlow-KB>.
- You can enable the WS-Management protocol on the local system and set up the default configuration for remote management by using the `winrm quickconfig` command.
- Maximum Powershell memory must be set to at least 1024 MB. To set the memory setting to 1024 MB:
 1. Open a Windows command prompt as Administrator.
 2. Enter the `winrm set winrm/config/winrs @{MaxMemoryPerShellMB="1024"}` command.

Remote Windows Hosts

- Maximum Powershell memory (MaxMemoryPerShellMB) must be set to at least 1024 MB.
- WinRM must be installed and configured.
 - WinRM is installed automatically with all currently-supported versions of Windows.
 - The WinRM service starts automatically on Windows Server 2008. On Windows Vista, the service must be started manually.

You can enable the WS-Management protocol on the local computer and set up the default configuration for remote management by using the `winrm quickconfig` command.

Permissions for Installing or Upgrading Remote Agents

The following table describes the user permissions required to install or upgrade remote agents. For information about modifying permissions, see the “Access Control” section in the “Automation Platform” chapter of the *ElectricFlow User Guide* at http://docs.electric-cloud.com/eflow_doc/FlowIndex.html.

Allowed action	Object	Required privilege ("allow")
Install a resource in a zone	The “Resources” system object	Read
		Modify
		Execute
	Zone (such as <code>US1</code>)	Read
		Modify
Upgrade a resource	Resource (such as <code>ResourceA</code>)	Read
		Modify
		Execute

Installing Remote Agents Using the Web Interface



In the **Cloud > Resources** page of the Automation Platform, click the **+** (add) menu and then click **Install Resource(s)** to install resources on host machines to use as agent hosts for your ElectricFlow resources. If the **Install Resource(s)** menu option is not visible, you must log out and then log in as a user with the required permissions. For details about required permissions, see [Permissions for Installing or Upgrading Remote Agents](#) on page 3-69

Note: On Solaris platforms, the installation directory path must have fewer than 70 characters.

The **Install Resource(s)** dialog box appears:

Install Resources			
Prerequisites Ensure that the ElectricFlow agent installer is available for your platform.			
	Platform	Published Versions	Actions
	Linux x86	Not published	Publish Installer
<input checked="" type="radio"/>	Linux x64	7.0.0.110595 ▼	Re-Publish Installer
	Windows x86	Not published	Publish Installer
<input type="radio"/>	Windows x64	7.0.0.110595 ▼	Re-Publish Installer
<input type="radio"/>	MacOS	4.2.0.56612 ▼	Re-Publish Installer
<input type="radio"/>	Solaris	4.2.0.56612 ▼	Re-Publish Installer
	Solaris x86	Not published	Publish Installer
<input type="radio"/>	HP-UX	4.2.0.56612 ▼	Re-Publish Installer
Cancel		Next	

If an ElectricFlow agent installer is already published and available for your desired platform, then you can select it to continue with the installation. Otherwise, click the appropriate **Publish Installer** or **Re-Publish Installer** link for the appropriate platform, then click **Next**.

The **Run Procedure / Publish Installer** page opens:

Enter the following information:

Parameters

- **fromDir**—Directory on the resource used for publishing. For example, enter `/var/tmp` for Linux or `C: /<temp>` for Windows.
- **platform**—Host platform type.
- **repository**—Repository name. Use **default** to use the repository installed during the ElectricFlow installation or enter another name.
- **resource**—Name of the resource used for publishing.
- **version**—Build version for the ElectricFlow installation. For example, enter `5.0.0.56390`. You can find the build version from the file name of the ElectricFlow installer.

(Optional) Advanced

Set advanced options if needed.

Click **Run**.

When the Publish Installer procedure runs, you can see the job status on the **Job Details** page.

If you want to see the published installers, click the **Artifacts** tab and then click the `com.electriccloud:installer` artifact. The **Artifact Details** page lists the published installers.

Note: You can republish a new agent installer version. To do so, return to the **Cloud > Resources** page and



choose (add) > **Install Resource(s)** and click the radio button for the installer.

If the **Install Resource(s)** menu option is not visible, you must log out and then log in as a user with the required permissions. For details about required permissions, see [Permissions for Installing or Upgrading Remote Agents](#) on page 3-69

The **Install Targets** dialog box appears:

Install Resources

Install Targets: Details for the target host machines for installation

Installation target hostnames or IP addresses:

Required

Zone: Default ▼

Required

Driving Resource: Default ▼

Required

Resource Name Format Hostname/IP ▼

Template: {name}

Cancel
Previous
Next

Enter the information for the target host machines as follows:

- (Required) **Installation target hostnames or IP addresses**—One or more host names or IP addresses separated by any combination of spaces, commas, semicolons, or newlines. Host names cannot contain spaces.
- (Required) **Zone**—Zone where you are installing agents. If only one zone exists, this option is grayed out. For multiple zones, only the zones for which you have permissions are displayed.

You can install agents to only one zone at a time. A functioning gateway must exist before an agent can be installed into the nondefault zone, so installing the first agent into a zone must be done manually.

- (Required) **Driving Resource**—Resource to perform all actions for installing a host on behalf of the server. This is the agent that installs the agents on the remote hosts.

The driving resource must belong to the same platform and zone as the hosts being installed or upgraded. Also, on Windows systems containing the driving resource, Windows PowerShell 3.0 or newer must be installed on older Windows versions.

- **Resource Name Format**—Naming scheme for newly-created resources to correspond to the hosts on which to perform the installation.

- **Hostname/IP**—Use fully-qualified domain names (FQDNs), which will be derived from the host names or IP addresses of the hosts.
- **Short name**—Use host names that do not include domain names. For example, the `host11` part of `host11.bigco.com`.

If you provided IP addresses for the target hosts for installation, resources are named using the short names of the FQDNs, which are derived from those IP addresses. If you provided short host names for installation, resources are named using the short names.

If you provided IP addresses for the target hosts for installation and the FQDNs cannot be determined, the installation will fail for those resources (but will continue with the rest of the installation). Also, because there might be duplicate resource names across domains and subdomains if you use short names, if a resource by that name already exists, the installation will fail for the resource (but will continue with the rest of the installation).

- **Custom**—Custom template for resource naming. The value in this field undergoes property expansion in a global context and is scanned for the `{name}`, `{counter}`, and `{shortname}` special tokens.

Click **Next**.

One of two **Authentication Settings** dialog boxes appears depending on the platform.

Linux- or UNIX-Based Platforms

The screenshot shows a dialog box titled "Install / Upgrade Resources". It has a dark header bar with the text "Authentication Options" and "Credentials for logging into target machines". Below this, there is a section for "Authentication Type" with a dropdown menu currently set to "SSH User". Underneath, there are two text input fields labeled "User Name" and "Password". At the bottom of the dialog, there are three buttons: "Cancel", "Previous", and "Next". The "Next" button is highlighted in blue.

From the **Authentication Type** menu, choose **SSH User** or **SSH Key** and enter authentication information required to connect to the remote machines:

- **SSH User:**
 - (Required) **User Name**—User name. This user must have administrator privileges on the target machines.
 - (Required) **Password**—Password for the user name.
- **SSH Key:**
 - (Required) **User Name**—User name. This user must have administrator privileges on the target machines.
 - (Required) **Public Key Path**—Path to the SSH public key file.
 - (Required) **Private Key Path**—Path to the SSH private key file.
 - (Optional) **Passphrase**—Passphrase for unlocking the private key file.

Windows Platforms

The screenshot shows a dialog box titled "Install / Upgrade Resources". Below the title bar is a dark grey header with the text "Authentication Options" and "Credentials for logging into target machines". The main area of the dialog contains the following fields:

- Authentication Type:** A dropdown menu currently showing "Domain User".
- User Name:** A text input field.
- Password:** A text input field.
- Domain:** A text input field.

At the bottom of the dialog, there are three buttons: "Cancel", "Previous", and "Next". The "Next" button is highlighted in blue.

Note: Before using a domain user to install agents on remote hosts, you must enable multi-hop support for WinRM on the driving resource system and on each remote host. For more information, see the *Enabling Multi-Hop Support for Windows Remote Management Before Installing or Upgrading Remote Agents* KB article at <https://helpcenter.electric-cloud.com/hc/en-us/sections/200516863-ElectricFlow-KB>.

Enter the authentication information required to connect to the remote machines:

- (Required) **User Name**—User name. This user must have administrator privileges on the target machines.
- (Required) **Password**—Password for the user name.
- (Required) **Domain**—User's Windows domain.

Click **Next**.

One of two **Agent Configuration** dialog boxes appears depending on the platform.

Linux- or UNIX-Based Platforms

The screenshot shows a dialog box titled "Install / Upgrade Resources". Below the title bar is a dark grey header with the text "Agent Configuration" and "Options passed to the installer." in white. The main area of the dialog is white and contains the following elements:

- Authentication Type:** A dropdown menu with "Local User" selected.
- Agent Service User:** A text input field.
- Agent Service Password:** A text input field.
- Agent type:** A dropdown menu with "Concurrent" selected.
- Advanced:** A section header with a right-pointing triangle icon.
- Buttons:** At the bottom, there are three buttons: "Cancel" (grey), "Previous" (grey), and "Next" (blue).

Provide the agent settings to be passed to the ElectricFlow agent installer. The agent settings appear in two sections: Information for the user account to be used to run the agent and the agent or resource configurations such as agent port:

- (Required) **Agent Service User**—User that the ElectricFlow agent runs as.
- (Required) **Agent Service Group**—Group that the ElectricFlow agent runs as.

Windows Platforms

The screenshot shows a window titled "Install / Upgrade Resources". Inside, there is a section labeled "Agent Configuration" with the subtitle "Options passed to the installer." Below this, there are several input fields and dropdown menus:

- Authentication Type:** A dropdown menu currently showing "Domain User".
- Agent Service User:** A text input field.
- Agent Service Password:** A text input field.
- Agent Service Domain:** A text input field.
- Agent type:** A dropdown menu currently showing "Concurrent".

Below these fields is a link that says "► Advanced". At the bottom of the window are three buttons: "Cancel", "Previous", and "Next". The "Next" button is highlighted in blue.

Provide the agent settings to be passed to the ElectricFlow agent installer. The agent settings appear in two sections: Information for the user account to be used to run the agent and the agent or resource configurations such as agent port:

- **Local User:**
 - (Required) **User Name**—User name.
 - (Required) **Password**—Password for the user name.
- **Domain User:**
 - (Required) **User Name**—User name.
 - (Required) **Password**—Password for the user name.
 - (Required) **Domain**—User's Windows domain.
- **Built-in User Account:**
 - (Required) **Service Account**—**LocalSystem** is the only option available in this release.

All Platforms

- **Agent type**—Agent type. (This menu appears only if a mixed-mode license is installed on the server for both registered and concurrent hosts.)
 - If the license on the server is a concurrent resource license, the host type defaults to **Concurrent**, and this menu does not appear.
 - If the license on the server is a registered host license, the host type defaults to **Registered**, and this menu does not appear.
 - If the license on the server is a mixed-mode license (concurrent resources and registered hosts), you must choose the host type when adding a resource.

(Optional) Click **Advanced** to view additional options:

▼ Advanced

Agent Port: 7800

Install Directory: C:\Program Files\Electric Cloud

Data Directory: C:\ProgramData\Electric Cloud

Trusted Agent: ☐

Workspace: ▼

Additional Parameters: --agentMaxMemoryMB 256
--agentWindowsServiceStartType delayed_auto_start

Cancel Previous Next

Complete the advanced options as follows:

- **Agent Port**—Port for communication between the local agent and the server. The default is 7800.
- **Install Directory**—Path to the install directory. Spaces are allowed. Do not use quotation marks.
- **Data Directory**—Path to the directory where your modified ElectricFlow files are stored (configuration and log files). Spaces are allowed. Do not use quotation marks.

- **Trusted Agent**—Check this check box if all agent machines being installed or upgraded are trusted. For more information, see the “Switching a Non-Trusted Agent to Trusted” section in the “Automation Platform” chapter of the *ElectricFlow User Guide* at http://docs.electric-cloud.com/eflow_doc/FlowIndex.html.
- **Workspace**—Name of the workspace to set as the default workspace for the new resource(s).
- **Additional Parameters**—Parameters that are not available in the UI. For example, --agentMaxMemoryMB 256. All arguments for a parameter must be on the same line.

Click **Next**.

The **Post Installation Step** dialog box appears. (You can skip the dialog box by clicking **Next** at this point.)

You can choose **Project** or **Plugin**. The following examples use **Project**.

Enter a project name into the **Name** dialog box. For example:

The screenshot shows a dialog box titled "Install Resources". Inside, there is a section titled "Post Installation Step" with the instruction: "Provide details for the procedure that should be run on the resources after installation. You can skip this screen if no post installation step is required." Below this, there are two radio buttons: "Project" (which is selected) and "Plugin". Under the "Project" radio button, there is a "Name:" label followed by a text input field containing "Installations and Upgrades". Below the "Name:" field, there is a "Procedure:" label followed by a text input field containing "Run Deployment Tests". At the bottom of the dialog box, there are three buttons: "Cancel", "Previous", and "Next" (which is highlighted in blue).

Enter a procedure name. If the procedure has parameters, the **Parameters** menu appears, which includes the parameter fields for the selected procedure. For example:

Install Resources

Post Installation Step
Provide details for the procedure that should be run on the resources after installation. You can skip this screen if no post installation step is required.

☒ Project ☐ Plugin

Name:

Procedure:

▼ Parameters

Log test results: ☐

Test Suite Name:

Required

Cancel

Previous

Next

Enter the parameters for the selected procedure. For example:

Install Resources

Post Installation Step
Provide details for the procedure that should be run on the resources after installation. You can skip this screen if no post installation step is required.

☒ Project ☐ Plugin

Name:
Installations and Upgrades

Procedure:
Run Deployment Tests

Parameters

Log test results: ☒

Test Suite Name: WindowsAgentTests

Required

Cancel

Previous

Next

Note: Credential parameters are not supported in post-installation steps.

Click **Next**.

The **Ready to Install** dialog box appears, which summarizes your settings. For example:

3-80

Install Resources		
Ready to Install		
Review the settings before starting the install.		
Target host(s):	win1, win2	
Zone:	default	
Template:	Short name	
Windows UserName:	admin	
Windows Password:	[PROTECTED]	
Windows Domain:	mydomain.com	
Agent Service User:	admin	
Agent Service Password:	[PROTECTED]	
Agent Port:	7800	
Install Directory:	C:\Program Files\Electric Cloud	
Data Directory:	C:\ProgramData\Electric Cloud	
Trusted Agent:	Untrusted	
Additional Parameters:	--agentWindowsServiceStartType delayed_auto_start	
Post Install Step:	Installations and Upgrades Run Deployment Tests	
Parameters:	{"Log test results": "true", "Test Suite Name": "WindowsAgentTests"}	
Select Finish to run the install.		
Cancel	Previous	Finish

For more than five hosts, the number of hosts appears (instead of a space-separated list of host names).

Review your settings in the **Ready to Install** dialog box, then click **Finish** to start the installation or upgrade. The **Job Details** page appears.

When the installation or upgrade finishes, you can return to the **Resources** page to see the hosts that were just installed. To verify a resource version, click a resource name (in table view) to open the **Resource Details** panel for that resource.

Installing Remote Agents Using the API

You can automate remote installations by using scripts that call the following ElectricFlow APIs. These APIs provide the same remote installation capability as the web interface.

Parameters for Running the Procedure to Publish an Installer to the Artifact Repository

The `Publish Installer` procedure is used to publish an ElectricFlow installer to the artifact repository. The parameters for the `Publish Installer` procedure appear below. These parameters have equivalent options in the web interface.

Parameter	Description
fromDir	Directory where the installer is located. Only the installer itself will be published.
platform	Platform of the installer. The possible values are Linux_x86, Linux_x64, Windows_x86, Windows_x64, Darwin, SunOS, SunOSx86, or HP-UX.
repository	ElectricFlow repository server where the installer will be published.
resource	Resource used by the publishing procedure.
version	Version of the installer. Use the format <code><major>.<minor>.<patch>.<build></code> . Because the build number changes with every build of the corresponding plugin, you should check this number often and update it if needed.

Parameters for Running the Procedure to Perform the Installations

A top-level wrapper procedure named `Centralized Agent Management` is used for installer publishing. This procedure calls the `Install Agent` procedure to publish installers to the repository.

The parameters for the `Centralized Agent Management` procedure appear below. Most of these parameters have equivalent options in the web interface.

Parameter	Windows Support	Other Platform Support	Description
Installation or Upgrade Parameters Based on Platform and Target Host			
isUpgrade	Yes	Yes	Specifies whether you are installing agents or upgrading existing agents. Use <code>0</code> if installing agents and <code>1</code> if upgrading agents.
resources	Yes	Yes	(Upgrades only) Space-delimited list of resources to upgrade. The resources should belong to the same platform (such as Windows or Linux) and the same zone.
hostnames	Yes	Yes	(Installations only) List of hostnames or IP addresses on which to install the agent. You can delimit the host names or IP addresses by using any combination of spaces, commas, semicolons, or newlines. Host names cannot contain spaces. The hosts should belong to the same platform and the same network.

Parameter	Windows Support	Other Platform Support	Description
zoneName	Yes	Yes	ElectricFlow zone in which the agents will be installed or upgraded.
drivingResource	Yes	Yes	Existing resource within the selected zone that will be used to perform all proxy commands for installing or upgrading the target hosts. The driving resource must belong to the same platform and the same zone as the hosts.
resourceNameTemplate	Yes	Yes	(Installations only) String templates. Used for naming the resources in ElectricFlow for the corresponding installation target hosts. The special tokens {name}, {counter}, and {shortname} can be used as place holders for the host name and for incrementing counters in the template value. For example, {name}-Win-{counter}.
version	Yes	Yes	Version of the agent previously published to the repository server to use. For example, 7.0.0.110954.
platform	Yes	Yes	Platform of the resources being upgraded or installed. The possible values are Linux_x86, Linux_x64, Windows_x86, Windows_x64, Darwin, SunOS, SunOSx86, or HP-UX.
Connection Parameters for Communicating With the Target Hosts			
connectionType	Yes	Yes	Type of authentication to use when connecting to the target machines. Use <code>WINDOWS</code> when installing or upgrading resources on Windows. Otherwise, use either <code>SSH_PASSWORD</code> or <code>SSH_KEY</code> depending on whether the SSH user name and password will be used or an SSH key will be used.
connectionCredential	Yes	Yes	Connection credentials based on the type of authentication used for connecting to the target hosts. Use the user account name and password if connectionType is <code>WINDOWS</code> or <code>SSH_USER</code> . Use the user name and SSH key passphrase if connectionType is <code>SSH_KEY</code> .
connectionDomain	Yes	–	(Windows platforms only) Name of the windows domain for the specified user account.

Parameter	Windows Support	Other Platform Support	Description
connectionPublicKey	–	Yes	Path to the SSH public key file on the driving resource.
connectionPrivateKey	–	Yes	Path to the SSH private key file on the driving resource.
Agent Configuration Parameters for Installing Agents on Target Hosts (There are additional parameters for agent configuration that are not listed here. To specify additional agent configuration parameters, use the Additional Parameters field as described above.)			
agentPort	Yes	Yes	(Installations only) Port that will be used by the agents installed on the target hosts. The default is 7800.
installDirectory	Yes	Yes	(Installations only) Path to the installation directory on the target hosts.
dataDirectory	Yes	Yes	(Installations only) Path to the data directory on the target hosts.
agentWorkspaceName	Yes	Yes	(Installations only) Name of the workspace that will be used by the agents installed on the target hosts.
agentCredential	Yes	Yes	(Installations only) Password for the user account that the installed agent service should run as. (This is not applicable if you are using <code>LocalSystem</code> as the agent user on Windows.)
agentDomain	Yes	–	(Installations only) Windows domain name for the agent user, if the user account belongs to a domain.
trustedAgent	Yes	Yes	(Installations only) Specifies whether the agent will be installed as a trusted agent. A trusted agent is "certificate verified," which means that the agent verifies the server or upstream agent's certificate. If this is set to 1, the agent will be installed as a trusted agent. For more information, see Installing or Upgrading Remote Agents on page 3-68. If there are multiple zones, and you try to install a trusted agent to a custom zone through the gateway, both gateway agents must be in trusted mode. Otherwise, the agent will be installed but will be unreachable.

Parameter	Windows Support	Other Platform Support	Description
agentGroup	–	Yes	(Installations only) Group of the user account that the installed agent service should run as.
hostType	Yes	Yes	(Installations only) Host type when the service is licensed for both concurrent and registered agent hosts. Specify <code>CONCURRENT</code> to install concurrent host agents. Specify <code>REGISTERED</code> to install registered host agents.
additionalParameters	Yes	Yes	<p>Specifies additional agent configuration parameters.</p> <p>Note: Do not use this parameter to specify any agent configuration parameters (listed above) in conjunction with their equivalent arguments in the standard agent installer. Doing so might cause unpredictable behavior. For example, do not use <code>ectool runProcedure EC-AgentManagement-1.2.0.111083 --procedureName "Centralized Agent Management" --actualParameter "additionalParameters=--agentPort 7800"</code>. Instead, use <code>ectool runProcedure EC-AgentManagement-1.2.0.111083 --procedureName "Centralized Agent Management" --actualParameter "agentPort=7800"</code>. For a complete list of agent installer arguments, see Silent Install Arguments on page 3-54.</p>
Post-Installation Parameters			
postStepProjectName	Yes	Yes	Name of the project that contains the procedure to run as a post-installation step on each of the just-installed or upgraded resources.
postStepProcedureName	Yes	Yes	Name of the procedure to run as a post-installation step on each of the just-installed or upgraded resources.
postStepParameters	Yes	Yes	Parameters in JSON format. You must escape special characters that are not supported by JSON.

Examples for Publishing an ElectricFlow Installer to the Artifact Repository

As mentioned above, `Publish Installer` is a procedure used to publish an ElectricFlow installer to the artifact repository. Note that the `ec-perl` and `DSL` scripts below contain commands for using the promoted plugin version without hard-coding the version.

ec-perl Example

```
use strict;
use ElectricCommander;
my $ec = new ElectricCommander();
my $xpath = $ec->getPlugin("EC-AgentManagement");
my $pluginProject = $xpath->findvalue('//projectName')->value;
my $jobId = $ec->runProcedure($pluginProject, {
    procedureName => "Publish Installer",
    actualParameter => [
        {
            actualParameterName => 'fromDir',
            value => '/home/build'
        },
        {
            actualParameterName => 'platform',
            value => 'Linux_x64'
        },
        {
            actualParameterName => 'repository',
            value => 'default'
        },
        {
            actualParameterName => 'resource',
            value => 'local'
        },
        {
            actualParameterName => 'version',
            value => '7.0.0.110576'
        }
    ]
})->findvalue('//jobId')->value();
print "jobId is: $jobId";
```

ectool Example

```
ectool runProcedure EC-AgentManagement-1.2.0.111083 --procedureName "Publish Installer"
--actualParameter "fromDir=/home/build"
--actualParameter "platform=Linux_x86"
--actualParameter "repository=default"
--actualParameter "resource=local"
--actualParameter "version=7.0.0.110576"
```

DSL Example

```
ectool evalDsl "
def pluginProject = getPlugin(pluginName: 'EC-AgentManagement').projectName;
runProcedure(pluginProject, procedureName: 'Publish Installer',
    actualParameter: [
        fromDir: '/home/build',
        platform: 'Linux_x86',
        repository: 'default',
```

```

        resource:  'local',
        version:   '7.0.0.110576'
    ]
}"

```

Examples for Installing Remote Agents

As mentioned above, `Centralized Agent Management` is a top-level wrapper procedure that calls the `Install Agent` procedure for installing or upgrading individual agents on the agent hosts. Note that the DSL scripts below contain commands for using the promoted plugin version without hard-coding the version.

ectool Example (Linux)

```

ectool runProcedure EC-AgentManagement-1.2.0.111083 --procedureName "Centralized Agent
Management"
--actualParameter "isUpgrade=0"
--actualParameter "platform=Linux_x64"
--actualParameter "version=7.0.0.110954"
--actualParameter "zoneName=default"
--actualParameter "hostnames=192.168.4.208 192.168.4.210"
--actualParameter "drivingResource=local"
--actualParameter "resourceNameTemplate=prefix-{counter}"

--actualParameter "agentCredential=agentCredential"
--actualParameter "agentGroup=remote"
--actualParameter "agentPort=7800"

--actualParameter "trustedAgent=0"
--actualParameter "hostType=CONCURRENT"

--actualParameter "connectionType=SSH_PASSWORD"
--actualParameter "connectionCredential=connectionCredential"

--credential connectionCredential=remote
--credential agentCredential=remote

```

ectool Example (Linux with Post-Installation Procedure)

```

ectool runProcedure EC-AgentManagement-1.3.0.0 --procedureName "Centralized Agent Mana
gement" \
--actualParameter "isUpgrade=0" \
--actualParameter "platform=Linux_x64" \
--actualParameter "version=7.0.0.111324" \
--actualParameter "zoneName=default" \
--actualParameter "hostnames=10.200.1.109" \
--actualParameter "drivingResource=local" \
--actualParameter "resourceNameTemplate=prefix-{counter}" \
--actualParameter "agentCredential=agentCredential" \
--actualParameter "agentGroup=vagrant" \
--actualParameter "agentPort=7800" \
--actualParameter "trustedAgent=0" \
--actualParameter "hostType=CONCURRENT" \
--actualParameter "connectionType=SSH_PASSWORD" \
--actualParameter "connectionCredential=connectionCredential" \
--actualParameter "postStepProcedureName=postinstall" \
--actualParameter "postStepProjectName=postinstall" \
--actualParameter "postStepParameters={\"another_resource\": \"local\", \"checkbox\":
\"true\", \"credential\": \"\", \"dropdown\": \"option3\", \"project\": \"Default\",
\"radio\": \"option3\", \"required1\": \"required1\", \"saved_filter\":

```

```
\"/projects/EC-Examples/ec_savedSearches/Test Outcome Filter\", \"textarea\":  
\"Lorem ipsum amet\ntempor incididunt \nquis nostrud\nconsequat.\")\" \  
--credential connectionCredential=vagrant \  
--credential agentCredential=vagrant
```

ectool Example (Windows)

```
ectool runProcedure EC-AgentManagement-1.2.0.111083 --procedureName "Centralized Agent  
Management"  
--actualParameter "isUpgrade=0"  
--actualParameter "platform=Windows_x64"  
--actualParameter "version=7.0.0.110954"  
--actualParameter "zoneName=default"  
--actualParameter "hostnames=10.1.216.227"  
--actualParameter "drivingResource=3EC-IT-3614"  
--actualParameter "resourceNameTemplate={name}"  
  
--actualParameter "connectionDomain=electric-cloud"  
--actualParameter "agentCredential=agentCredential"  
--actualParameter "agentPort=7800"  
  
--actualParameter "trustedAgent=0"  
--actualParameter "hostType=CONCURRENT"  
  
--actualParameter "connectionType=WINDOWS"  
--actualParameter "connectionCredential=connectionCredential"  
  
--credential connectionCredential=remote  
--credential agentCredential=remote
```

DSL Example (Linux)

In the following example, `agentGroup` is the group for the Linux user to connect to the remote machines.

```
ectool evalDsl "  
def pluginProject = getPlugin(pluginName: 'EC-AgentManagement').projectName;  
runProcedure(pluginProject, procedureName: 'Publish Installer',  
  actualParameter: [  
    isUpgrade: '0',  
    platform: 'Linux_x64',  
    version: '7.0.0.110954',  
    zoneName: 'default', hostnames: '192.168.4.210',  
    drivingResource: 'local',  
    resourceNameTemplate: 'prefix-{counter}',  
  
    agentCredential: 'agentCredential',  
    agentGroup: 'remote',  
    agentPort: '7800',  
  
    trustedAgent: '0',  
    hostType: 'CONCURRENT',  
  
    connectionType: 'SSH_PASSWORD',  
    connectionCredential: 'connectionCredential'  
  ],  
  credential: [  
    [  
      credentialName: 'connectionCredential',
```

```

        userName: 'remote',
        password: 'Rem0te2'
    ],
    [
        credentialName: 'agentCredential',
        userName: 'remote',
        password: 'Rem0te3'
    ]
]
)"

```

DSL Example (Linux with Post-Installation Procedure)

```

ectool evalDsl "
def pluginProject = getPlugin(pluginName: 'EC-AgentManagement').projectName;
runProcedure(pluginProject, procedureName: 'Centralized Agent Management',
    actualParameter: [
        isUpgrade: '0',
        platform: 'Linux_x64',
        version: '7.0.0.111324',
        zoneName: 'custom',
        hostnames: '10.200.1.109',
        drivingResource: 'local',
        resourceNameTemplate: '{shortname}-{name}-{counter} ',

        agentCredential: 'agentCredential',
        agentGroup: 'vagrant',
        agentPort: '7800',

        trustedAgent: '0',
        hostType: 'CONCURRENT',

        connectionType: 'SSH_PASSWORD',
        connectionCredential: 'connectionCredential',

        postStepProcedureName: 'postinstall',
        postStepProjectName: 'postinstall',
        postStepParameters: '{"another_resource": "local","checkbox":
"true","credential": "", "dropdown": "option3","project": "Default",
"radio": "option3","required1": "required1","required2": "req2",
"saved_filter": "/projects/EC-Examples/ec_savedSearches/Test Outcome
Filter","textarea": "Lorem ipsum amet\ntempor incididunt \nquis
nostrud\ nconsequat."}'
    ],
    credential: [
        [
            credentialName: 'connectionCredential',
            userName: 'remote',
            password: 'remote4'
        ],
        [
            credentialName: 'agentCredential',
            userName: 'remote',
            password: 'remote5'
        ]
    ]
]"

```

DSL Example (Windows)

In the following example, `connectionDomain` is the Windows domain for the Windows user to connect to the remote machines.

```
ectool evalDsl "  
def pluginProject = getPlugin(pluginName: 'EC-AgentManagement').projectName;  
runProcedure(pluginProject, procedureName: 'Publish Installer',  
  actualParameter: [  
    isUpgrade: '0',  
    platform: 'Windows_x64',  
    version: '7.0.0.110954',  
    zoneName: 'default',  
    hostnames: '10.1.216.235',  
    drivingResource: '10.1.216.227',  
    resourceNameTemplate: '{name}',  
  
    connectionDomain: 'electric-cloud',  
    agentCredential: 'agentCredential',  
    agentPort: '7800',  
  
    trustedAgent: '0',  
    hostType: 'CONCURRENT',  
  
    connectionType: 'WINDOWS',  
    connectionCredential: 'connectionCredential'  
  ],  
  credential: [  
    [  
      credentialName: 'connectionCredential',  
      userName: 'remote',  
      password: 'Rem0te6'  
    ],  
    [  
      credentialName: 'agentCredential',  
      userName: 'remote',  
      password: 'Rem0te7'  
    ]  
  ]  
)"
```

Upgrading Remote Agents Using the Web Interface

In the **Cloud > Resources** page of the Automation Platform, check the check boxes for resources that you want to upgrade on agent hosts. If the selected resources have different platforms or zones, an error message appears:

Resources belonging to different platforms and zones cannot be upgraded at the same time. Please select resources belonging to the same platform and the same zone.

If any selected resource is a proxy resource, an error message appears:

Proxy resources are not supported for upgrade. Select the resources acting as proxy agents directly for upgrade.



Then click the (add) menu and then click **Upgrade Resource(s)**. If the **Upgrade Resource(s)** menu option is not visible, you must log out and then log in as a user with the required permissions. For details about required permissions, see [Permissions for Installing or Upgrading Remote Agents](#) on page 3-69

Note: On Solaris platforms, the installation directory path must have fewer than 70 characters.

The **Upgrade Resources** dialog box appears. For example:

	Platform	Published Versions	Actions
<input type="radio"/>	Windows x86	7.1.0.113340 ▼	Re-Publish Installer
<input checked="" type="radio"/>	Windows x64	7.1.0.113340 ▼	Re-Publish Installer

Cancel Next

Either one or two platforms for the selected resource are listed. Two platforms are listed if the platform (such as Linux x86 or Linux x64) could not be determined.

Choose the platform and published version, click the appropriate **Re-Publish Installer** link, then click **Next**.

The **Run Procedure / Publish Installer** page opens:

Run Procedure / Publish Installer

Parameters

fromDir: Required

platform: Linux x86 Required

repository: default Required

resource: local Required

version: Required

Advanced

Priority: normal

Impersonation: ☒ Use pre-defined credential ☐ Use specific credential ☐ Use a specific user

Run Cancel

Enter the following information:

Parameters

- (Required) **fromDir**—Directory on the resource used for publishing. For example, enter `/var/tmp` for Linux or `C:/<temp>` for Windows.
- (Required) **platform**—Host platform type.
- (Required) **repository**—Repository name. Use **default** to use the repository installed during the ElectricFlow installation or enter another name.
- (Required) **resource**—Name of the resource used for publishing.
- (Required) **version**—Build version for the ElectricFlow installation. For example, enter `5.0.0.56390`. You can find the build version from the file name of the ElectricFlow installer.

(Optional) Advanced

Set advanced options if needed.

When the Publish Installer procedure runs, you can see the job status on the **Job Details** page.

If you want to see the published installers, click the **Artifacts** tab and then click the `com.electriccloud:installer` artifact. The **Artifact Details** page lists the published installers.

Click **Run**.

The **Upgrade Targets** dialog box appears.

Enter a list of resource names to upgrade into the **Upgrade target resource names** field. Names must be separated by any combination of spaces, commas, semicolons, or newlines. The resources must already exist, must be in the same zone, and must be of the same platform. For example:

Upgrade Resources

Upgrade Targets
Details for resources to upgrade.

Upgrade target resource names:*
local gclocal2, gclocal3

Zone:* default

Driving Resource:*

Cancel Previous Next

The **Zone** field displays the zone of the selected resources for upgrade. **Driving Resource** is the resource to perform all actions for installing or upgrading a host on behalf of the server. This is the agent that upgrades the agents on the remote hosts.

Click **Next**.

One of two **Authentication Settings** dialog boxes appears depending on the platform.

Linux- or UNIX-Based Platforms

The screenshot shows a dialog box titled "Install / Upgrade Resources". Inside, there's a section for "Authentication Options" with the subtitle "Credentials for logging into target machines". The "Authentication Type" is set to "SSH User" via a dropdown menu. Below this, there are input fields for "User Name" and "Password". At the bottom, there are three buttons: "Cancel", "Previous", and "Next".

From the **Authentication Type** menu, choose **SSH User** or **SSH Key** and enter authentication information required to connect to the remote machines:

- **SSH User:**
 - (Required) **User Name**—User name. This user must have administrator privileges on the target machines.
 - (Required) **Password**—Password for the user name.
- **SSH Key:**
 - (Required) **User Name**—User name. This user must have administrator privileges on the target machines.
 - (Required) **Public Key Path**—Path to the SSH public key file.
 - (Required) **Private Key Path**—Path to the SSH private key file.
 - (Optional) **Passphrase**—Passphrase for unlocking the private key file.

Windows Platforms

The screenshot shows a dialog box titled "Install / Upgrade Resources" with a tab labeled "Authentication Options". Below the tab is the text "Credentials for logging into target machines". The main area contains three input fields: "Authentication Type:" with a dropdown menu showing "Domain User", "User Name:" with a text box, "Password:" with a text box, and "Domain:" with a text box. At the bottom are three buttons: "Cancel", "Previous", and "Next".

Enter the authentication information required to connect to the remote machines:

- (Required) **User Name**—User name. This user must have administrator privileges on the target machines.
- (Required) **Password**—Password for the user name.
- (Required) **Domain**—User's Windows domain.

Click **Next**.

The **Post Upgrade Step** dialog box appears. (You can skip the dialog box by clicking **Next** at this point.)

You can choose **Project** or **Plugin**. The following example uses **Project**. Enter a project name, procedure name, and if the selected procedure has parameters, enter its parameters. For example:

Upgrade Resources

Post Upgrade Step
Provide details for the procedure that should be run on the resources after upgrade. You can skip this screen if no post upgrade step is required.

☒ Project ☐ Plugin

Name:

Procedure:

▼ Parameters

Log test results: ☒

Test Suite Name: Required

Note: Credential parameters are not supported in post-installation steps.

The **Ready to Upgrade** dialog box appears, which summarizes your settings. For example:

Upgrade Resources		
Ready to Upgrade Review the settings before starting the upgrade.		
Target resource(s):	local, gclocal2	
Zone:	default	
Template:	Hostname/IP	
Windows UserName:	build	
Windows Password:	[PROTECTED]	
Windows Domain:	mydomain.com	
Post Upgrade Step:	Installations and Upgrades Run Deployment Tests	
Parameters:	{"Log test results": "true", "Test Suite Name": "WindowsAgentTests"}	
Select Finish to run the upgrade.		
Cancel	Previous	Finish

For more than five hosts, the number of hosts appears (instead of a space-separated list of host names).

Review your settings in the **Ready to Upgrade** dialog box and click **Finish** to start the installation or upgrade. The **Job Details** page appears.

When the installation or upgrade finishes, you can return to the **Resources** page to see the hosts that were just upgraded. To verify a resource version, click a resource name (in table view) to open the **Resource Details** panel for that resource.

Upgrading Remote Agents Using the API

You can automate remote upgrades for multiple resources by using scripts that call ElectricFlow APIs to run those upgrades. These APIs provide the same remote upgrade capability as the web interface.

Parameters for Running the Procedure to Perform the Installations

See [Parameters for Running the Procedure to Perform the Installations](#) on page 3-82 above.

Examples for Upgrading Remote Agents

As mentioned above, `Centralized Agent Management` is a top-level wrapper procedure that calls the `Install Agent` procedure for installing or upgrading individual agents on the agent hosts. Note that the DSL scripts below contain commands for using the promoted plugin version without hard-coding the version.

ectool Example (Linux)

```
ectool runProcedure EC-AgentManagement-1.2.0.111083 --procedureName "Centralized Agent Management"
--actualParameter "isUpgrade=1"
```

```

--actualParameter "platform=Linux_x64"
--actualParameter "version=7.0.0.110954"
--actualParameter "zoneName=default"
--actualParameter "resources=4.208 4.210"
--actualParameter "drivingResource=local"

--actualParameter "connectionType=SSH_PASSWORD"
--actualParameter "connectionCredential=connectionCredential"
--actualParameter "agentCredential=agentCredential"

--credential connectionCredential=remote
--credential agentCredential=remote

```

ectool Example (Windows)

```

ectool runProcedure EC-AgentManagement-1.2.0.111083 --procedureName "Centralized Agent
Management"
--actualParameter "isUpgrade=1"
--actualParameter "platform=Windows_x64"
--actualParameter "version=7.0.0.110954"
--actualParameter "zoneName=default"
--actualParameter "resources=10.1.216.227"
--actualParameter "drivingResource=3EC-IT-3614"

--actualParameter "connectionDomain=electric-cloud"

--actualParameter "connectionType=WINDOWS"
--actualParameter "connectionCredential=connectionCredential"
--actualParameter "agentCredential=agentCredential"

--credential connectionCredential=remote
--credential agentCredential=remote

```

DSL Example (Linux)

```

ectool evalDsl "
def pluginProject = getPlugin(pluginName: 'EC-AgentManagement').projectName;
runProcedure(pluginProject, procedureName: 'Publish Installer',
  actualParameter: [
    isUpgrade: '1',
    platform: 'Linux_x64',
    version: '7.0.0.110954',
    zoneName: 'default',
    resources: '4.208',
    drivingResource: 'local',

    connectionType: 'SSH_PASSWORD',
    connectionCredential: 'connectionCredential',
    agentCredential: 'agentCredential'
  ],
  credential: [
    [
      credentialName: 'connectionCredential' ,
      userName: 'remote',
      password: 'Rem0te4'
    ],
    [

```

```
        credentialName: 'agentCredential' ,
    ]
}
)"
```

DSL Example (Windows)

```
ectool evalDsl "
def pluginProject = getPlugin(pluginName: 'EC-AgentManagement').projectName;
runProcedure(pluginProject, procedureName: 'Publish Installer',
    actualParameter: [
        isUpgrade: '1',
        platform: 'Windows_x64',
        version: '7.0.0.110954',
        zoneName: 'default',
        resources: '10.1.216.235',
        drivingResource: '10.1.216.227',

        connectionDomain: 'electric-cloud',

        connectionType: 'WINDOWS',
        connectionCredential: 'connectionCredential',
        agentCredential: 'agentCredential'

    ],
    credential: [
        [
            credentialName: 'connectionCredential',
            userName: 'remote',
            password: 'Rem0te5'
        ],
        [
            credentialName: 'agentCredential',
        ]
    ]
)"
```

Moving the Artifact Repository in Linux

In this scenario, the `/opt/electriccloud/electriccommander/conf/repository/wrapper.conf` file has these settings:

- `set.default.INSTALL_DIRECTORY=/opt/electriccloud/electriccommander`
- `set.default.DATA_DIRECTORY=/opt/electriccloud/electriccommander`

If the Linux server where the current artifact repository is stored is full and you want to move it to a new device with more disk space, map the artifact repository to a new network location.

In this procedure, the `set.default.DATA_DIRECTORY` value will change to the `NEW_DATA_DIRECTORY` value. The `REPOSITORY_BACKING_STORE` value in `/opt/electriccloud/electriccommander/conf/repository/server.properties` will remain the same, relative to the `DATA_DIRECTORY`, which will be `NEW_DATA_DIRECTORY`.

In the example, the `DATA_DIRECTORY` will be changed to `/vagrant_data`, and the `REPOSITORY_BACKING_STORE` value will be relative to this directory.

Note: The `REPOSITORY_BACKING_STORE` value is always relative to the `DATA_DIRECTORY` and cannot be set to a fully qualified absolute path. The solution in <http://ask.electric-cloud.com/questions/2192/how-to-configure->

[which-directory-the-artifact-repository-uses](#) will not work until you change the `DATA_DIRECTORY` as described in this procedure.

1. Publish the artifact under the `/opt/electriccloud/electriccommander/repository-data/<artifact_name>/<version>` directory:

```
ectool publishArtifactVersion --artifactName <artifact_name> --version <version>
--fromDirectory <directory> --compress 0
```

Example: To publish an artifact under the `/opt/electriccloud/electriccommander/repository-data/jdoe/2001/1.0.0` directory, enter:

```
ectool publishArtifactVersion --artifactName "jdoe:2001" --version "1.0.0"
--fromDirectory /tmp/job1 --compress 0
```

The `fromDirectory (/tmp/job1)` contains the files to be uploaded.

2. Stop the repository server:

```
/etc/init.d/commanderRepository stop
```

3. Move the artifact repository data from the `/DATA_DIRECTORY` directory to the `/NEW_DATA_DIRECTORY` directory:

```
mv /opt/electriccloud/electriccommander/repository-data/NEW_DATA_DIRECTORY
```

4. In `/opt/electriccloud/electriccommander/conf/repository/wrapper.conf`, change `set.default.DATA_DIRECTORY=/opt/electriccloud/electriccommander` to `set.default.DATA_DIRECTORY=/NEW_DATA_DIRECTORY`.

Example: Change `set.default.DATA_DIRECTORY=/opt/electriccloud/electriccommander` to `set.default.DATA_DIRECTORY=/vagrant_data`.

5. Create a `logs/repository/` subdirectory under `NEW_DATA_DIRECTORY`.

Example: Create `/vagrant_data/logs/repository` using the following commands:

```
mkdir /vagrant_data/logs
mkdir /vagrant_data/logs/repository
```

6. Copy `/opt/electriccloud/electriccommander/conf/repository/server.properties` and `/opt/electriccloud/electriccommander/conf/repository/keystore` to the `NEW_DATA_DIRECTORY/conf/repository` directory.

Example: Copy the directories to the `/vagrant_data/conf/repository` using the following commands.

```
mkdir /vagrant_data/conf
mkdir /vagrant_data/conf/repository
cp /opt/electriccloud/electriccommander/conf/repository/server.properties/
vagrant_data/conf/repository
cp /opt/electriccloud/electriccommander/conf/repository/keystore/
vagrant_data/conf/repository
```

7. In `/etc/init.d/commanderRepository`, change

`DATADIR=/opt/electriccloud/electriccommander` to `DATADIR=/NEW_DATA_DIRECTORY`.

Example: Change `DATADIR=/opt/electriccloud/electriccommander` to `DATADIR=/vagrant_data`.

8. Start the repository server:

```
/etc/init.d/commanderRepository start
```

9. Check if ElectricFlow has started using the artifact repository in the new network location:

```
netstat -aon | grep 8200
```

Example:

```
vagrant@commander1:~$ netstat -aon | grep 8200
```

```
tcp        0      0 0.0.0.0:8200          0.0.0.0:*            LISTEN
off (0.00/0/0)
```

This shows that the `DATA_DIRECTORY` repository has moved to `/vagrant_data`.

The `repository.service.log`, `repository.log` and `repository.pid` files will be created in the `/vagrant_data/logs/repository` directory.

10. Publish a new artifact and check if it is published:

```
ectool getArtifactVersions | grep artifactVersionName
```

Moving the Artifact Repository in Windows

In this scenario, the `DATA` directory definition in `C:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\wrapper.conf` or `C:/Program Files/Electric Cloud/ElectricCommander/repository/bin/wrapper-windows-x86-64.exe` (depending on where ElectricFlow is installed) has these settings:

- `set.default.INSTALL_DIRECTORY=C:/Electric Cloud/ElectricCommander`
- `set.default.DATA_DIRECTORY=C:/Electric Cloud/ElectricCommander`

If the Windows server where the current artifact repository is stored is full and you want to move it to a new device with more disk space, map the artifact repository to a new network location.

In this procedure, the `set.default.DATA_DIRECTORY` value will change to the `NEW_DATA_DIRECTORY` value. The `REPOSITORY_BACKING_STORE` value in `C:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\server.properties` will remain the same, relative to the `DATA_DIRECTORY`, which will be `NEW_DATA_DIRECTORY`.

In the example, the `DATA_DIRECTORY` will be changed to `d:\ecdata`, and the `REPOSITORY_BACKING_STORE` value will be relative to this directory.

Note: The `REPOSITORY_BACKING_STORE` value is always relative to the `DATA_DIRECTORY` and cannot be set to a fully qualified absolute path. The solution in <http://ask.electric-cloud.com/questions/2192/how-to-configure-which-directory-the-artifact-repository-uses> will not work until you change the `DATA_DIRECTORY` as described in this procedure.

1. Publish the artifact under the current directory (`C:\ProgramData\Electric Cloud\ElectricCommander\repository-data\<artifact_name>\<version>`):

```
ectool publishArtifactVersion --artifactName <artifact_name> --version <version>
--fromDirectory <directory> --compress 0
```

Example: To publish an artifact under the current directory (C:\ProgramData\Electric Cloud\ElectricCommander\repository-data\jdoe\2001\1.0.0), enter:

```
ectool publishArtifactVersion --artifactName "jdoe:2001" --version "1.0.0"
--fromDirectory d:/temp/artest --compress 0
```

The from Directory (d:/temp/artest) contains the files to be uploaded.

2. Stop the repository server one of these ways:

- Use the Windows service pane.
- If you have admin user permissions, enter **net stop CommanderRepository**.

3. Move the artifact repository data from the current /DATA_DIRECTORY directory to the /NEW_DATA_DIRECTORY directory using one of these methods:

- Use Windows Explorer.
- Enter **move * <destinationDirectory>** where the <destinationDirectory> is the /NEW_DATA_DIRECTORY directory.

Example: To move the data to D:\ECDATA, enter **move * d:\ecdata**.

4. In C:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\wrapper.conf, change set.default.DATA_DIRECTORY=C:/ProgramData/Electric Cloud/ElectricCommander to set.default.DATA_DIRECTORY=/NEW_DATA_DIRECTORY.

Example: Change set.default.DATA_DIRECTORY=C:/ProgramData/Electric Cloud/ElectricCommander to set.default.DATA_DIRECTORY=D:/ECDATA.

5. Create a logs/repository/ subdirectory under the NEW_DATA_DIRECTORY using one of these methods:

- Use Windows Explorer.
- Enter the **mkdir** command as in the following example:

Example: Create D:\ECDATA\logs\repository using the following commands.

```
mkdir D:\ECDATA\logs
mkdir D:\ECDATA\logs\repository
```


6. Copy C:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\server.properties and C:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\keystore to the NEW_DATA_DIRECTORY/conf/repository directory using one of these methods:

- Use Windows Explorer.
- Enter the `mkdir` command as in the following example:

Copy the directories to the /ECDATA/conf/repository using the following commands:

```
mkdir /ECDATA/conf
mkdir /ECDATA/conf/repository
copy C:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\
server.properties D:\ECDATA\conf\repository\
copy C:\ProgramData\Electric Cloud\ElectricCommander\conf\repository\
keystore D:\ECDATA\conf\repository\
```

7. Start the repository server using Windows Service pane.
8. Check if ElectricFlow has started using the artifact repository in the new network location:

```
netstat -aon | find "8200"
```

Example:

```
C:\windows\system32> netstat -aon | find "8200"
```

```
TCP    0.0.0.0:8200          0.0.0.0:0             LISTENING          22868
```

This shows that the DATA_DIRECTORY repository has moved to D:\ECDATA.

The repository service.log, repository.log and repository.pid files will be created in the D:\ECDATA \logs\repository directory.

9. Publish a new artifact and check if it is published:

```
ectool getArtifactVersions | grep artifactVersionName
```

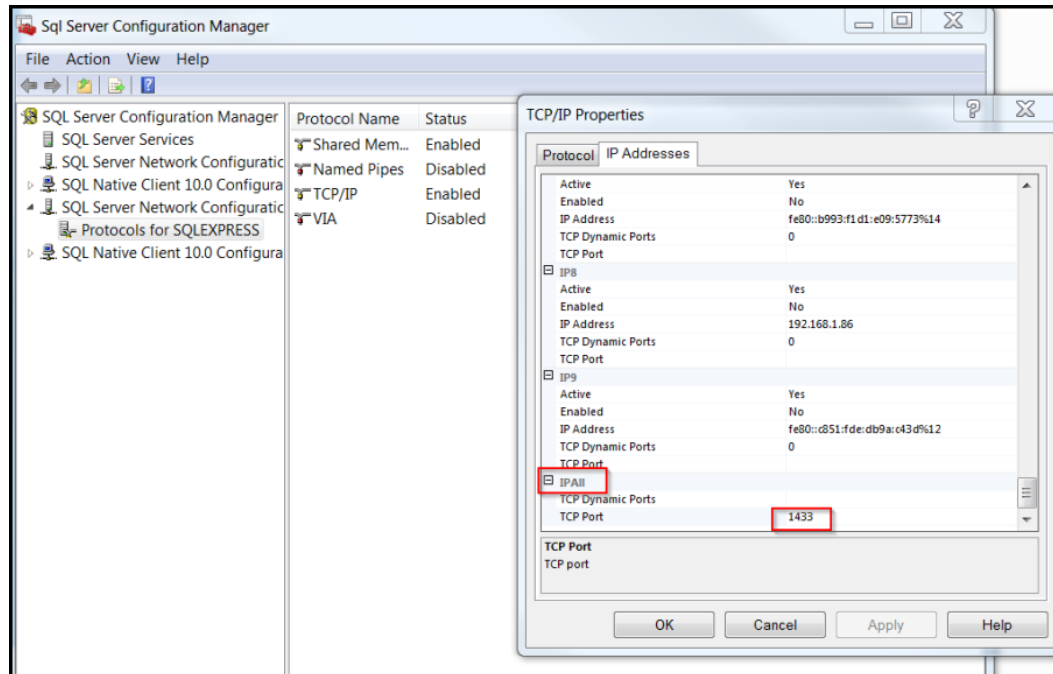
Connecting ElectricFlow to a Microsoft SQL Server Named Instance

If you are using a named instance of the SQL server, you need to configure the SQL server to listen on specific port so that the ElectricFlow server can connect to it.

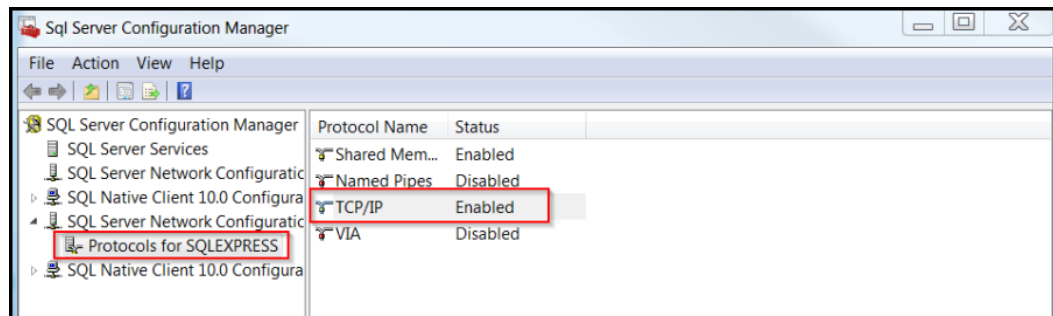
1. In SQL Server Configuration Manager, expand the **SQL Server Network Configuration**, and click on the server instance that you want to configure.
2. In the right pane, double-click **TCP/IP**.
3. In the **TCP/IP Properties** dialog box, click the **IP Addresses** tab.
4. In the IPAll section, enter an available port number in the **TCP Port** field.
For example, enter **1433**.
5. Click **OK** to close the dialog box.
6. When a message appears that the server must be restarted, click **OK**.

7. In the left pane, click **SQL Server Services**.
8. In the right pane, right-click the instance of SQL Server that you selected in Step 1, and click **Restart**.

When the Database Engine restarts, it will listen on port 1433.



9. Enable **TCP/IP**.



Installing the MySQL JDBC Driver

Starting with Commander 4.2.3, ElectricFlow no longer includes the MySQL JDBC driver. If you are upgrading an ElectricFlow server, this change has no effect (the existing driver is saved).

If you plan to perform a clean ElectricFlow server installation that will connect to a MySQL database, you must obtain and install the MySQL JDBC driver.

Follow these steps:

1. Run the installer.

Make sure that you do not install the built-in database.

2. After the installation completes, download the MySQL JDBC driver from <http://dev.mysql.com/downloads/connector/j/>.
3. Install the driver in the `<install-dir>/server/lib` directory.
4. Restart the server.
5. Go to the Database Configuration page and configure the server to use a MySQL database.

Logging Into the ElectricFlow Web Interface

This task describes how to log into the ElectricFlow web interface. If you chose during installation to configure an external database, you will not be able to log into ElectricFlow until you set up a database configuration.

1. Enter the URL of the ElectricFlow server in a browser window. For example, `https://123.123.1.222`

The login screen appears.



IMPORTANT: For a new installation, the default admin account user name is *admin* and the password is *changeme*. You should change the default admin account as soon as possible.

2. Enter a user name and password, and click **Login**.

Chapter 4: Clustering

This section provides you with guidelines and procedures for adding horizontal scalability and high availability to your ElectricFlow environment. Horizontal scalability and high availability is achieved by adding additional machines to an ElectricFlow configuration to create a server cluster. A clustered configuration of ElectricFlow servers also requires these software components:

- A centralized service for maintaining and synchronizing group services in a cluster called Apache ZooKeeper.
- A software or hardware load balancer for routing work to machines in the cluster.

These components typically need to be managed by your IT department.

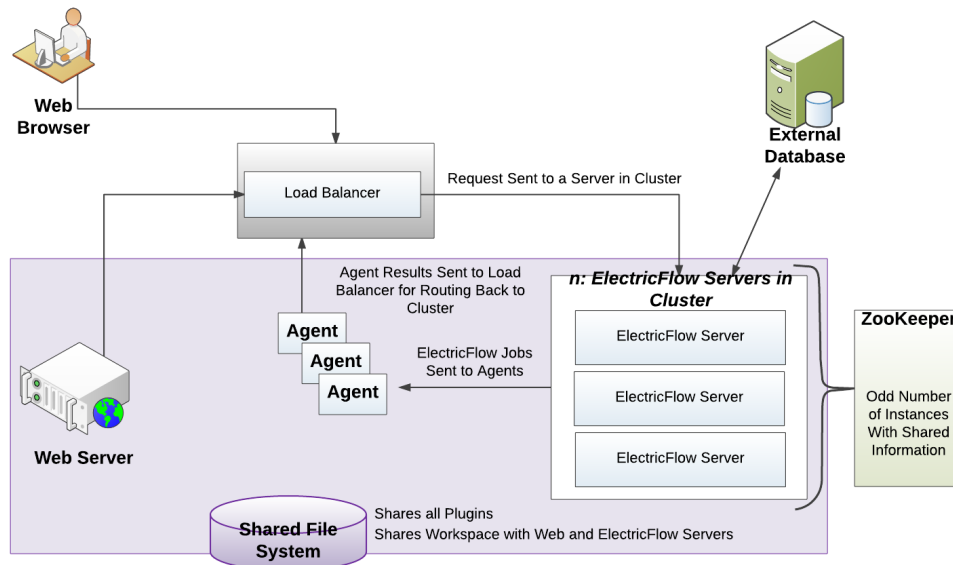
Benefits from Clustering

A clustered ElectricFlow configuration has the following benefits:

- Add fault tolerance by rerouting jobs and API requests to running ElectricFlow servers
- Increase the supported number of simultaneous jobs and API requests
- Distribute API requests across multiple ElectricFlow servers
- Expand capacity over time by adding additional ElectricFlow servers
- Distribute ElectricFlow requests across multiple web servers

Architecture of an ElectricFlow Cluster

The following figure shows an example of the ElectricFlow architecture in a clustered configuration.



In the clustered configuration, a node refers to the machine on which an agent is installed, and a server refers to the ElectricFlow server cluster. In the *server status*, the load balancer reports only status and error messages about a specific server.

As shown in the previous diagram, we strongly recommend that you point the web browser to the load balancer, not to the web server.

Resource, Agent, and Procedure Configuration Considerations

This section describes various ElectricFlow behaviors and possible modifications that you may want to make in your ElectricFlow configuration.

Default Local Resource Use

In a default installation, an agent is installed alongside the ElectricFlow server, and a resource named *local* is automatically created for the agent. In a cluster, the local resource points to only one node. By default, some of the plugins and sample code installed with ElectricFlow automatically use the local resource because it is usually present.

Many ElectricFlow users also write procedures that use the local resource for these reasons:

- A user knows the local resource exists.
- The user needs local access to the file system, such as access to the log files or configuration files.
- The user needs local access to resources on the ElectricFlow server to perform tasks such as checking the CPU, memory, or disk space usage.

IMPORTANT: Any existing procedures for local access to ElectricFlow server file systems or resources need to be modified when you change your installation to run in a server cluster. There are now multiple ElectricFlow server nodes, each with their own log files, configuration files, and local resources.

The name of the ElectricFlow log file on an ElectricFlow server node in a cluster has also been changed to

```
commander-<hostname>.log to facilitate collecting the multiple logs together without them overwriting each other.
```

Unsupported Host

ElectricFlow does not support the local host IP address (127.0.0.1) for any ElectricFlow configuration, because it is ambiguous when multiple servers are used.

Separate Local Agents For Improved Performance

If you are using clustering for performance reasons, you must manage your machine resources efficiently. You use additional machine resources if you run an ElectricFlow agent and ElectricFlow server on the same machine. If you are concerned about performance, remove any agents from your ElectricFlow server node machines. You should also verify that any agents that may have been installed as part of an earlier configuration are also removed. *An agent is installed automatically with any ElectricFlow service by default.*

Pool Local Agents For Improved Reliability

If you are using clustering only for reliability reasons, you must reduce or eliminate single points of failure. In this configuration, having a single server node that runs the local agent is counterproductive. If you need a more reliable configuration, you can install agents on all of your ElectricFlow server node machines and put them in a resource pool named *local*. Local agents should be used for broadcast and maintenance type work. See the "Agent Resource Strategies" section below for more information.

Procedure Strategies

If you have a local agent on each ElectricFlow server, it may be appropriate for some procedures to have one or more procedure steps that are broadcast across all the resources in the local resource pool. These procedure steps are probably followed by a step that aggregates the resulting data in an appropriate way. For other procedures, it may be more appropriate to use the shared file system to which all the ElectricFlow servers have access, and still have the step run on a single agent.

Agent Resource Strategies

If you do not have any agents local to any of your server machines, a local resource is not automatically created. There are two possible strategies to handle this:

- Do not have a local resource and remove any local resource on your system. For everything that is currently configured to use the local resource, reconfigure them to use other resources.
- Create a resource pool called *local* for agents on non-ElectricFlow-server machines. Everything that is configured to use the local resource now runs on an agent of the local pool. However, this agent is no longer local to any of your ElectricFlow server nodes.

If anything relies on the local resource being local to the server, it must be modified to work across multiple server nodes, to function remotely from an agent running on another machine, or to be both. In some configurations, you may need to use the shared file system between the ElectricFlow servers and the remote agent.

For example, if you want to write log-parsing procedures using nonlocal agents, you can configure all of your ElectricFlow servers to write their logs to different locations in this shared file system. A procedure step running on a remote agent machine with access to this shared file system can then parse all the server logs.

Database Restriction

The built-in database is not supported in a clustered ElectricFlow configuration. You must use an alternate database listed in [Supported Alternate Databases](#) on page 2-13: Oracle, MS SQL Server, or MySQL.

broker-data Directory Restriction

The contents of the DATA_DIRECTORY/broker-data directory can never be shared between nodes in an ElectricFlow cluster.

This may occur when a virtual machine running an ElectricFlow server is cloned, and the DATA_DIRECTORY/broker-data directory is also cloned as part of the cloning operation.

If this occurs, remove the DATA_DIRECTORY/broker-data directory from the new virtual machine (VM) as follows:

1. Shut down the ElectricFlow server on the new VM.
2. Delete the DATA_DIRECTORY/broker-data directory on the new VM.
3. Restart the server on the new VM.

Software for Clustering

Apache Zookeeper

Apache ZooKeeper is a centralized service required for clustering.

- [Apache ZooKeeper](#) is a critical part of the clustering architecture. You must use ZooKeeper Version 3.4.6 or later to maintain and synchronize group services in the ElectricFlow cluster. Electric Cloud has also provided a tool called ZKConfigTool, which you can use to quickly populate ZooKeeper with ElectricFlow configuration information.
- [Exhibitor](#) can be used in conjunction with ZooKeeper to monitor the synchronization between the ZooKeeper nodes. This software is not required to implement an ElectricFlow cluster configuration, but can provide instance monitoring, backup, recovery, cleanup and visualization services. For more information, see the [Exhibitor documentation](#).

Load Balancer

You must use a load balancer in an ElectricFlow cluster. You can choose to use any hardware load balancer or load-balancing software for a cluster configuration as long as the load balancer can act as an SSL endpoint and support returning HTTP redirections.

Note: Transport Layer Security (TLS) has replaced Secure Sockets Layer version 3.0 (SSLv3) on the ElectricFlow web server and the ElectricFlow server.

Dependencies for Clustering

A clustered configuration has the following minimum requirements:

- Two or more copies of the ElectricFlow server. Clustering is supported starting with ElectricFlow 5.0.
- Two or more ElectricFlow agents.
- At least one ElectricFlow web server on its own machine, or two or more servers if you are using clustering for reliability improvements.
- An enterprise license. The license is required by ElectricFlow to connect to an external database.
- Apache ZooKeeper as the centralized service for maintaining configuration information.
 - ZooKeeper should be installed on a machine without an ElectricFlow server or load balancer. This separation of services is advised to optimize the performance and reliability of your configuration.

- ZooKeeper must be installed on an odd number of machines. For example, you may need 1, 3, 5, or more instances of the software depending on your environment.
- The ElectricFlow servers must be configured to appear as a single instance in ZooKeeper.
- A hardware load balancer or load-balancing software installed on one machine.
- The web servers, agents, and ElectricFlow servers should share a common file system for plugin information.

Electric Cloud recommends at least one ElectricFlow web server on its own machine or two or more servers if you are using clustering for reliability improvements.

Note: Multiple ElectricFlow clusters can use the same database server, but not the same database schema instance.

Configuring Clustering

There are two different approaches you can take when you configure your ElectricFlow software for horizontal scalability. The approach you choose depends on the needs for your particular ElectricFlow environment for reliability versus performance.

IMPORTANT: Whichever of the following approaches you choose, we recommend that you install multiple ElectricFlow services (agent, web server, ElectricFlow server, and repository) on more than one physical machine (for example, not just virtual machines) to eliminate single points of failure.

- **Reliability** - Choose this type of configuration if your only concern is redundancy for the ElectricFlow application. This approach requires only the addition of multiple ElectricFlow machines to the server cluster. Multiple ElectricFlow services can reside on a machine, but multiple instances of the service software should exist. For example, an ElectricFlow server and agent can reside on the same machine as long as other instances of the components exist on different physical machines.
- **Performance** - Choose this type of configuration if your ElectricFlow cluster will be in a high load environment. This approach requires the installation of the ElectricFlow software services on a sufficient number of dedicated machines. You should install the ElectricFlow server, repository, web server, and agent services on separate machines from the server nodes to avoid competition for system resources.

Note: You can change a reliability configuration to a performance configuration at a later time, but additional configuration of your ElectricFlow software will be required. For more information, see [Separating Agents from ElectricFlow Servers](#) on page 4-26.

Separating Agents from ElectricFlow Servers

Use this procedure if you need to separate ElectricFlow services and agents. By default, an ElectricFlow agent is installed with the ElectricFlow server, web server, and repository. For more information, see [Resource, Agent, and Procedure Configuration Considerations](#) on page 4-2 and [Verifying ElectricFlow Services](#) on page 4-26.

1. Verify that no ElectricFlow agents are installed on any of the ElectricFlow server nodes. If necessary, remove the agent software from the ElectricFlow server nodes.
2. Verify that none of the ElectricFlow utilities use a local resource. If you are not sure if a local resource is in use, create an agent resource called */ocal/* and monitor the system.
3. Remove the local resource.
4. Create a new agent resource with a new name for each agent on each ElectricFlow server node machine.
5. Create a resource pool named */ocal/* containing all these resources.

Preparing Your Cluster Resources

Before you install any ElectricFlow software, you must complete the following tasks:

1. Identify all machines to be used in the horizontally scalable configuration. It is helpful to have all the network information and machine descriptions available before you begin any work so IP addresses can be used consistently throughout a cluster.
 - a. Identify which systems will have a new installation of the ElectricFlow software and which pre-existing systems will be converted to operate in a cluster. Because traffic between the load balancer and the ElectricFlow server nodes is not encrypted, for security reasons all the ElectricFlow server nodes should be located on the same private network as the load balancer, preferably in the same data center.
 - b. Record the IP addresses of:
 - The load balancer machine
 - The machines that will run ZooKeeper
 - The web server machines
 - The ElectricFlow server you will use to import configuration information into ZooKeeper
 - The remaining ElectricFlow server machines that will make up the cluster
 - c. Record the fully qualified domain name of the load balancer machine. This name will be used in several stages of the configuration process, and should be used consistently throughout the process.
2. Install the load balancer on a machine. For more information, see the instructions from the manufacturer.
3. Install ZooKeeper on an odd number of machines. To eliminate a single point of failure, three or five instances of the software are recommended. For more information, go to [Installing ZooKeeper](#) on page 4-7.

Installing and Configuring a Load Balancer

You must use a load balancer in an ElectricFlow cluster. You can choose to use any hardware load balancer or load-balancing software for a cluster configuration as long as the load balancer can act as an SSL endpoint and support returning HTTP redirections.

Note: Transport Layer Security (TLS) has replaced Secure Sockets Layer version 3.0 (SSLv3) on the ElectricFlow web server and the ElectricFlow server.

When configuring your load balancer, follow these general guidelines.

- You must configure a load-balancer IP address for each node in your cluster.
- Load-balance traffic on port 8000 across the ElectricFlow servers on port 8000
- Act as an SSL endpoint for port 8443 and load-balance the traffic on that port across the ElectricFlow servers on port 8000.

- SSL for STOMP:
 - If the load balancer does not act as a SSL endpoint for STOMP port 61613 but instead does SSL pass-through, SSL/TLS bridging or re-encryption, make sure the **Use SSL for Stomp** checkbox (in the **Edit Server Settings** dialog box) is *checked*. This ensures that Commander knows that the STOMP packets are encrypted and will decrypt them.
 - If the load balancer acts as an SSL endpoint (meaning that it does SSL termination) for STOMP port 61613, make sure the **Use SSL for Stomp** checkbox is *unchecked*. This ensures that Commander knows that the load balancer is forwarding STOMP packets unencrypted.
- The load balancer should be able to perform frequent health check HTTP GET requests for a specific URL and take servers temporarily out of rotation if they receive an HTTP status 503 response. For example, for the ElectricFlow server, this URL can be used: `http://<server-host-name>:<server-http-port>/commanderRequest/health`.
- There are no requirements for the state associated with a user session to be replicated across the cluster.

For an example of how to configure a widely used load balancer such as HAProxy, see the [KBEC-00281 - Configuring Load Balancers in ElectricFlow Clusters](#) Knowledge Base article.

You can use the previous example as a model for the load balancer configuration in your system and modify it to meet the system requirements of your particular model of load balancer and system configuration.

Electric Cloud does not support any specific load balancer. For information using about HAProxy, go to the [HAProxy](#) website. For more information about using another load balancer, go to the website for that load balancer.

Note: If you are using HAProxy and are exporting or importing data in a large XML file, a *504 Gateway Timeout* error, also called an *HTTP 504* error, may occur. We recommend that you change the timeout value in the `/etc/haproxy/haproxy.cfg` configuration file from 50 seconds

```
clitimeout 50000
srvtimeout 50000
```

to

10 minutes

```
clitimeout 600000
srvtimeout 600000
```

Installing ZooKeeper

Use Apache Zookeeper Version 3.4.6 or later to maintain and synchronize group services in a clustered ElectricFlow configuration. For more information, go to the [Apache ZooKeeper website](#).

For your convenience, ZooKeeper 3.4.6 is bundled in your ElectricFlow installation here: `<install_dir>/utils`.

To install ZooKeeper:

1. Use either the bundled ZooKeeper or download ZooKeeper from the [ZooKeeper website](#).
2. Extract and install the files into an appropriate location. For example, `/opt/zookeeper-<release_version>`

IMPORTANT: You must install ZooKeeper on an odd number of machines. The number of machines will determine if you install ZooKeeper in standalone mode (for one machine) or in replicated mode (for three or five machines).

3. Create a `zoo.cfg` configuration file for each machine with an instance of ZooKeeper. For example, `zookeeper-<release_version>/conf/zoo.cfg`.

- For standalone mode on a single ZooKeeper machine, the file has these values:

```
tickTime=2000
dataDir=/var/lib/zookeeper
clientPort=2181
```

- For replicated mode across multiple ZooKeeper machines, the `zoo.cfg` file on each server should have these values:

```
tickTime=2000
dataDir=/var/lib/zookeeper
clientPort=2181
initLimit=5
syncLimit=2
server.1=<ZooKeeper_hostname_1>:2888:3888
server.2=<ZooKeeper_hostname_2>:2888:3888
server.3=<ZooKeeper_hostname_3>:2888:3888
server.4=<ZooKeeper_hostname_4>:2888:3888
server.5=<ZooKeeper_hostname_5>:2888:3888
```

Where `<ZooKeeper_hostname_1>` through `<ZooKeeper_hostname_5>` are the hostnames of the servers for the ZooKeeper service.

Note: The file has only three server value lines for a three-ZooKeeper configuration.

Create `/var/lib/zookeeper/myid` files on each ZooKeeper server, with each containing a single ASCII digit: 1 for the first server, 2 for the second server, and so on, corresponding to their `server.<digit>` values in the `zoo.cfg` files

4. Create `/var/lib/zookeeper/myid` files on each ZooKeeper server, with each containing a single ASCII digit as follows:
 - 1 for the first server,
 - 2 for the second server
 - Up to 5, corresponding to the appropriate `server.<digit>` value in the `zoo.cfg` file.

Running ZooKeeper as a Service on Linux

Install ZooKeeper on machines other than those running ElectricFlow. ZooKeeper must not be run on the same machines as those running the ElectricFlow servers.

To run ZooKeeper as a service, follow these steps:

1. For each machine on which you want to install ZooKeeper, navigate to the `zookeeper-wrapper.zip` file in the ElectricFlow `<install_dir>/utils` directory.
2. Copy or move the `zookeeper-wrapper.zip` file to each of the machines that you want to use for ZooKeeper.

3. Extract `zookeeper-wrapper.zip` to a directory of your choice.

For example, `/opt/zookeeper-<release_version>`.

4. Using a text editor, open the `wrapper.conf` file in the extracted `zookeeper-wrapper/conf` directory.

For example, `/opt/zookeeper-<release_version>/zookeeper-wrapper/conf`.

5. Add the path to the ZooKeeper home directory. For example:

```
# Path to unpacked zookeeper
set.default.ZOOKEEPER_HOME=/opt/zookeeper-<release_version>
```

6. Add the path to the JRE. For example:

```
set.default.JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/jre
```

Now you are ready to start ZooKeeper as a service.

7. Run `zookeeper start`

Running ZooKeeper as a Service on Windows

Install ZooKeeper on machines other than those running ElectricFlow. ZooKeeper must not be run on the same machines as those running the ElectricFlow servers.

To run ZooKeeper as a service, follow these steps:

1. For each machine on which you want to install ZooKeeper, navigate to the `zookeeper-wrapper.zip` file in the ElectricFlow `<install_dir>\utils` directory.

2. Copy or move the `zookeeper-wrapper.zip` file to each of the machines that you want to use for ZooKeeper.

3. Extract `zookeeper-wrapper.zip` to a directory of your choice.

For example, `C:\Users\Administrator\zooservice`

4. Using a text editor, open the `wrapper.conf` file located in the extracted `zookeeper-wrapper\conf` directory.

For example, `C:\Users\Administrator\zooservice\zookeeper-wrapper\zookeeper-wrapper\conf`

5. Add the path to the ZooKeeper home directory.

For example:

```
# Path to unpacked zookeeper
set.default.ZOOKEEPER_HOME=C:\Users\Administrator\zooservice\zookeeper-3.4.6
```

Now you are ready to install and start ZooKeeper as a service.

6. Navigate to and click `InstallZooKeeper-NT.bat` to install ZooKeeper as a service.

The file is in the `zookeeper-wrapper\bin` directory.

For example, `C:\Users\Administrator\zooservice\zookeeper-wrapper\zookeeper-wrapper\conf`.

7. Navigate to and click `StartZooKeeper-NT.bat` to start ZooKeeper as a service.

The file is located in the `zookeeper-wrapper\bin` directory.

For example, `C:\Users\Administrator\zooservice\zookeeper-wrapper\zookeeper-wrapper\conf`.

If you choose to use the command-line interface or a script to start the service, enter `ZooKeeperCommand.bat start`.

Ensuring ZooKeeper Can Locate Java

Because ZooKeeper is a Java application, ensure Java is installed and ZooKeeper can locate it.

The default value for `JAVA_HOME` setting (in the `zookeeper-wrapper/conf/wrapper.conf` file) is

```
set.default.JAVA_HOME=/opt/electriccloud/electriccommander/jre
```

If ElectricFlow is not installed or not installed in its default directory, set `JAVA_HOME` in `wrapper.conf` to the location for Java. For example:

```
set.default.JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
```

Verifying that ZooKeeper is Running

To quickly check that the ZooKeeper software is running, follow these steps:

1. Log in to each ZooKeeper machine and run the command:

```
echo ruok | nc 127.0.0.1 2181
```

2. Confirm that you get the following response from each ZooKeeper instance:

```
imok
```

Note: If you get no response, or a 'broken pipe' error, ZooKeeper is not running.

3. Obtain more information about the status of Zookeeper by logging into each ZooKeeper machine and running the command:

```
echo status | nc 127.0.0.1 2181
```

Exhibitor Software

After installing ZooKeeper you might want to install the optional Exhibitor software on every machine with an instance of Zookeeper. The Exhibitor software provides a web interface that allows you to monitor the status of ZooKeeper. It also keeps the configurations of all the ZooKeeper nodes in sync when any of them are changed, and provides tools to rotate and prune the ZooKeeper logs, to prevent them from growing indefinitely. For more information, see the [Exhibitor documentation](#).

If you choose to install Exhibitor, you must configure your ElectricFlow server nodes so they know how to contact Exhibitor. If you have already set up your ElectricFlow server cluster and ZooKeeper servers and are later adding Exhibitor to it, this can be done using the `ecconfigure` tool, which is normally found at `/opt/electriccloud/electriccommander/bin/ecconfigure` on Linux or `C:\Program Files\Electric Cloud\ElectricCommander\bin\ecconfigure.exe` on Windows. Bring down all nodes in your cluster, and run `ecconfigure` on each ElectricFlow node in the cluster with the option (as the user that ElectricFlow runs as, or with administrative privileges):

```
ecconfigure --serverExhibitorConnection <Exhibitor_servers>
```

where `<Exhibitor_servers>` is a comma-separated (no spaces) list of the IP addresses or fully-qualified domain names of your three or five (or for a test system possibly just one) Exhibitor servers (the port number that

ElectricFlow uses to connect to Exhibitor is always port 80). For example, 10.0.2.1,10.0.2.2,10.0.2.3 for a three-ZooKeeper/Exhibitor cluster.

Note: It is not necessary to use the command with a single exhibitor and single ZooKeeper server.

Configuring a Multi-ZooKeeper Cluster

If you plan to use a multi-ZooKeeper cluster, you must configure each ZooKeeper with a unique number from the range 1, 2, 3 for a 3-ZooKeeper cluster, or 1, 2, 3, 4, 5 for a 5-ZooKeeper cluster. You must include this number in the following file:

```
<dataDir>/myid
```

where *<dataDir>* is the path you set in your zoo.cfg file.

For example, you can run these commands:

```
sudo touch /var/lib/zookeeper/myid
sudo -- sh -c 'echo <number> > /var/lib/zookeeper/myid'
```

where *<number>* is the appropriate number between 1 and 3, or 1 and 5.

ZooKeeper Requires a Majority of Nodes to Be Up

ZooKeeper requires a majority of its nodes to be up in order for it to be functional. A majority is:

- 1 of 1
- 2 of 3
- 3 of 5

If a majority of nodes is not up, the expected behavior is a "not currently serving requests" error from ZooKeeper.

Installing ElectricFlow Software

You must install the ElectricFlow software components on all the nodes in your ElectricFlow cluster. Where you install the individual components depends on the type of cluster configuration you need to create. For more information on how to install the ElectricFlow software, see [Installing ElectricFlow](#) on page 3-1.

Use the reliability approach if you want to minimize single points of failure in your ElectricFlow installation; use the performance approach if (in addition to minimizing single points of failure), you want to maximize throughput of your ElectricFlow server at the cost of using more hardware.

In the reliability approach, other ElectricFlow components such as agents, repositories, and web servers are placed on the same machine as a node of the ElectricFlow server cluster; in the performance approach, they are placed on other servers to leave as many resources as possible available for the ElectricFlow server node.

Choose one of the following four installation approaches for your environment:

- [New ElectricFlow Installation for Reliability](#) on page 4-12
- [New ElectricFlow Installation for Performance](#) on page 4-12
- [Converting an Existing ElectricFlow Installation for Reliability](#) on page 4-13
- [Converting an Existing ElectricFlow Installation for Performance](#) on page 4-14

For any of these approaches, when you install agent, repository, and web server services, you can save time by configuring the software to point to a remote server location. You must also register all of these service agents as resources on the ElectricFlow server. For more information, see [Duplicating Repository Contents](#) on page 4-14.

- For a command-line installation, set the option `--remoteServer <load_balancer_FQDN>`. If you are doing an advanced installation on Linux, when prompted for the remote ElectricFlow server, enter the `<load_balancer_FQDN>`.
- For a graphical user interface installation, set the Server Host Name field in the "Remote ElectricFlow server" installer page to `<load_balancer_FQDN>:8000`.
The `load_balancer_FQDN` is the fully qualified domain name of your ElectricFlow server's load balancer machine.

New ElectricFlow Installation for Reliability

The reliability approach allows multiple ElectricFlow services to run on a machine, but multiple instances of the service should exist to prevent single points of failure.

1. Install the ElectricFlow server and agent software on one node in the ElectricFlow cluster.

Note: If you do not already have an ElectricFlow web server that you can temporarily point at this ElectricFlow server node, you might want to also install an ElectricFlow web server that can be used for the following two steps in this section. Before you install the ElectricFlow server and agent software on the remaining nodes in the ElectricFlow cluster, turn off the web server. You turn off the web server on Linux by using the command `/etc/init.d/commanderApache stop`, or on Windows by stopping the service and setting the Startup Type to Manual.

2. Configure ElectricFlow to use an external database. At this time, the ElectricFlow node is in a single-server configuration.

For more information, see [Switching to an Alternate Database](#) on page 11-10.

3. Move the plugins directory on the ElectricFlow server software node to a location on the shared file system.

For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

4. Install the ElectricFlow server and agent software on the remaining nodes in the ElectricFlow cluster.
5. Install the ElectricFlow repository service on one or more machines.
6. Register agents on these machines as resources on the ElectricFlow server.
7. Install the ElectricFlow web server service on one or more machines.

New ElectricFlow Installation for Performance

The performance approach requires separate machines for each ElectricFlow service.

1. Install just the ElectricFlow server software on all the nodes in the ElectricFlow cluster.

For more information, see [Silent Unattended Installation Method](#) on page 3-53.

Note: If you do not already have an ElectricFlow web server that you can temporarily point at this ElectricFlow server node, you may also want to install an ElectricFlow web server that can be used for the following two steps in this section. Before you install the ElectricFlow server and agent software on the remaining nodes in the ElectricFlow cluster, turn off the web server. You turn off the web server on Linux by using the command `/etc/init.d/commanderApache stop`, or on Windows by stopping the service and setting the Startup Type to Manual.

2. Configure one instance of the ElectricFlow server software to use an external database.

At this time, the ElectricFlow node is in a single-server configuration.

For more information, see [Switching to an Alternate Database](#) on page 11-10.

3. Move the plugins directory on the ElectricFlow server software node to a location on the shared file system.

For more information, see [Moving the Plugins Directory to a Pre-Configured Network Location](#) on page 5-10.

4. Install the following software services on one or more individual machines.

Each service should not be installed with any other ElectricFlow software components.

- ElectricFlow agent
- ElectricFlow repository server
- ElectricFlow web server

5. Remove any agents that were automatically installed with the ElectricFlow server.

For more information, see [Separating Agents from ElectricFlow Servers](#) on page 4-26 and [Verifying ElectricFlow Services](#) on page 4-26.

Converting an Existing ElectricFlow Installation for Reliability

Because this is a conversion of an existing ElectricFlow system, one or more machines with the ElectricFlow server, agent, web server, and repository software already exist. The reliability approach allows multiple ElectricFlow services to run on a machine, but multiple instances of the service should exist to prevent single points of failure.

1. Upgrade the existing ElectricFlow software according to the instructions in [Roadmap to Upgrade ElectricFlow on page 6-1](#).

Horizontal scalability is supported starting with ElectricFlow 5.0.

2. Verify that the ElectricFlow software is pointing to an external database.

To verify which database is in use:

- a. Log in to ElectricFlow.
- b. Select **Administration > Database Configuration** to see the current database.

The database connection is successfully configured if you can log into ElectricFlow.

See [Switching to an Alternate Database](#) on page 11-10 if additional configuration is required.

3. Verify that the ElectricFlow software is configured to use a plugins directory located on the shared file system.

For more information, see [Universal Access to the Plugins Directory](#).

4. Install the ElectricFlow server and agent software on the remaining nodes for the ElectricFlow cluster.
5. Install the ElectricFlow repository server on one or more machines.
6. Install the ElectricFlow web server on one or more machines.
7. Register the machine agents as resources on the ElectricFlow server.

Converting an Existing ElectricFlow Installation for Performance

Because this is a conversion of an existing ElectricFlow system, one or more machines with the ElectricFlow server, agent, web server, and repository software already exist. The performance approach requires separate machines for each ElectricFlow service.

1. Upgrade the existing ElectricFlow software according to the instructions in [Roadmap for the Upgrade Process](#).

Horizontal scalability is supported starting with ElectricFlow 5.0.

2. Verify that the ElectricFlow software is pointing to an external database.

To verify which database is in use:

- a. Log in to ElectricFlow.
- b. Select **Administration > Database Configuration** to see the current database.

The database connection is successfully configured if you can log into ElectricFlow.

See [Switching to an Alternate Database](#) on page 11-10 if additional configuration is required.

3. Verify that the ElectricFlow software is configured to use a plugins directory located on the shared file system.

For more information, see [Universal Access to the Plugins Directory](#).

4. Remove any web server or agent software that is installed with the original ElectricFlow machine.

This software will be reinstalled on a separate system.

5. Install the ElectricFlow server software on the nodes for the ElectricFlow cluster.

6. Install the following software services on one or more individual machines.

Each machine should not be installed with any other ElectricFlow software services.

- ElectricFlow agent
- ElectricFlow repository server
- ElectricFlow web server

7. Remove any agents that were automatically installed with the ElectricFlow server, web server, and repository services.

The original ElectricFlow machine as well as the new installations should be checked to verify that the agent software is removed.

For more information, see [Separating Agents from ElectricFlow Servers](#) on page 4-26 and [Verifying ElectricFlow Services](#) on page 4-26

8. Remove any repository server software that is installed with the original ElectricFlow machine after you duplicate the repository server contents.

For more information, see [Installing ElectricFlow Software](#) on page 4-11.

If necessary, install the repository software on additional machines.

Duplicating Repository Contents

To duplicate the contents of an existing repository server into a newly installed repository server:

1. Stop the repository servers.
2. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (`<datadir>/repository-data`) is:

- UNIX - `/opt/electriccloud/electriccommander/repository-data`
- Windows - `C:\ProgramData\Electric Cloud\ElectricCommander\repository-data`

3. Start both repository servers.

Running a Cluster in Clustered Mode

You must configure the machines installed with the ElectricFlow software to operate as a cluster.

To configure them in clustered mode:

1. Share the plugins directory across the ElectricFlow servers, agents, and web servers if you have not done so already. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

2. Configure all the ElectricFlow servers for clustered mode operation, giving them a unique identifier that points to the load balancer, the location of the ZooKeeper servers, and optionally also the Exhibitor servers if you are using Exhibitor.

You must use the `ecconfigure` commands on each ElectricFlow server, web server, and agent on which the software component is installed. You must enter the commands on each server that you add. There are no problems if you rerun the commands on a machine. Use one of these methods:

- Use the `ecconfigure` tool.

- a. Locate the `ecconfigure` tool.

For Linux, it is usually at

```
/opt/electriccloud/electriccommander/bin/ecconfigure.
```

For Windows, it is usually at `C:\Program Files\Electric`

`Cloud\ElectricCommander\bin\ecconfigure.exe`.

- b. Run it with the following options on each ElectricFlow server node (as the user that ElectricFlow runs as, or with administrative privileges):

```
ecconfigure --serverName <load_balancer_FQDN>
--serverZooKeeperConnection <ZooKeeper_servers>
```

- Use Exhibitor and enter:

```
ecconfigure --serverName <load_balancer_FQDN>
--serverZooKeeperConnection <ZooKeeper_servers>
--serverExhibitorConnection <Exhibitor_servers>
```

where

- `<load_balancer_FQDN>` is the fully-qualified domain name of your load balancer machine.
- `<ZooKeeper_servers>` is a comma-separated (no spaces) list of the `IP_address_or_FQDN:port_number` for each of your three or five (or for a test system, possibly just one) ZooKeeper servers (the port number for ZooKeeper is normally 2181).
- If you are using Exhibitor, `<Exhibitor_servers>` is a comma-separated (no spaces) list of the IP addresses or fully-qualified domain names of your three or five (or for a test system possibly just one) Exhibitor servers. The port number that ElectricFlow uses to connect to Exhibitor is always port 80.) For example, `10.0.2.1,10.0.2.2,10.0.2.3` for a three-ZooKeeper/Exhibitor cluster.

Example for a three-ZooKeeper cluster:

```
ecconfigure --serverName machine.company.com --serverZooKeeperConnection
10.0.2.1:2181,10.0.2.2:2181,10.0.2.3:2181
```

These `ecconfigure` commands start the ElectricFlow server nodes.

3. Configure the load balancing software for the ElectricFlow server cluster and the ElectricFlow web servers.
 - a. Stop all ElectricFlow server nodes that you want to cluster.
 - b. Back up the `<data_dir>\conf` directory on all ElectricFlow servers nodes.
 - c. Choose an ElectricFlow server node (usually the first node) from which to copy the `<data_dir>\conf` directory to the other nodes.
 - d. Empty the contents of the `<data_dir>\conf` directory on the other nodes.
 - e. Copy the contents of `<data_dir>\conf` from the first node to `<data_dir>\conf` on the other nodes. This ensures that `<data_dir>\conf` on all clustered ElectricFlow servers is identical.

Running a Cluster in Single-Server Mode

A couple of rarely-used ElectricFlow operations are not supported in clustered mode. If you need to perform any of the following operations, you must restart the server in single-server mode:

- Changing the database configuration
This operation can be done in the web UI using **Administration > Database Configuration** or with `ectool`. After performing this operation, you must rerun `ZKConfigTool` to upload the changed configuration from the local `database.properties` file to ZooKeeper before switching back to clustered mode.
- Doing a full import using `ectool`
- Doing a full export using the safe mode with either of these options:
 - Restart
 - Shutdown

To run an ElectricFlow machine in single-server mode:

1. Identify the machine you need to work on and shut down the other machines in the ElectricFlow cluster.
2. Verify that the `database.properties`, `keystore`, `passkey`, and `commander.properties` configuration files on the machine you are working on are up to date and match those that were uploaded to ZooKeeper.
3. Switch this remaining ElectricFlow machine to single-server mode by running the following command:

```
ecconfigure --serverEnableClusteredMode=0
```

The `commanderAgent` and `commanderServer` services restart.

4. Complete your work on the ElectricFlow machine.
5. If you changed the database configuration, use `ZKConfigTool` to upload the updated `database.properties` file to ZooKeeper.

If you used the `eccert` tool (for added trusted agents, revoked certificates, and so on), use `ZKConfigTool` to upload the updated `keystore` file and/or `conf/security` folder to ZooKeeper.

If you updated the `commander.properties` file, use `ZKConfigTool` to upload it to ZooKeeper.

See [Uploading Configuration Files to ZooKeeper](#) on page 4-18 for more information.

6. Switch this remaining ElectricFlow machine back to clustered mode by running the following command:

```
ecconfigure --serverEnableClusteredMode=1
```

7. Start all the other nodes in the ElectricFlow cluster back up

Adding the Configuration to ZooKeeper

After performing the steps in [Running a Cluster in Clustered Mode](#) on page 4-15, complete the following steps:

1. Confirm that the `<data_dir>\conf` directories on all ElectricFlow servers to be clustered are identical.
2. Use ZKConfigTool to upload configuration files from the existing ElectricFlow server to ZooKeeper by using the steps in [Uploading Configuration Files to ZooKeeper](#) on page 4-18.

Uploading Configuration Files to ZooKeeper

Before starting the ElectricFlow server cluster, you must populate an Apache ZooKeeper server with configuration files that all ElectricFlow server nodes will use in a clustered configuration. This command-line tool imports your ElectricFlow database configuration files into your ZooKeeper server.

The following minimum set of files from the `<data_dir>\conf` directory is imported:

- `database.properties`
- `keystore`
- `passkey`
- `commander.properties`

Prerequisites

- The ElectricFlow server package must be installed.
- The system must be running a version of Java supported by ElectricFlow. Java is automatically installed with ElectricFlow as part of the Tools installation.
- The ZooKeeper software must be installed on the network.

Location

ElectricFlow includes the ZKConfigTool in the following default locations:

- Windows:
`C:\Program Files\Electric Cloud\ElectricCommander\server\bin\zk-config-tool-jar-with-dependencies.jar`
- Linux:
`/opt/electriccloud/electriccommander/server/bin/`
`zk-config-tool-jar-with-dependencies.jar`

Command

ZKConfigTool is best run from the `<ElectricFlow install dir>/conf` directory.

```
$ java -jar zk-config-tool-jar-with-dependencies.jar -<options>
```

Option	Description
<code>-commanderPropertiesFile <path_to_file></code>	Import the ElectricFlow commander.properties file.
<code>-confSecurityFolder <path_to_folder></code>	Import the ElectricFlow Server conf/security folder.
<code>-databasePropertiesFile <path_to_file></code>	Import the ElectricFlow database.properties file.
<code>-help</code>	Show the command help.
<code>-keystoreFile <path_to_file></code>	Import the ElectricFlow keystore file.
<code>-passkeyFile <path_to_file></code>	Import the ElectricFlow passkey file.
<code>-readFile <path_on_zookeeper> <path_to_file></code>	Read the specified file from the ZooKeeper server.
<code>-readFolder <path_on_zookeeper> <path_to_folder></code>	Read the specified folder from the ZooKeeper server.
<code>-writeFile <path_on_zookeeper> <path_to_file></code>	Write the specified file to the ZooKeeper server.
<code>-writeFolder <path_on_zookeeper> <path_to_folder></code>	Write the specified folder to the Zookeeper server.

Import Files

Run `ZKConfigTool` to completely populate the ZooKeeper server with configuration information. The system must have the Tools installation and must be able to communicate with ZooKeeper.

- If you run the command with defined file options, the tool immediately attempts to contact the specified ZooKeeper system and import the specified files.

Example command run from the `<data_dir>\conf` directory:

```
COMMANDER_ZK_CONNECTION=<ZooKeeper_Server_IP>:2181
../jre/bin/java -jar
../server/bin/zk-config-tool-jar-with-dependencies.jar
com.electriccloud.commander.cluster.ZKConfigTool
--databasePropertiesFile database.properties
--keystoreFile keystore
```

```
--passkeyFile passkey
--commanderPropertiesFile commander.properties
```

- If you run the command without defined file options, the tool displays a series of prompts for the required ElectricFlow information. After the correct information is entered, the tool contacts the specified ZooKeeper system to import the specified files.

Example command run from the `<data_dir>\conf` directory:

```
COMMANDER_ZK_CONNECTION=<ZooKeeper_Server_IP>:2181 ../jre/bin/java -jar
../server/bin/zk-config-tool-jar-with-dependencies.jar
com.electriccloud.commander.cluster.ZKConfigTool
```

Next Steps

After you upload the new configuration files:

1. Shut down all running ElectricFlow servers in the cluster.
2. Start one ElectricFlow server in the cluster.
3. Check if the ElectricFlow server is fully running by using the following ectool command:

```
ectool --server localhost --timeout 900 getServerStatus --block 1 --serverSta
teOnly 1
```

This command runs for 900 seconds (15 minutes) or until `getServerStatus` displays either `bootstrap` or `running`.

4. If the output says `bootstrap`, enter the command again until it says `running`.

You can also “tail” the `<data_dir>\logs\commander-<hostname>.log` file to check for errors that could prevent the ElectricFlow server from going to `running` state.

5. After the first ElectricFlow server is in `running` state, start all the other ElectricFlow servers so that they can join the cluster.

After all ElectricFlow servers are in `running` state, you can check the “view” of the cluster.

On Windows platforms, enter:

```
<install_dir>\bin\ectool --server localhost getServerStatus --diagnostics 1 | findstr
"<view> <service_name> <participants>" 2>NULL
```

On Linux platforms, enter:

```
<install_dir>/bin/ectool --server localhost getServerStatus --diagnostics 1 | grep -i
"\(<view>|\<service_name>|\<participants>>)"
```

Getting information on the ElectricFlow Server Cluster from ZooKeeper

Use `ClusterInfoTool` to get information on the running ElectricFlow server cluster from ZooKeeper.

Prerequisites

- The ElectricFlow server cluster must be installed and running on the network.
- Configuration files that all ElectricFlow server nodes will use in a clustered configuration must be uploaded to Apache ZooKeeper server using the ZKConfigTool.
- The ZooKeeper cluster must be running an odd number of Zookeeper nodes and there must be a leader node.
- The system must be running a version of Java supported by ElectricFlow. Java is automatically installed on a system with the ElectricFlow software as part of the Tools installation.

Locations

The ElectricFlow installer adds the ClusterInfoTool to the following default locations:

- **Windows:** C:\Program Files\Electric Cloud\ElectricCommander\server\bin\cluster-info-tool-jar-with-dependencies.jar
- **Linux:** /opt/electriccloud/electriccommander/server/bin/cluster-info-tool-jar-with-dependencies.jar

Command

```
$ java -DCOMMANDER_ZK_CONNECTION=<ZooKeeper_Server1_IP>:2181,<ZooKeeper_Server2_IP>:2181,<ZooKeeper_Server3_IP>:2181 -jar cluster-info-tool-jar-with-dependencies.jar
```

where DCOMMANDER_ZK_CONNECTION must point to the ZooKeeper system to which the ElectricFlow server cluster is connected.

Output

This is sample output generated by ClusterInfoTool:

Checking /commander/jgroups/hornetq:

```
582d3642-9736-e87f-4e19-c22d7776ccab ->
      WIN-M3Q09A2PNFP-21066    6c80e9a9-2fed-6352-e437-fe76b65aa80d    10.0
      .175.78:5446      F
      WIN-M3Q09A2PNFP-28728    0afa39df-d072-b05a-034a-aed74b7a39ee    10.0
      .238.179:5446      F
      WIN-M3Q09A2PNFP-28295    c69f0430-df04-bf24-1294-b471a4a3f151    10.0
      .2.207:5446      F
      WIN-M3Q09A2PNFP-15869    582d3642-9736-e87f-4e19-c22d7776ccab    10.0
      .2.206:5446      T
```

Checking /commander/jgroups/commander:

```
0c32103a-d4e7-da91-f8c4-1fd8e125c156 ->
      WIN-M3Q09A2PNFP-19713    0c32103a-d4e7-da91-f8c4-1fd8e125c156    10.0
      .2.206:5447      T
      WIN-M3Q09A2PNFP-14432    75afe3eb-7a04-d160-c8d0-d64f8ac3c796    10.0
      .175.78:5447      F
      WIN-M3Q09A2PNFP-33530    1cf1795f-e658-3cae-c515-546082bffec9    10.0
      .238.179:5447      F
      WIN-M3Q09A2PNFP-60743    cf8b4201-3fac-71c4-18da-0a885b3e1e61    10.0
      .2.207:5447      F
```

How to interpret ClusterInfoTool output:

- The nodes `/commander/jgroups/hornetq` and `/commander/jgroups/commander` contain information on these JGroups clusters:
 - `commander` for the ElectricFlow server cluster
 - `hornetq` for the HornetQ cluster
- The children nodes under each of the JGroups nodes represent the participating ElectricFlow servers in the cluster. Each child node entry is in this form:

```
<Logical Name>    <UUID>    <IP address>:<port>    T|F
```
- The number of entries in both JGroups nodes should be the same, with matching IP addresses but with different port numbers and distinct logical names and UUIDs. The coordinator node in each JGroups cluster is identified with a 'T' against its entry.

Configuring Web Server Properties

You must update the `httpd.conf` file on each web server in the cluster. The `httpd.conf` file is usually in `apache/conf` on a Linux machine and `ProgramData\Electric Cloud\ElectricCommander\apache\conf` on a Windows machine.

To configure all the web servers for clustered mode operation and give them the name of the load balancer:

1. Locate the `ecconfigure` tool.

On Linux, it is usually at

```
/opt/electriccloud/electriccommander/bin/ecconfigure.
```

On Windows, it is usually at

```
C:\Program Files\Electric Cloud\ElectricCommander\bin\ecconfigure.exe.
```

2. Run the tool with the following option on each web server. You may need to do this as root or with administrative privileges.

```
ecconfigure --webTargetHostName <load_balancer_FQDN>
```

where `<load_balancer_FQDN>` is the fully qualified domain name of your ElectricFlow server's load balancer machine.

Configuring Repository Server Properties

Additional configuration is required for any repository servers that existed before being converted to operate in a cluster, or were not configured to point to a remote server location during installation.

To configure all the repository servers for clustered mode operation and give them the name of the load balancer:

1. Locate the `ecconfigure` tool.

On Linux, it is usually at

```
/opt/electriccloud/electriccommander/bin/ecconfigure.
```

On Windows, it is usually at

```
C:\Program Files\Electric Cloud\ElectricCommander\bin\ecconfigure.exe.
```

2. Run the tool with the following option on each repository server, You may need to do this as root or with administrative privileges.

```
ecconfigure --repositoryTargetHostName <load_balancer_FQDN>
```

where *<load_balancer_FQDN>* is the fully qualified domain name of your ElectricFlow server's load balancer machine.

Configuring ElectricFlow Agents

You must configure ElectricFlow agents to function within a resource pool.

Note: Transport Layer Security (TLS) has replaced Secure Sockets Layer version 3.0 (SSLv3) on the ElectricFlow web server and the ElectricFlow server.

1. Start and log in to ElectricFlow.
2. Go to the **Cloud > Resources** page.
3. Delete any resource named *local*.
4. Create a resource pool named *local*.
5. Create resources for all the machines that had the ElectricFlow agent software installed.
6. Add agent resources to the local resource pool.

Choose the appropriate step for your approach.

- If you are creating a reliability configuration where each ElectricFlow server machine also has an agent installed, the local resource pool should consist of the set of agents local to the ElectricFlow server machines.
 - If you are creating a performance configuration where none of the ElectricFlow server machines have agents installed, the local resource pool (which in this configuration is not actually local to the servers) should contain several agents to handle any work that may be assigned. For example, work may be assigned due to old default resource settings.
7. Verify that the default resource pool contains two or more resources for reliability.

You must create and add resources to this pool if none exist.

8. Go to the **Administration > Server** page.
9. Click on the **Settings** link near the top right to open the Edit Server Settings form.
10. Set the **Server IP address** entry in the form to the fully qualified domain name of your ElectricFlow server's load balancer.

This setting controls how agents contact the ElectricFlow server when they send results from jobs and similar messages.

Configuring the Cluster Workspace

You must edit the default workspace for log files across the ElectricFlow servers, agents, and web servers.

IMPORTANT: As you need and create more workspaces over time, each workspace should be in a shared network location that all machines in the ElectricFlow cluster can access.

To edit the default workspace:

1. Select **Cloud > Workspaces**.
2. Edit the default workspace entry to reference a shared network location.

For more information, see the “Workspaces and Disk Space Management “ topic in the ElectricFlow Help.

Configuring ElectricFlow Repositories

You must configure ElectricFlow repositories.

1. Go to the **Artifacts > Repositories** page.
2. Verify that the repository server URL points to the load balancer machine.

Adding Trusted Agents to Clusters

Perform the following procedures in these situations:

- Every time that you create trusted agents.
- Every time that you revoke the certificates of existing trusted agents.
- To create a gateway to a zone with trusted agents at both ends. You have to do this only a few times.
- When you want all agents in cluster to be trusted.

You can select one of these methods to add a trusted agent to a cluster:

- Before adding the trusted agent, shut down all but one node in the cluster. Go to [Method 1](#) on page 4-25.
- Keep most of the nodes up while adding trusted agents. Go to [Method 2](#) on page 4-26.

Preparing Your Cluster Environment

Perform the following steps one time when configuring the cluster to work with trusted agents. You can skip these steps if you have already done them.

1. Select the server node. This should be the node you used to upload configuration files to ZooKeeper while performing the procedure in [Uploading Configuration Files to ZooKeeper](#) on page 4-18.
2. Confirm that the certificate files from the node you selected and from ZooKeeper match by using the `verifyClusterCertificate.pl` script. For detailed instructions and the script output, see the [KBEC-00342 - Using the verifyClusterCertificate.pl script for Trusted Agents](#) Knowledge Base article.

In the script output, look for "SUCCESS" to confirm that certificate files match.

If this fails, you may be on the wrong node.

3. Replace `conf/security` folder on all nodes with `conf/security` folder from node selected in [Step 1](#).
4. If you want to add a trusted agent to a cluster and it is acceptable to shut down all but one node in the cluster, go to [Method 1](#) on page 4-25.
5. If you want to keep most of the nodes up while adding trusted agents, perform the rest of the steps in this procedure.

6. Upload the `conf/security` folder from the node you selected to ZooKeeper by using the following command:

```
COMMANDER_ZK_CONNECTION=<ZooKeeper_Server_IP>:2181 <INSTALL_
DIRECTORY>/jre/bin/java -cp <INSTALL_DIRECTORY>/server/bin/zk-config-tool-jar-
with-dependencies.jar com.electriccloud.commander.zkconfig.ZKConfigTool -
writeFolder /commander/conf/security <DATA_DIRECTORY>/conf/security
```

7. On *all* cluster nodes:

- a. Open `wrapper.conf` in the `<DATA_DIRECTORY>/conf` directory.
- b. Uncomment the following line:

```
wrapper.java.additional.603=-DCLUSTER_CERTIFICATE_SERVICE_USE_ZOOKEEPER
=true
```

- c. Make sure that `DCLUSTER_CERTIFICATE_SERVICE_USE_ZOOKEEPER` is set to `true`.

Ensure that there is no conflict with number 603 and that it is not already used in the system.

8. Restart the nodes with updated configurations.
9. Go to [Method 2](#) on page 4-26.

Method 1

Follow these steps to add a trusted agent to a cluster by first shutting down all but one node in the cluster. Perform the tasks from [Step 2](#) to [Step 4](#) on the agent machine.

1. Shut down all but one node in the cluster.
2. On the machine with an agent that you want to make a trusted agent, enter commands such as the following to create the trusted agent:

- a. `ectool --server <Server_host> login admin changeme`
to log into the server with the specified hostname or IP address and save the session ID.

- b. To make a remote agent trusted, enter:

```
<INSTALL_DIRECTORY>/bin/eccert initAgent --remote --force
```

to generate a certificate request for this agent, send a certificate authority (CA) request to the ElectricFlow server (the CA), receive a signed certificate from the CA for this agent, and add the CA certificate and the agent's private key (also signed by the CA) to the agent's keystore.

To make a local agent (that is, local to the ElectricFlow server) trusted, enter:

```
<INSTALL_DIRECTORY>/bin/eccert initAgent --remote --force
```

The ElectricFlow server keeps a copy of the signed agent certificate in the `$INSTALL_DIRECTORY/conf/security/certs` directory.

3. On the agent machine, enter

```
ectool createResource <agent_name> --hostName <agent_FQDN_or_IP> --trusted true
```

to add the agent as a trusted agent to the ElectricFlow server in the previous step, where `<agent_FQDN_or_IP>` is the fully-qualified domain name or IP address of the agent.

4. Restart the agent on the agent machine.

Method 2

Perform the following steps on an agent machine to add a trusted agent to a cluster without shutting down server nodes. This procedure works only in ElectricFlow 6.3 or later.

Due to limitations in ZooKeeper, using this method imposes a maximum of around 500 signed certificates. If you want to use this method and are likely to need more than 500 trusted agents, we recommend re-using a certificate across multiple trusted agents.

1. On the machine with an agent that you want to make a trusted agent, enter commands such as the following to create the trusted agent:

- a. `ectool --server <Server_host> login admin changeme`
to log into the server with the specified hostname or IP address and save the session ID.

- b. To make a remote agent trusted, enter:

```
<INSTALL_DIRECTORY>/bin/eccert initAgent --remote --force
```

to generate a certificate request for this agent, send a certificate authority (CA) request to the ElectricFlow server (the CA), receive a signed certificate from the CA for this agent, and add the CA certificate and the agent's private key (also signed by the CA) to the agent's keystore.

To make a local agent (that is, local to the ElectricFlow server) trusted, enter:

```
<INSTALL_DIRECTORY>/bin/eccert initAgent --remote --force
```

The ElectricFlow server keeps a copy of the signed agent certificate in the `$INSTALL_DIRECTORY/conf/security/certs` directory.

2. On the agent machine, enter

```
ectool createResource <agent_name> --hostName <agent_FQDN_or_IP> --trusted true
```

to add the agent as a trusted agent to the ElectricFlow server in the previous step, where `<agent_FQDN_or_IP>` is the fully-qualified domain name or IP address of the agent.

3. Restart the agent on the agent machine.

Separating Agents from ElectricFlow Servers

Use this procedure if you need to separate ElectricFlow services and agents. By default, an ElectricFlow agent is installed with the ElectricFlow server, web server, and repository. For more information, see [Resource, Agent, and Procedure Configuration Considerations](#) on page 4-2 and [Verifying ElectricFlow Services](#) on page 4-26.

1. Verify that no ElectricFlow agents are installed on any of the ElectricFlow server nodes. If necessary, remove the agent software from the ElectricFlow server nodes.
2. Verify that none of the ElectricFlow utilities use a local resource. If you are not sure if a local resource is in use, create an agent resource called `local` and monitor the system.
3. Remove the local resource.
4. Create a new agent resource with a new name for each agent on each ElectricFlow server node machine.
5. Create a resource pool named `local` containing all these resources.

Verifying ElectricFlow Services

You can verify what services are on a machine installed with ElectricFlow software by the following methods:

- If you have a Linux system
 1. Go to the `/etc/init.d/` directory.
 2. Look for scripts starting with "commander". For example, `commanderAgent`, `commanderApache`, `commanderRepository`, `commanderServer`.
 3. As root, from any directory, use this command format:


```
/etc/init.d/<service_name> status
```

 where `<service_name>` is the ElectricFlow service you are interested in, such as


```
/etc/init.d/commanderServer status
```
- If you have a Windows system:
 1. Go to the Services control panel.
 2. Look for services starting with the name "ElectricFlow". For example, ElectricFlow Agent, ElectricFlow Database, or ElectricFlow Server.
 3. If the services have a status of Started, they are installed and running.

Accessing ElectricFlow with Clustering

You access an ElectricFlow server in a clustered configuration the same way you would for a single-server configuration.

Enter the address of one of the web servers into your browser address bar.

If you are using `ectool`, use the `--server` option to direct your request to the fully qualified domain name of the load balancer.

Health Check for the ElectricFlow Cluster

In a clustered configuration, it is important that all ElectricFlow servers that are set up to participate in the cluster can communicate with ZooKeeper and with each other through JGroups. So each ElectricFlow server runs a periodic critical services health check to test that it has a valid session with ZooKeeper and that the server is part of the JGroups cluster for ElectricFlow.

The health check is run every minute by default. If it fails after five repeated attempts, the server goes into bootstrap mode so that it cannot serve any requests. Once in bootstrap mode, the server will periodically attempt to reinitialize its services. If it is successful and is able to join the JGroups cluster, it can serve incoming requests again.

Additional Ways to Improve an ElectricFlow Cluster

Clustering your ElectricFlow configuration does not necessarily remove all single points of failure or potential performance bottlenecks from your system. There are other components that are part of the ElectricFlow environment that you must consider to eliminate single points of failure or prevent performance bottlenecks.

Third-Party Software

The following items are widely-used third-party commercial products that are used in conjunction with ElectricFlow. A variety of solutions and strategies to increase the reliability and scalability of these products and eliminate remaining single-points-of-failure are available from other vendors and sources.

- Network
- Load balancer
- External database
- File server used for the shared file system

ElectricFlow Components

You can address some or all of the following potential issues with the following ElectricFlow components. The issues you address depends on the level of reliability and performance you need for your system.

- Repository server - You can mirror your artifacts across multiple repository servers.
- ElectricFlow procedures or steps - You can specify a resource pool of agents rather than a single agent.

Chapter 5: Configuration

This section contains the configuration tasks you must perform after you install ElectricFlow.

IMPORTANT: The following situation might occur when the workspace files are in a directory other than the default *workspace* directory and the ElectricFlow configuration links to this workspace directory. When you install a new version, ElectricFlow creates a workspace directory in the default location. It does not recognize the preconfigured workspace link in the previous configuration.

When configuring ElectricFlow after upgrading it, you cannot use `ecconfigure` to move the workspace directory to the preconfigured network location. You must manually specify the link to the workspace directory in the new configuration.

Applying an Enterprise License Key

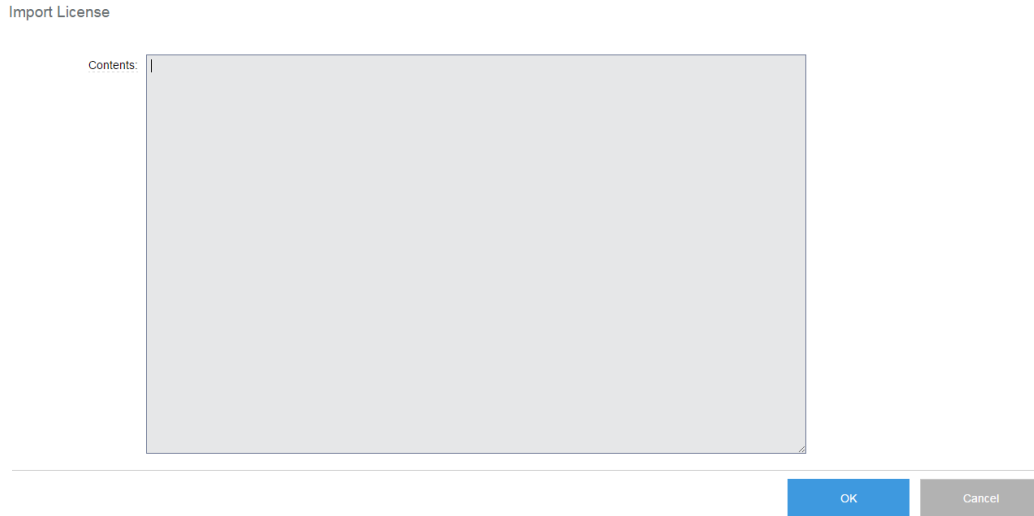
Use the following task to add an enterprise license to an ElectricFlow server.

1. Log into the ElectricFlow server.

For more information, see [Logging Into the ElectricFlow Web Interface](#) on page 3-104.

2. Go to **Administration > Licenses**.
3. Click **Import License**.

The Import License text box appears.



4. Open the license file in a text editor and copy and paste the entire license text in the Import License text box.
5. Click **OK** to import the ElectricFlow license.

External Database Configuration

During the ElectricFlow server installation, if you elected *not* to install the built-in [default] database, you need to configure your alternate database choice. An ElectricFlow enterprise license is required to configure an alternate database. For more information about supported databases, see [Database Requirements](#) on page 2-13. For installation instructions, see [Configuring ElectricFlow to Use an Alternate Database](#) on page 5-3.

Database Interactions

Your database administrator (DBA) must create a database for use specifically by ElectricFlow. The ElectricFlow server interacts with the database using a JDBC driver for each of the databases that ElectricFlow supports.

For MySQL, the JDBC driver is not installed by the ElectricFlow installer (for licensing reasons). It must be downloaded and installed separately from the MySQL website. For more information, go to [Installing the MySQL JDBC Driver](#) on page 3-103.

The first step in any interaction is to present user credentials to the database. This information is stored in the ElectricFlow `database.properties` file as a user name plus a password. The password is stored as an encrypted string, using the “passkey” generated by the server.

Database User

For MS SQL Server and MySQL, your DBA should create a database user for use specifically by ElectricFlow.

For Oracle, your DBA must create a database user for use specifically by ElectricFlow.

The ElectricFlow database user must have permissions to add or delete rows from the database at all times. The database user must also have rights to create or delete tables, and add or remove a columns, indices, and constraints to a table at certain defined times.

IMPORTANT:

Electric Cloud only supports installations where the database user has rights to create and delete tables at all times.

To upgrade the MySQL databases, the database user must also have permissions to create and delete schema (databases) for the entire duration of the upgrade process.

For security reasons, permissions granted to database users to create and delete schema in multi-tenant MySQL databases may be revoked after the upgrade process is completed.

When the ElectricFlow server first starts up, ElectricFlow creates a schema in the specified database so the database user should be the owner of the ElectricFlow database. This will allow the ElectricFlow software to make required schema changes.

If the ElectricFlow server could not connect to the database, it will continue to run waiting for a valid database configuration. Check the log files for a successful database connection.

Recommended MySQL Settings

See the [KBEC-00038 - Improving ElectricCommander server performance by tuning memory allocation](#) Knowledge Base article.

Configuring ElectricFlow to Use an Alternate Database

If you deselected the “database” check box during installation, you will *not* be able to log into ElectricFlow until you set up a database configuration pointing to an external database. An ElectricFlow enterprise license is required, see [External Database Configuration](#) on page 5-2. You can change the database configuration through the ElectricFlow web interface or the ElectricFlow command-line tool.

If you are currently using ElectricFlow in clustered mode and need to change the database configuration, you must reset ElectricFlow to single-server mode before changing the database configuration. For more information, go to [Running a Cluster in Single-Server Mode](#) on page 4-17.

Setting the Database with the Web Interface

Use this procedure to set the database with the ElectricFlow web interface. You will *not* be able to log into ElectricFlow until you set up a database configuration pointing to an external database, but you can use the ElectricFlow web interface to connect to an external database.

1. Go to the Administration tab in the ElectricFlow UI and select Database Configuration.

The Database Configuration screen appears.

The screenshot displays the ElectricFlow web interface's Database Configuration page. At the top, a dark header bar shows the ElectricFlow logo and the user is logged in as 'System Administrator'. Below this is a navigation menu with tabs for Home, Flow Tools, Projects, Jobs, Workflows, Cloud, Artifacts, Search, and Administration. The Administration tab is selected, and within it, the Database Configuration sub-tab is active. The main content area is titled 'Database Configuration' and contains several form fields: 'Database Type' (a dropdown menu set to 'Oracle'), 'Database Name' (text input 'ecloud'), 'Host Name' (text input 'ora-db-test'), 'Port' (text input '1521'), 'Database Credentials' section with 'User Name' (text input 'DB1'), 'Password' (text input), and 'Retype Password' (text input). Each of these fields has a 'Required' label to its right. At the bottom right of the form, there are two buttons: 'Save and Restart Server' (in blue) and 'Cancel' (in gray).

2. Select your **Database Type** from the drop-down menu.
3. Fill-in your **Database Name**.
4. Fill-in the **Host Name** for your database server.
5. Accept the default **Port** or supply the port number you need for your database.
6. Supply the database **User Name** the ElectricFlow server will use to access your database.
7. Fill-in and confirm the **Password** for the database user you specified.
8. Click **Save and Restart Server** after supplying information in all fields.

Setting the Database from a Command Line

This section contains topics related to setting an alternate database for ElectricFlow through a command line.

SQL Server Authentication

SQL Server supports two types of user authentication:

- SQL Server Authentication
- Windows Authentication

You must find out from your DBA which authentication type is required for the ElectricFlow user because, the authentication type influences how information is provided in the ectool command `setDatabaseConfiguration`. See [setDatabaseConfiguration Command Examples](#) on page 5-6 for example command syntax.

Setting the Database with ectool

You use the ectool `setDatabaseConfiguration` command to change the database configuration from the command line.

1. Determine if your database is SQL Server. The type of user authentication used by the database impacts the syntax of the `setDatabaseConfiguration` command that you use in this procedure.

For more information, see these topics:

- [SQL Server Authentication](#) on page 5-4
- [setDatabaseConfiguration Command Examples](#) on page 5-6
- [Setting the Database as a SQL Server with SSO Login on Windows](#) on page 5-6

2. (ElectricFlow 5.0 and later) Set the database configuration based on the type of user authentication used by the database:

- For SQL server authentication or a SQL Server with NTLM login, enter

```
setDatabaseConfiguration <--options>
```

where <--options> are the options that you specify based on the type of user authentication.

For more information, go to [setDatabaseConfiguration Command Options](#) on page 5-8.

- For a SQL server with the SSO login on Windows, see [Setting the Database as a SQL Server with SSO Login on Windows](#) on page 5-6.

After you change the ElectricFlow database configuration, the server attempts to connect to the database to do the initial schema setup.

Note: Do not restart the ElectricFlow server at this time. (You should restart the server only if it was already connected to a built-in or external database and the `ectool setDatabaseConfiguration` command has been used to connect the server to an entirely new database.)

3. Enter the following command and wait for the output:

```
ectool --server localhost --timeout 900 getServerStatus --block 1 --serverStateOnly 1
```

This command runs for 900 seconds (15 minutes), or until Commander finishes creating all the schema objects, or until `getServerStatus` displays either `bootstrap` or `running`.

4. If the output says `bootstrap`, enter the command again until it says `running`.

The `commander.log` file shows `commanderServer` is running as in the following snippet:

```
2016-02-10T19:19:06.582 | 10.0.2.206 | INFO | bootstrap | | | ServerStatus
| commanderServer is running
```

Failing to wait and restarting Commander while it is creating schema objects will cause it to fail to start (during the next manual start) with an error:

```
2016-09-07T15:08:55.825 | DEBUG | bootstrap | | | upgradeData | OperationIn
voker | Exception: InvalidSchema: Unable to validate the database schema: cou
ld not extract ResultSet 2016-09-07T15:08:55.831 | ERROR | bootstrap | | |
| BootstrapCommanderServerImpl | Unable to validate the database schema: cou
ld not extract ResultSet
com.electriccloud.errors.EcException: Unable to validate the database schema:
could not extract ResultSet
at com.electriccloud.errors.EcException.create(EcException.java:165)
at com.electriccloud.errors.EcExceptionBuilder.build(EcExceptionBuilder.java:
34)
at com.electriccloud.upgrade.UpgradeManagerImpl.doDataMaintenance(UpgradeMana
gerImpl.java:672)
at com.electriccloud.upgrade.UpgradeDataOperation.perform(UpgradeDataOperatio
n.java:50)
at com.electriccloud.upgrade.UpgradeDataOperation.perform(UpgradeDataOperatio
n.java:26)
```

5. Use `getServerStatus` to look for problems logging into the database or creating the schema. This command shows log messages from the server bootstrap process.

Note: Before the server is completely up, `getServerStatus` does not require a login session, but after the server is up, it does. Thus, if you enter the `getServerStatus` call and get a “session expired” error, the server is up.

setDatabaseConfiguration Command Examples

Following are examples of how to use the `setDatabaseConfiguration` command.

SQL Server Authentication:

```
ectool setDatabaseConfiguration
--databaseType sqlserver
--databaseName commander
--hostName localhost
--port 1433
--userName commander
--password commander
```

The `--userName` and `--password` options must be included in the `setDatabaseConfiguration` command.

SQL Server with NTLM login:

```
ectool setDatabaseConfiguration
--databaseType sqlserver
--databaseName commander
--hostName localhost
--port 1433
--userName commander@domain.com
--password commander
```

The user name must include the domain name. For example, `user@domain.com` or `domain\user`.

Setting the Database as a SQL Server with SSO Login on Windows

1. Download the JDBC driver from Microsoft.

Select the appropriate JDBC driver version from the Microsoft JDBC driver download page (<https://www.microsoft.com/en-us/download/details.aspx?id=11774>). See the information in the “Details” and “System Requirements” sections to help select the correct driver.

For example, if you download the `sqljdbc_4.0.2206.100_enu.tar.gz` file and unzip it, you get this file:

```
sqljdbc_4.0.2206.100_enu.tar.gz\sqljdbc_4.0.2206.100_enu\sqljdbc_4.0\enu\auth\x64\sqljdbc_
auth.dll
```

2. Check whether you already have the JDBC driver file, such as `sqljdbc4.jar`, in the `C:\Program Files\Electric Cloud\ElectricCommander\server\wars\commander-server.war\WEB-INF\lib\` directory.

This file ships with ElectricFlow and should be version 4.0 or later.

3. Copy the `sqljdbc_auth.dll` file from [Step 1](#) to the `C:/Program Files/Electric Cloud/ElectricCommander/server/lib` directory, which is the same folder set as the `java.library.path` property.

Note: If you are using the 32-bit agent installer, you must copy the `sqljdbc_auth.dll` file in `sqljdbc_4.0.2206.100_enu.tar.gz\sqljdbc_4.0.2206.100_enu.tar\sqljdbc_4.0\enu\auth\x86\` to the `C:/Program Files/Electric Cloud/ElectricCommander/server/lib` directory. If you do not, the following error appears in the `commander.log` file:

```
2015-07-07T18:04:34.737 | WARN | CP-pool -watch -thread -commander | | |
"com.microsoft.sqlserver.jdbc.SQLServerException"
"This driver is not configured for integrated authentication"
```

4. Enter the following command to update the `COMMANDER_CUSTOM_DB_URL` and `COMMANDER_DB_URL` properties in the `C:\ProgramData\Electric Cloud\ElectricCommander\conf\database.properties` file with the JDBC URL specified in the `customDatabaseUrl` argument:

```
ectool setDatabaseConfiguration --databaseType sqlserver --databaseName comma
nder --hostName localhost --port 1433 --customDatabaseUrl "jdbc:sqlserver://l
ocalhost:1433;integratedSecurity=true;databaseName=commander;applicationName=
ElectricCommander;selectMethod=cursor"
```

Note: ElectricFlow uses `COMMANDER_DB_URL` for information only. If `COMMANDER_CUSTOM_DB_URL` is set, this value is used instead of `COMMANDER_DB_URL`.

5. Make sure that the ElectricCommander Server service is set to run as a user who can log into the SQL Server database using Windows authentication.

6. If you needed to set the ElectricCommander Server service to run as a user who can log into the SQL Server database using Windows authentication, restart the ElectricFlow server. Otherwise, check the commander.log file to verify that the ElectricFlow server connects to the SQL Server database with SSO login.

The ElectricFlow server connects to the SQL Server with SSO login. If this is successful, the following lines appear in the commander.log file:

```
USERDNSDOMAIN=ELECTRIC -CLOUD.COM
USERDOMAIN=ELECTRIC -CLOUD
USERNAME=<user who can log into the SQL Server database using Windows authentication>
USERPROFILE=C:\Users\<user who can log into the SQL Server database using Windows authentication>
..
java.library.path=C:/Program Files/Electric Cloud/ElectricCommander/server/lib
...
2015-07-07T18:08:13.255 | INFO | bootstrap | | | 2015-07-07T18:08:13.319 | WARN | bootstrap | | |
...
2015-07-07T18:08:14.479 | INFO | bootstrap | | | 2015-07-07T18:08:14.479 | INFO | bootstrap | | |
```

setDatabaseConfiguration Command Options

The following options are available for use with the ectool command `setDatabaseConfiguration`. The option command syntax is:

```
setDatabaseConfiguration
[--databaseType <mysql|sqlserver|Oracle|builtin>]
[--databaseName <database name>]
[--hostName <host name>]
[--ignorePasskeyMismatch <Boolean flag>]
[--ignoreServerMismatch <Boolean flag>]
[--password <password>]
[--port <port number>]
[--preserveSessions <Boolean flag>]
[--userName <user name>]
[--customDatabaseDialect <custom database dialect>]
[--customDatabaseDriver <custom database driver>]
[--customDatabaseUrl <custom database URL>]
```

The following table describes the command options:

Option	Description
databaseType	Selects the database type—supported options are <code>mysql sqlserver Oracle builtin</code>
databaseName	The name of your alternate database—this is not the host name, but the name the DBA gave the database object.
hostName	The host name where your database is running
ignorePasskeyMismatch	<Boolean flag - 0 1 true false> - If the server is started with a different passkey, ignore the mismatch if “true”. Note: This action discards all saved passwords.
ignoreServerMismatch	<Boolean flag - 0 1 true false> - If the server is started on a different host than where the server previously started, ignore the mismatch if “true”.
port	The port number used by the database Server—if omitted, port 1433 is used
preserveSessions	<Boolean flag - 0 1 true false> - If ignoring a server mismatch, default behavior invalidates all sessions. Setting this flag to “true” saves all sessions, allowing the server to reconnect to running jobs. This option is used in combination with <code>ignoreServerMismatch</code> .
userName	The user name to use when connecting to the database—For MS SQL Server, see the information on the next page regarding support for SQL Server Authentication and Windows Authentication.
password	The password to use to connect to the database—For MS SQL Server, see the information on the next page regarding support for SQL Server Authentication and Windows Authentication.
customDatabaseDialect	Internal option - use only at the request of Electric Cloud support
customDatabaseDriver	Internal option - use only at the request of Electric Cloud support
customDatabaseUrl	Internal option - use only at the request of Electric Cloud support

Enabling Full Database URL Control

If you want to enable full control over the database URL:

1. Go to the `database.properties` file.
2. Create a property called `COMMANDER_CUSTOM_DB_URL`.
3. Add the explicit connect string to that property.

Note: `COMMANDER_DB_URL` is a “write-only” property that it is never read by ElectricFlow. It is generated based on other settings in that file and serves as information only.

If you were using ElectricFlow in clustered mode, you should now upload your modified `database.properties` file to ZooKeeper and then change ElectricFlow back to clustered mode. For more information, go to [Running a Cluster in Clustered Mode](#) on page 4-15.

Universal Access to the Plugins Directory

A plugin is a collection of one or more features, or a third-party integration or tool that can be added to ElectricFlow. The ElectricFlow server installs all plugins into a configurable location named the plugins directory. This directory must be readable by the web server and any agents that need access to the content of one or more plugins. There are two ways to make the plugins directory readable by the web server and any agents. You can configure the ElectricFlow server, agents, and web servers to point to a central network location, or you can replicate the contents of the plugins directory on remote agents and web servers.

Configuring Universal Access for a Network Location

Use these procedures to configure ElectricFlow server, agents, and web servers to point to a universally accessible network location. This is the recommended approach because newly installed plugins are immediately available to all agents and web servers. We strongly recommend that you do this when you are running ElectricFlow in clustered mode, because it allows all the server nodes to share a common plugins directory. You also avoid the overhead of managing multiple plugins directories.

Note: Windows local system accounts cannot access remotely mounted plugin directories.

Ways to Configure Universal Access to Plugins

A network location for the plugins directory can be set up in one of two ways:

- [Moving the Plugins Directory to a Pre-Configured Network Location](#) on page 5-10 - This approach is recommended if you already have a network file system accessible to the ElectricFlow server and all remote agents and web servers.
- [Leaving the Plugins Directory on the ElectricFlow Server](#) on page 5-11 - This approach leaves the plugins directory in the current location on the ElectricFlow server and shares the location across the network. *This approach is only recommended if you do not already have a network location available.*

Moving the Plugins Directory to a Pre-Configured Network Location

Use this task to move the plugins directory to a pre-configured network location. You must have a network file system accessible to the ElectricFlow server and all remote agents and web servers to perform this task.

IMPORTANT: You must have root privileges to use the `--webPluginsDirectory` option.

1. Create an empty directory in the network accessible location.
2. Move the contents of the plugins subdirectory from the ElectricFlow server's data directory to this new directory.
3. Run the following commands on the ElectricFlow server:

- a. `ectool setProperty /server/settings/pluginsDirectory "<PLUGINS>"`

This command gives the ElectricFlow server the location of `pluginsDirectory`.

- b. Run this command *only* if a web server was installed on this machine (by default, during an Express Server installation):

```
ecconfigure --webPluginsDirectory "<PLUGINS>"
```

This command modifies Apache web server configuration files (`ectool` lacks this ability). This command properly configures the Apache web server for the new plugins directory.

- c. Run this command *only* if an agent was installed on this machine (by default during an Express Server install):

```
ecconfigure --agentPluginsDirectory "<PLUGINS>"
```

This command properly configures the ElectricFlow agent for the new plugins directory.

- d. Run this command **ONLY** if you plan on installing remote Windows agents or web servers:

```
ectool setProperty "/server/Electric Cloud/windowsPluginsShare" "<PLUGINS>"
```

- e. Run this command **ONLY** if you plan on installing remote UNIX agents or web servers:

```
ectool setProperty "/server/Electric Cloud/unixPluginsShare" "<PLUGINS>"
```

Where `<PLUGINS>` is either the Windows path to the shared directory for Windows machines, or the UNIX path to the shared directory for UNIX machines.

Note: When installing remote agents or web servers, you will be prompted to enter information about the ElectricFlow server. Select the check box to "discover the plugins directory", and the correct location is automatically picked up from the server and set during installation.

4. Run the following command on remote agents that were already installed:

```
ecconfigure --agentPluginsDirectory "<PLUGINS>"
```

Where `<PLUGINS>` is either the Windows path to the shared directory for Windows machines, or the UNIX path to the shared directory for UNIX machines.

5. Run the following command on remote web servers that were already installed:

```
ecconfigure --webPluginsDirectory "<PLUGINS>"
```

Where `<PLUGINS>` is either the Windows path to the shared directory for Windows machines, or the UNIX path to the shared directory for UNIX machines.

Leaving the Plugins Directory on the ElectricFlow Server

Use this task to leave the plugins in the current location on the ElectricFlow server and share that location across the network so remote agents and web servers can obtain access.

IMPORTANT: This approach is recommended only if you do not already have a network location available to the ElectricFlow server and all remote agents and web servers. See [Moving the Plugins Directory to a Pre-Configured Network Location](#) on page 5-10 if you do have a universally available network location.

IMPORTANT: You must have root privileges to use the `--webPluginsDirectory` option.

1. Choose the appropriate step based on the ElectricFlow server platform:
 - If your ElectricFlow server is a Windows machine, the plugins directory is automatically shared by the name "commander-plugins" during installation. When you install remote agents and web servers on

Windows, they will discover this location and be configured to use it.

- If you are installing remote agents or web servers on UNIX machines, follow these steps:
 - a. Create a Samba mount on a UNIX machine pointing to the plugins share on the Windows machine.
`//<COMMANDER_SERVER_HOST_NAME>/commander-plugins`
 - b. Export the Samba mount as a network file system share on the same UNIX machine used in the previous step.
 - i. Add the following entry to `/etc/exports`:
`/opt/electriccloud/electriccommander/plugins by default): <EXPORT> *
(ro, sync)`
Where `<EXPORT>` is the directory you want to export.
 - ii. Start/restart the NFS server.
 - c. Mount the network file system share to an available directory before installation on UNIX remote agents and web servers. Make sure to mount the share to the same directory across all machines, henceforth referred to as `<UNIX_PLUGINS>`.
 - i. Create `<UNIX_PLUGINS>`.
 - ii. Add the following entry to `/etc/fstab`:
`<HOST>:<EXPORT> <UNIX_PLUGINS> nfs defaults 0 0`
Where `<HOST>` is the host name of the machine on which the directory has been exported and `<EXPORT>` is the directory being exported on that machine.
 - iii. Call: `mount -a`.
 - d. Run the following command on the ElectricFlow server machine:
`ectool setProperty "/server/Electric Cloud/unixPluginsShare" "<UNIX_PLUGINS>"`
- 2. Perform the following steps if your ElectricFlow server is a Linux machine and you are installing remote agents or web servers on other UNIX machines. The plugins directory is not automatically shared as on Windows.
 - a. Export the local plugins directory as a network file system share on the ElectricFlow server machine:
 - i. Add the following entry to
`/etc/exports (/opt/electriccloud/electriccommander/plugins by default):
<EXPORT> * (ro, sync)`
Where `<EXPORT>` is the directory you want to export.
 - ii. Start/restart the NFS server.
 - b. Mount the network file system share to an available directory before installation on UNIX remote agents and web servers. You must mount the share to the same directory across all machines. This share is referred to as `<UNIX_PLUGINS>` in the following steps:
 - i. Create `<UNIX_PLUGINS>`.
 - ii. Add the following entry to `/etc/fstab` (replace `<HOST>` with the host name of the machine on which the directory has been exported and `<EXPORT>` with the directory being exported on that machine):

```
<HOST>:<EXPORT> <UNIX_PLUGINS> nfs defaults 0 0
```

iii. Call: `mount -a`.

c. On the ElectricFlow server machine, run the following command:

```
ectool setProperty "/server/Electric Cloud/unixPluginsShare"
"<UNIX_PLUGINS>"
```

d. If you are installing remote agents or web servers on Windows machines, perform the following steps:

i. Create a Samba share on the ElectricFlow server that is accessible to Windows machines under the name `<WINDOWS_PLUGINS>`.

ii. Run the following command:

```
ectool setProperty "/server/Electric Cloud/windowsPluginsShare"
"<WINDOWS_PLUGINS>"
```

3. Choose the appropriate step to configure remote agents or web servers.

- If you need to install the software for a new remote agent or web server, you must select the **"discover the plugins directory"** option. This will allow the installer to automatically detect and set the correct location from the server.
- If you need to configure remote agents that were previously installed, run the following command:

```
ecconfigure --agentPluginsDirectory "<PLUGINS>"
```

 Where `<PLUGINS>` is the Windows path to the shared directory for Windows machines, or the UNIX path to the shared directory for UNIX machines.
- If you need to configure a remote web servers that were already installed, run the following command:

```
ecconfigure --webPluginsDirectory "<PLUGINS>"
```

 Where `<PLUGINS>` is the Windows path to the shared directory for Windows machines, or the UNIX path to the shared directory for UNIX machines.

Replicating the Plugin Directory on Remote Systems

Use this procedure to keep the Plugins directory in its default server location and replicate the contents to remote agents and web servers.

IMPORTANT: This approach requires you to manage multiple plugin directories. Every time a new plugin is installed on the ElectricFlow server, you must synchronize the changes across all remote copies of the directory. **Only replicate the Plugins directory if you cannot use or configure a central network location.**

- Copy the plugins directory to remote agents and web servers using any file copy mechanism.
 - The copied plugins directories must be readable by only the remote agents and web servers.
 - Plugins should be copied to a plugins subdirectory within the data directory for each remote agent and web server.
 - Every time the ElectricFlow server Plugins directory is updated, you must synchronize the changes across all remote copies.

Environment Proxy Server Configuration

In your environment proxy servers might exist between an intranet and internet. Because proxy servers can inhibit certain types of internet access, you will need to set proxy settings for each impacted machine in your installation.

ElectricFlow servers or web servers can be deployed behind a proxy server.

It is not a common practice to place repository servers or agent systems behind a proxy server since these systems communicate with ElectricFlow through an intranet connection.

Configuring Proxy Settings for Servers

Use `ecconfigure` to set proxy settings for any web server or ElectricFlow server in your configuration that is deployed behind a proxy server. Repository servers are not typically placed behind a proxy server.

1. Select the appropriate perl scripts to run depending on the server type.
 - To set ElectricFlow server proxy settings, enter:

```
ec-perl src/ecconfigure.pl
--serverProxyHost <IP_ADDRESS_PROXY>
--serverProxyPort <PORT>
--serverNoProxyHosts "<HOST1,HOST2>"
```

- To set web server proxy settings, enter:

```
ec-perl src/ecconfigure.pl
--webProxyUrl http://<IP_ADDRESS:PORT>
--webNoProxyHosts <HOST1,HOST2,HOST3>
```

Where:

<IP_ADDRESS_PROXY> is the IP address of the proxy server,

<PORT> is the server port for the proxy server, and

<HOST1,HOST2> is one or more comma separated host names for the servers in the configuration.

2. Restart all the servers where you have applied a proxy setting

IMPORTANT: If you do not restart the servers, the proxy settings will not work.

Testing Server Proxy Settings

Use the following task to verify your proxy server settings.

- Perform the following steps depending on your server type.
 - If you have a web server:
 1. Go to the **Plugin Manager** web page.
 2. Verify the catalog can be viewed and no errors are reported when accessing the catalog URL.
 - If you have an ElectricFlow server:
 1. Go to the **Plugin Manager** web page.
 2. Verify you can install a plugin from the catalog.

Configuring Proxy Agents

Use `ecconfigure` to set proxy settings for any agent system that is deployed behind a proxy server. A proxy server is not usually placed between agents and an ElectricFlow server.

IMPORTANT: When you use a proxy agent, the proxy target *must* run an SSH v2 server. For more information on proxy target agents, see the Configuration and Resource online help topics.

1. Run the following command to set Agent proxy settings:

```
ec-perl src/ecconfigure.pl
--agentProxyHost <IP_ADDRESS_PROXY>
--agentProxyPort <PORT>
--agentNoProxyHosts "<HOST1,HOST2>"
```

Where:

<IP_ADDRESS_PROXY> is the IP address of the proxy server,

<PORT> is the server port for the proxy server, and

<HOST1,HOST2> is one or more comma separated host names for the servers in the configuration.

2. Set the cygwin 1.7 privilege by running the following commands.

Note: Certain commands require administrator privileges to run (for example, `net stop xxx`) using cygwin 1.7 `sshd`. These commands can fail with “access denied” errors. These errors did not occur in cygwin 1.5. The ElectricFlow proxy agent relies on `sshd` being privileged. To set this privilege on cygwin 1.7, you need to run an additional setup script (in addition to `ssh-host-config`).

- a. `cyglisa-config`
- b. `reboot`

Increasing File Descriptors on Linux

A file descriptor is an object a process uses to read or write to an open file and open network sockets (although there are other uses).

Operating systems place limits on the number of file descriptors a process can open. In addition to per-process limits, an OS also has a global limit on the number of file descriptors that all its processes, together, might consume.

A common bottleneck in the default Linux operating system configuration is a lack of file descriptors.

ElectricFlow Server

An ElectricFlow Server uses approximately one file descriptor per running job step and three per uncompleted job.

The following example configures ElectricFlow to use a new limit of 32768:

1. Add the following line to the `init` script for the ElectricFlow Server (in `/etc/init.d/commander`) before the `su -` command:

```
ulimit -n 32768
```

2. Restart the ElectricFlow server:

```
/etc/init.d/commanderServer restart
```

ElectricFlow Agent

An ElectricFlow Agent uses at least two file descriptors per running job step.

It is important to make sure that operating systems on high traffic sites are configured to provide sufficient numbers of file descriptors to ElectricFlow.

The following example describes how to raise the maximum number of file descriptors to 32768 for the ElectricFlow process on the Red Hat Linux distribution:

1. Allow all users to modify their file descriptor limits from an initial value of 1024 up to the maximum permitted value of 32768 by changing `/etc/security/limits.conf`. The following two lines should be part of the file contents:

- `soft nofile 1024`
- `hard nofile 32768`

2. In `/etc/pam.d/login`, add the following line if it does not already exist:

```
session required pam_limits.so
```

3. Configure ElectricFlow to use the new limits. Add the following line to the `init` script for the ElectricFlow Agent (in `/etc/init.d/ecmdrAgent` or `/etc/init.d/commanderAgent`):

```
ulimit -n 32768
```

4. Restart the ElectricFlow agent:

```
/etc/init.d/commanderAgent restart
```

Adjusting Swappiness on Linux

For Java-based machines (ElectricFlow server, repository server, agent), We recommend that you adjust the swappiness kernel parameter to favor applications over disk cache.

- To favor applications 100% over disk cache use:

```
sysctl -w vm.swappiness=0
```

The default of 60 can result in significant delays during garbage collection if any i/o intensive process runs on the machine.

Chapter 6: Roadmap to Upgrade ElectricFlow

IMPORTANT: If your system is running a version earlier than ElectricCommander 4.2, you first need to upgrade to ElectricCommander 4.2.x. You should be able to safely upgrade from ElectricCommander 3.10.x, 4.0.x, or 4.1.x to ElectricCommander 4.2.x, and then upgrade to ElectricFlow 7.x.

Upgrade		File to Download	Run the Migration Tool	Run the Back-grounder Tool	Run the Installer	Go to
From	To					
ElectricCommander 4.2.x	ElectricFlow 7.x	upgrade-bundle.zip	Yes	Yes	Yes	Upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x (Use Cases 1 to 6) on page 7-1
ElectricFlow 5.x	ElectricFlow 6.x, the latest version	ElectricFlow-<version>	No	No	Yes	Use Case 8: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version on page 9-1

Upgrade		File to Download	Run the Migration Tool	Run the Background Tool	Run the Installer	Go to
From	To					
ElectricFlow 5.x for a clustered environment	ElectricFlow 6.x, the latest version, for a clustered environment	ElectricFlow- <version>	No	No	Yes	Use Case 7: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version in a Clustered Environment on page 8-1
ElectricFlow 6.x	ElectricFlow 6.x, the latest (newer) version	ElectricFlow- <version>	No	No	Yes	Use Case 8: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version on page 9-1
ElectricFlow 6.x for a clustered environment	ElectricFlow 6.x, the latest (newer) version, for a clustered environment	ElectricFlow- <version>	No	No	Yes	Use Case 7: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version in a Clustered Environment on page 8-1

IMPORTANT: When upgrading from ElectricCommander 4.2.x to ElectricFlow 6.x, use the migration tool, background upgrader, and installer in the upgrade bundle.

When upgrading from ElectricFlow 5.x to ElectricFlow 6.x, use only the ElectricFlow installer (ElectricFlow-<version>), which works the same way as the installers in previous releases. Do not use the migration tool or the background upgrader.

If you are upgrading from ElectricCommander 4.2.x to ElectricFlow 6.x, your database is large, and you want to minimize the downtime during the upgrade, review the following information:

IMPORTANT: If you are installing on new hardware while performing the upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, be aware that you need to do an additional step when using the background upgrader.

When performing an upgrade on existing hardware, you run the background upgrader against your source database, and then run the installer to install the latest software version. Running the installer will run the background upgrader with the `--final` flag, ensuring that your database is finalized for the upgrade.

However, when installing new hardware (specifically, a new server that is not pointing at the newly upgraded database) while performing the upgrade, the `--final` flag will not run automatically for the background upgrader, and you will need to run it manually.

You need to run the background upgrader one more time with `--final` flag as an option in this command: `commander-upgrade.pl --final`. This will mark the database as being finalized and allow ElectricFlow to work properly.

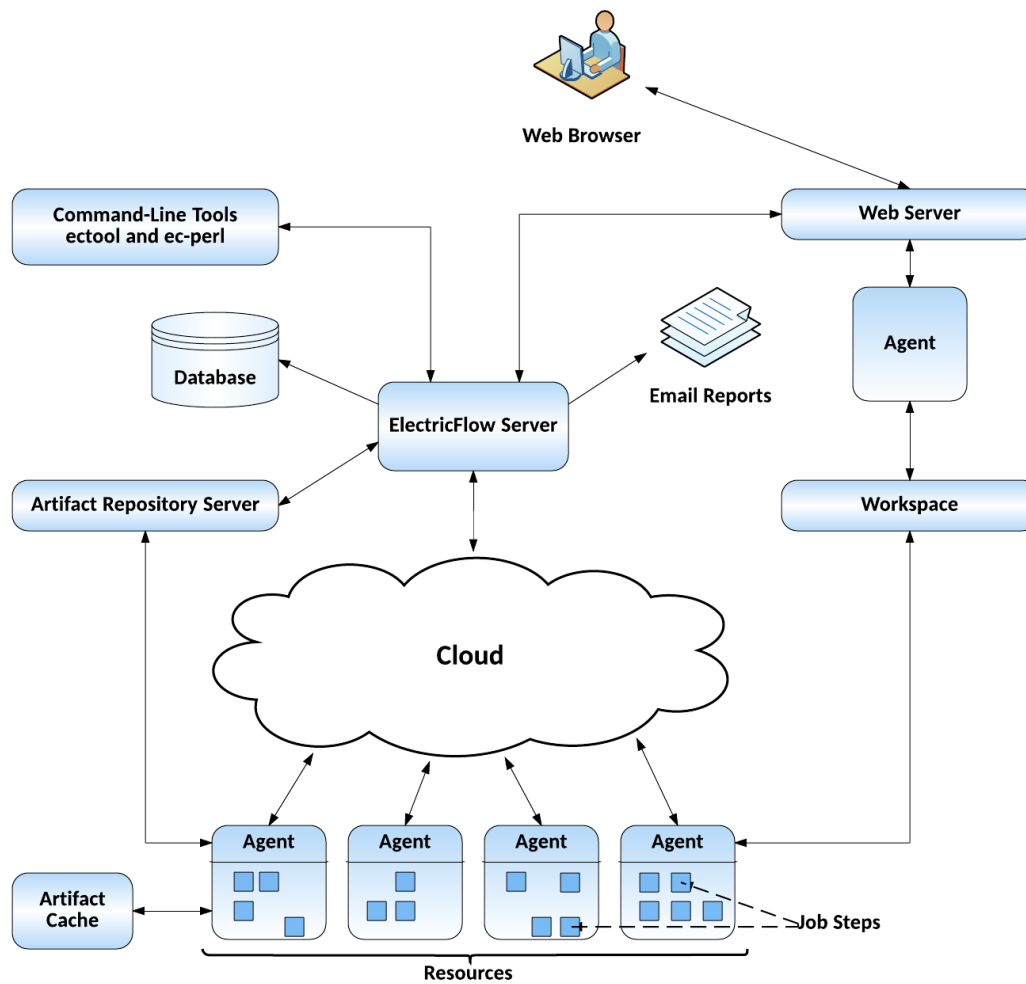
Your order of upgrade operations may be as follows:

1. Run the background upgrader against the ElectricCommander 4.2.x database on the existing hardware.
2. Install the ElectricFlow 7.x software on the new server.
3. Shut down the ElectricCommander 4.2.x server, and run the background upgrader script (`commander-upgrade.pl`) with the `--final` flag on the new hardware.
4. Point the new ElectricFlow 7.x server to the new target (ElectricFlow 7.x) database, either through the UI or using the `ectool setDatabaseConfiguration` command.

If you are running the standard installation (from ElectricFlow 5.x to ElectricFlow 7.x) on an existing ElectricFlow environment, you do not need to manually run the `commander-upgrade.pl` command with the `--final` flag, because this command is run automatically when you execute the installer to upgrade your existing ElectricFlow server to the new version.

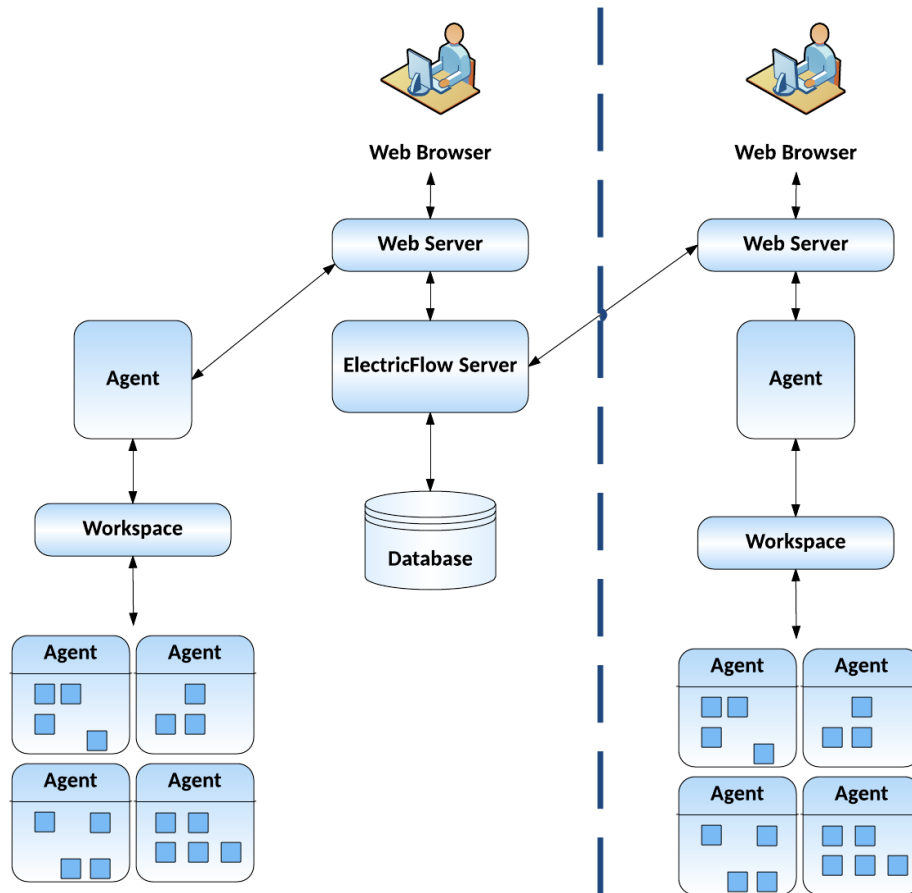
These are sample ElectricFlow configurations:

Single-Site Architecture



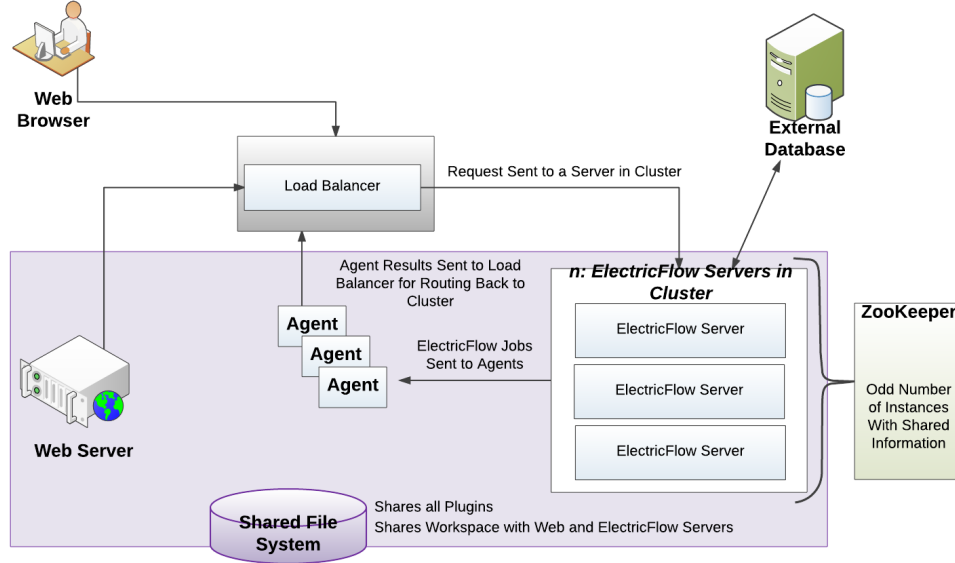
[Architecture](#) on page 1-4 for a complete description of this configuration.

Remote Web Server Configuration



[Architecture](#) on page 1-4 for a detailed description of this architecture.

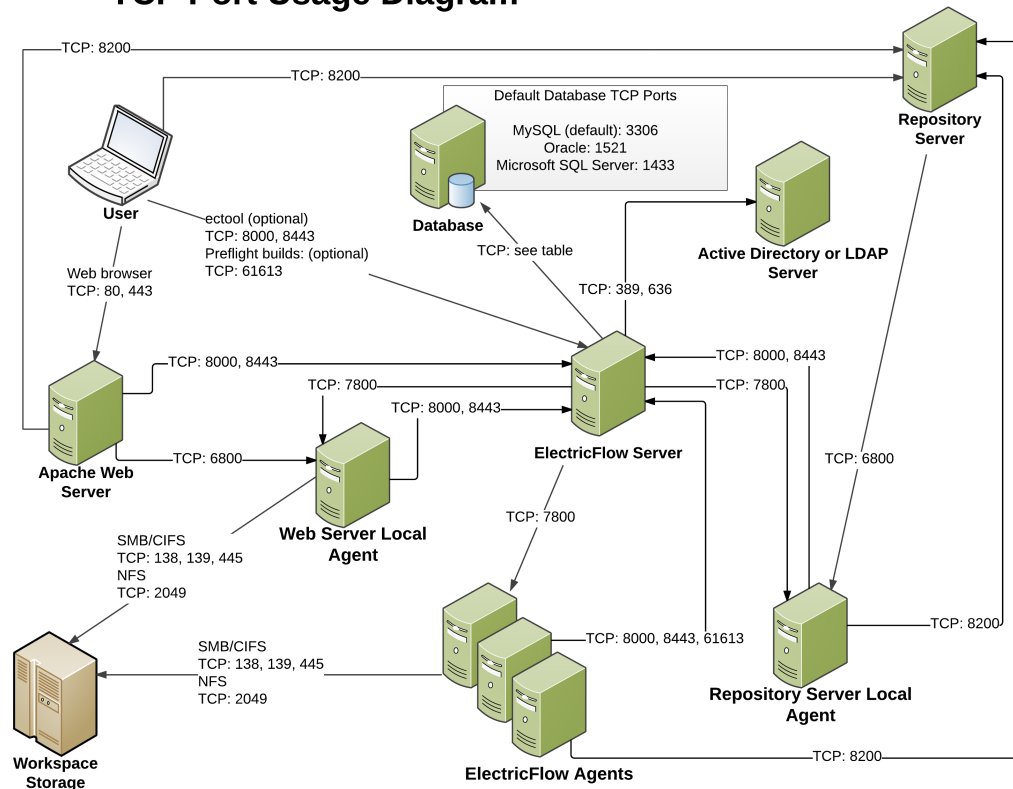
Clustered Configuration



[Architecture of an ElectricFlow Cluster](#) on page 4-1 for a detailed description of this configuration.

Server Components in ElectricFlow

TCP Port Usage Diagram



For a detailed description, go to <https://helpcenter.electric-cloud.com/hc/en-us/sections/200516863-Commander-KB> and search for KBEC-00041.

Chapter 7: Upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x (Use Cases 1 to 6)

Upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x is more complex and extensive than for previous releases.

- The software upgrade requires a database upgrade that is bigger than in previous releases.
- The upgrade process has changed. Starting in ElectricFlow 5.0, the database is modified to use universally unique identifiers (UUIDs) for entity IDs in all tables. This requires recreating all the foreign key relationships in the database, a time-consuming operation.

You use the tools in the upgrade bundle (upgrade-bundle.zip) to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x. The upgrade bundle includes a migration tool to convert `$(jobID)` and `$(workflowId)` references in your projects to `jobCounter` and `workflowCounter` references respectively and a background upgrader to upgrade the database from ElectricCommander 4.2.x to ElectricFlow 7.x.

Depending on your ElectricFlow configuration, you may run both the migration tool and the background upgrader before installing ElectricFlow 7.x.

- Running the ElectricFlow installer is the last step in the upgrade process. The installer works the same way as the installers in previous releases. The `ElectricFlow-<version>` installer, where `<version>` is the software version, collects the ElectricFlow service account credentials, uninstalls the current release, installs the latest ElectricFlow release, configures the system with all property values mined, and restores custom files and data.

IMPORTANT: If your system is running a version earlier than ElectricCommander 4.2, you first need to upgrade to ElectricCommander 4.2.x. You should be able to safely upgrade from ElectricCommander 3.10.x, 4.0.x, or 4.1.x to ElectricCommander 4.2.x, and then upgrade to ElectricFlow 7.x.

IMPORTANT: When upgrading from ElectricCommander 4.2.x to ElectricFlow 6.x, use the migration tool, background upgrader, and installer in the upgrade bundle.

When upgrading from ElectricFlow 5.x to ElectricFlow 6.x, use only the ElectricFlow installer (`ElectricFlow-<version>`), which works the same way as the installers in previous releases. Do not use the migration tool or the background upgrader.

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Upgrade Process Overview

IMPORTANT: If your system is running a version earlier than ElectricCommander 4.2, you first need to upgrade to ElectricCommander 4.2.x. You should be able to safely upgrade from ElectricCommander 3.10.x, 4.0.x, or 4.1.x to ElectricCommander 4.2.x, and then upgrade to ElectricFlow 7.x.

IMPORTANT: You cannot upgrade the ElectricCommander 4.2.x H2 database. If you want to continue using a built-in database in ElectricFlow 7.x, follow the procedures in [Use Case 5: Built-In Database Upgrade on Existing Hardware](#) on page 7-103 or [Use Case 6: Built-In Database Upgrade on New Hardware](#) on page 7-116.

The upgrade process includes these tasks:

- Using the tools in the upgrade bundle to upgrade the software from ElectricCommander 4.2.x to ElectricFlow 7.x.

IMPORTANT: Only use the background upgrader when upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x. Do not use it when you are updating ElectricFlow 5.x to a later ElectricFlow version.

- Running the migration tool and the background upgrader before running the ElectricFlow 7.x installer.
- Running the migration tool to convert `$(jobId)` and `$(workflowId)` to `jobCounter` and `workflowCounter` in the `jobNameTemplate` and `workflowName` template.

- Running the background upgrader to upgrade the database.

The benefits of using the background upgrader are:

- It minimizes downtime during the ElectricFlow software installation.
- When you run the installer without running the background upgrader first, the database is automatically updated as part of the installation process. However, this could take a long time. If your database is large, this could result in a downtime of up to several days.

The database upgrade process includes the following tasks:

- Creating a target (ElectricFlow 7.x) database in the same database instance as the source (original) database.

IMPORTANT: Before you start the database upgrade, you must create a new, empty database that is the same type as the existing source (original) database and is the target (ElectricFlow 7.x) database for the upgrade.

IMPORTANT: You must ensure that the database has sufficient disk space for the upgrade, which includes space for the data copied during the upgrade in addition to space for temporary usage during the upgrade for log files, redo files, and undo files. We recommend that you have at least three times (3x) the size of your current database available during the upgrade process.

- Prepopulating your target (ElectricFlow 7.x) database using the *background upgrader*, which replicates data and upgrades the database while the ElectricCommander 4.2.x server is still running.

While this can be a slow process, if your database is large, it reduces the downtime between turning off your ElectricCommander 4.2.x server and starting your ElectricFlow 7.x server by copying most of the data across while the ElectricCommander 4.2.x server is still running.

If a large number of jobs have run against (or other changes have been made to) the source (original) database between the last pass of the background upgrader and running the ElectricFlow installer, this will increase the downtime required to complete the software upgrade process. This downtime can be minimized by running the background upgrader again to incrementally replicate recent changes from the source (original) database to the target (the ElectricFlow 7.x) database shortly before running the ElectricFlow installer.

IMPORTANT: Change tracking is enabled when you upgrade to ElectricFlow 5.3 and later. This can significantly increase the time (about an hour) to complete the upgrade. If you want to upgrade without enabling change tracking, add this line to the `database.properties` file before starting the software upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

```
COMMANDER_DB_AUDITING_ENABLED=false
```

- Running the background upgrader more than once to ensure that the source (original) and target (ElectricFlow 7.x) databases remain synchronized.
- The database upgrade can be slow. To estimate the time that the upgrade may take, see the "Preparing for the Database Upgrade" step in the appropriate use case.

IMPORTANT: Change tracking is enabled when you upgrade to ElectricFlow 5.3 and later. This can significantly increase the time (about an hour) to complete the upgrade. If you want to upgrade without enabling change tracking, add this line to the `database.properties` file before starting the software upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

```
COMMANDER_DB_AUDITING_ENABLED=false
```

IMPORTANT: If you are installing on new hardware while performing the upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, be aware that you need to do an additional step when using the background upgrader.

When performing an upgrade on existing hardware, you run the background upgrader against your source database, and then run the installer to install the latest software version. Running the installer will run the background upgrader with the `--final` flag, ensuring that your database is finalized for the upgrade.

However, when installing new hardware (specifically, a new server that is not pointing at the newly upgraded database) while performing the upgrade, the `--final` flag will not run automatically for the background upgrader, and you will need to run it manually.

You need to run the background upgrader one more time with `--final` flag as an option in this command: `commander-upgrade.pl --final`. This will mark the database as being finalized and allow ElectricFlow to work properly.

Your order of upgrade operations may be as follows:

1. Run the background upgrader against the ElectricCommander 4.2.x database on the existing hardware.
2. Install the ElectricFlow 7.x software on the new server.
3. Shut down the ElectricCommander 4.2.x server, and run the background upgrader script (`commander-upgrade.pl`) with the `--final` flag on the new hardware.
4. Point the new ElectricFlow 7.x server to the new target (ElectricFlow 7.x) database, either through the UI or using the `ectool setDatabaseConfiguration` command.

If you are running the standard installation (from ElectricFlow 5.x to ElectricFlow 7.x) on an existing ElectricFlow environment, you do not need to manually run the `commander-upgrade.pl` command with the `--final` flag, because this command is run automatically when you execute the installer to upgrade your existing ElectricFlow server to the new version.

Properties Changed to UUIDs

Starting in ElectricFlow 5.0, entity types changed. ElectricFlow now uses a universally unique identifier (UUIDs) for all entity types. The most common entity changes that affect you are the `jobId` and `workflowId` properties. In previous releases, these properties were monotonically incrementing integers that were created directly by the database. The new auto-generated UUID is a 36-character string. This UUID is stored internally in the database as a 128-bit binary type.

For example:

- A new job in ElectricCommander 4.2.x may show 1809 as a `jobId`, but launching this same job in ElectricFlow 7.x may result in a `jobId` like this: 4fa765dd-73f1-11e3-b67e-b0a420524153.

- When you migrate a `jobId` from Commander 4.2.x to ElectricFlow 7.x, ElectricFlow converts the `jobId` to hexadecimal and adds zeroes before creating the 36-character string.

If you migrate a ElectricCommander 4.2.x job with the `jobId` 42978, the `jobId` becomes 00000000-0000-0000-0000-00000000a7e2 in ElectricFlow 7.x.

You most commonly reference `jobId` and `workflowId` in the Job name template and Workflow name template fields in these locations in the platform UI:

- **Administration > Server > Settings**
- Edit procedure or Edit workflow definition

Because the change to UUIDs most commonly occurs in `jobId` and `workflowId` and it appears in job and workflow names, Electric Cloud has provided a tool to help you convert the Job name templates and the Workflow name templates in your projects.

Upgrading to ElectricFlow 7.x

You use the tools in the upgrade bundle to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x. The upgrade bundle has a migration tool that automates and eases the transition to `jobCounter` and `workflowCounter`.

Reasons for the Change

Starting with ElectricFlow 5.0, IDs were changed to UUIDs for several reasons:

- ElectricFlow now allows you to configure a cluster of ElectricFlow servers, which adds horizontal scalability and high availability to your ElectricFlow implementation. Using UUIDs for all primary keys enables multiple ElectricFlow servers to run against the same database without having to coordinate primary key generation across the servers or through the database, which could cause a performance bottleneck.
- In ElectricCommander 4.2.x, the database generated IDs directly. This would create confusion if you moved from one database to another, such as moving from MySQL to Oracle, which would reset the IDs after you uploaded the new database. Using custom counters avoids this ID reset problem and is convenient when moving to a production database such as MS SQL Server or Oracle.
- Some customers were close to running out of 32-bit integer ID values.

Impact for ElectricCommander 4.2.x Users

Dependence on `jobId` or `workflowId`

Many customer implementations have relied on job and workflow names that depend on `$(jobId)` or `$(workflowId)`. Although jobs and workflows with these system-generated identifiers do not fail after upgrading to ElectricFlow 7.x, the readability of this information will become more difficult for users because previously sequential numbers are replaced by 36-character random hexadecimal UUIDs.

Order of Preference for Name Template Values

Whenever a job is run, ElectricFlow names the job according to the job name template value in these locations in descending order of preference:

1. The procedure
2. The property sheet in **Administration > Server > Settings** in the platform UI
3. The server's hard-coded default

Job Name Templates

Upgrading to ElectricFlow 7.x automatically updates ElectricFlow hard-coded default job and workflow name templates. However, you may not see this change in your job or workflow names.

ElectricFlow 7.x *does not automatically update* your name templates if you have ever saved your ElectricFlow server settings for the template default names. It does not matter if you made any changes; the only action that matters is saving the server settings. Saving the server settings creates a custom property sheet, even though it might contain all default values. The creation of the property sheet means job and workflow names are derived from the property sheet's name templates and not the hard-coded name templates. The hard-coded name templates were updated in ElectricFlow 5.0 or later, but because your job and workflow names are not derived from hard-coded name templates, their names are not updated automatically the next time you run a job or workflow. Similarly, your name templates remain as they were in ElectricCommander 4.2.x if they are located in an individual procedure. You need to run the migration tool to update occurrences of `$_[jobId]` or `$_[workflowId]`.

ElectricFlow 7.x *automatically updates* your name templates if you have never saved your ElectricFlow server settings. Because you have never saved the server settings, a custom property sheet does not exist. This means that ElectricFlow uses the default hard-coded name templates, which were updated in ElectricFlow 5.0 or later. Your job or workflow names automatically use the new default name templates the next time you run a job or workflow.

To determine which name templates ElectricFlow is using, follow these steps:

1. In the platform UI, go to **Administration > Server**.
2. Check if more than one property is listed in the Custom Server Properties section.

If multiple properties are listed, you are using a property sheet's templates, not hard-coded templates.

Default Job Name Template Impact

The ElectricCommander 4.2.x default job name template contains the `jobId`, as seen in:

```
job_$_[myJob/jobId]_$_[timestamp]
```

If you are using a property sheet's default job name template and you leave the default job name template as-is, job names for new jobs on the ElectricFlow 7.x server will display with a UUID entry instead of the sequential counter.

If you are using the hard-coded default job name template and you leave the default job name template as-is, your job names will have a job counter instead of the job ID number after you upgrade.

Examples:

Example 1	<p>You are using the property sheet's default job name template in ElectricCommander 4.2.x. This means you do not have a procedure-level job name template. You do not change anything after upgrading to ElectricFlow 7.x.</p> <p>Assume you have previously run 2500 jobs. In ElectricCommander 4.2.x, your next job name might be</p> <pre>job_2501_201401021303</pre> <p>In ElectricFlow 7.x that job name would now look like</p> <pre>job_4fa765dd-73f1-11e3-b67e-b0a420524153_201401021303</pre>
Example 2	<p>You are using the hard-coded default job name template in ElectricCommander 4.2.x. This means you do not have a custom property sheet nor a procedure-level job name template. You do not change anything after upgrading to ElectricFlow 7.x.</p> <p>Assume you have previously run 2500 jobs. In ElectricCommander 4.2.x, your next job name might be</p> <pre>job_2501_201401021303</pre> <p>In ElectricFlow 7.x that job name would now look like</p> <pre>job_2501_20140102130343</pre>

Customized Job Name Template Impact

If you customized your job name template but still used the `jobId`, the UUID now appears instead of the previous `jobId`. This is also true if you used `jobId` in a procedure-level job name template.

We strongly recommend running the migration tool.

Workflow Name Template Impact

The ElectricCommander 4.2.x default workflow name template contains the `workflowId`, as shown in:

```
workflow_${/myWorkflow/workflowId}_${/timestamp}
```

The impact to your workflow names would be the same as what is shown in the job name example.

Actions to Take

Note: This section references `jobId` to `jobCounter` changes, but also consider the same actions for `workflowID` to `workflowCounter` changes.

- We strongly recommend that you convert your naming templates to reference an incrementing job or workflow counter that is tied to the project. You can use the migration tool to help you with the conversion.

Examples of naming templates containing properties that increment automatically:

```
${projectName}_${/increment /myproject/jobCounter}_${/timestamp yyyyMMddHHmmss}
```

```
${projectName}_${/increment /myproject/workflowCounter}_${/timestamp yyyyMMddHHmmss}
```

- If you have any scripts that refer to object IDs and assume that they are numeric, you must update these scripts to handle the IDs as strings.

The migration tool is not able to make the same changes to any locations in your scripts or step commands that reference `jobId` or `workflowId`. You need to manually make the changes to `jobCounter` and `workflowCounter`.

For example, a Perl script with `my $jobId = $[jobId];` should be modified by quoting the job ID as follows: `my $jobId = '$[jobId]';`. This should be done to accommodate the change from a numeric value to an alphanumeric value, the UUID.

Migration Tool in the Upgrade Bundle

Electric Cloud provides the migration tool to ease the transition from `jobId` and `jobCounter`.

This script does the following:

- Examines every procedure in your system and makes the following changes to any `jobNameTemplate` it finds
 - `$[jobId]` becomes `$[/increment /myProject/jobCounter]` or `$[/increment /server/ec_counters/jobCounter]`
 - Within `$[/javascript myJob.jobId]`, `myJob.jobId` becomes `getProperty('/increment /myJob/project/jobCounter')` or `getProperty('/increment /server/ec_counters/jobCounter')`
- Examines every procedure in your system and makes the following changes to any `workflowNameTemplate` it finds
 - `$[workflowId]` becomes `$[/increment /myProject/workflowCounter]` or `$[/increment /server/ec_counters/workflowCounter]`
 - Within `$[/javascript myWorkflow.workflowbId]`, `myWorkflow.workflowbId` becomes `getProperty('/increment /myWorkflow/project/workflowCounter')` or `getProperty('/increment /server/ec_counters/workflowCounter')`
- Determines the current value of the global job ID counter by querying the most recent job. This is used as the starting value for a `jobCounter` property (you can override this value).

By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing the *before* and *after* values). You must run the script with an explicit option to actually make the changes.

You can run this script either before or after you upgrade to ElectricFlow 7.x.

Rather than waiting to upgrade to ElectricFlow 7.x before using `jobCounters`, we recommend that you begin exploring the use of `jobCounters` in your existing production environment. Changing your existing projects to reference a counter rather than `jobId` can be implemented at any time and allows you to start recording your project-level job-count metrics. Applying these changes to your system before upgrading to ElectricFlow 7.x allows you to make these changes at your own pace and avoid concerns being raised by users encountering UUIDs in their job/workflow names after the upgrade.

Default Job Name and Workflow Name Template Changes

Users who upgraded from ElectricCommander 4.2.x may notice that ElectricFlow updated the hard-coded job name and workflow name templates for ElectricFlow 7.x. They no longer contain `jobId` or `workflowId`.

The new default name templates are:

```
$[projectName]_$/increment /myproject/jobCounter]_$/timestamp yyyyMMddHHmmss]
$[projectName]_$/increment /myproject/workflowCounter]_$/timestamp yyyyMMddHHmmss]
```

Changing from a system `jobId` to a project-based `jobCounter` allows you to recognize the number of jobs that each project launches. The previous default of using a system `jobId` provided only a single system-level metric.

Disadvantages of Using `jobId` and `workflowId`

We strongly recommend using `jobCounter` and `workflowCounter` instead of `jobId` and `workflowId`. These are a few reasons to use `jobCounter` and `workflowCounter` instead of `jobId` and `workflowId`:

- The `jobId` and `workflowId` are no longer human-readable integers so they do not provide any identifiable information and cannot be used as counters.
- Because of the increased length of `jobId` and `workflowId`, you might encounter limitations such as maximum Windows path name length, maximum URL length, and maximum command line length.
- The default job name and workflow name templates no longer contain `jobId` and `workflowId`.

jobCounter Considerations

When using the migration tool to define a starting `jobCounter` value, keep in mind the following scenarios and their respective actions:

My job names will keep the same general format, replacing `jobId` with a new `jobCounter`

You must set your starting counter at a number higher than the most recent job that was used for this project.

However, you might find it more useful to use a more recognizable base point to allow you to differentiate jobs that began after making this change

Example:

- The last job run under this project as job 32348.
- Technically, setting your counter to “32349” would avoid any possible job name conflicts.
- However, it would be probably be more effective to set your counter to “40000”, “100000” or “1000000” to help clarify that any new jobs were jobs that were launched after making this change.

My job names will use a completely different format than in the past and will rely on a new `jobCounter`

You might simply want to set your `jobCounter` to “1”.

My job names did not rely on `jobId` previously; they relied on timestamps for unique naming

No change is necessary, but you might want to consider using a `jobCounter` in the future to help capture the relative number of jobs occurring under each project.

My job names already use a project-based `jobCounter` property

No changes are necessary.

If you plan to use Change Tracking on the project, mark the counter property as a counter in ElectricFlow, and consider moving it to a separate counter project.

My job names already use a server-based `jobCounter` property

Although you do not have to change this, be aware that using a project-based property instead might provide slight performance improvements during large bursts of job launches.

Use Cases

The actual steps that you perform to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x are based on your ElectricFlow environment.

Review the information in the following table, and select the use case that best matches your ElectricFlow environment to go to the detailed upgrade process steps.

Use Case	Database Type	Impact on Downtime	Hardware Changes	Link to the Upgrade Process Steps
1	Large	Significant (must minimize the downtime)	Same hardware before and after the upgrade	Use Case 1: Large Database Upgrade on Existing Hardware on page 7-11
2	Large	Significant (must minimize the downtime)	Installing a new ElectricFlow 7.x server	Use Case 2: Large Database Upgrade on New Hardware on page 7-43
3	Small	Not significant	Same hardware before and after the upgrade	Use Case 3: Small Database Upgrade on Existing Hardware on page 7-75
4	Small	Not significant	Installing a new ElectricFlow 7.x server	Use Case 4: Small Database Upgrade on New Hardware on page 7-89 Use Case 4: Small Database Upgrade on New Hardware
5	Built-in	n/a	Same hardware before and after the upgrade	Use Case 5: Built-In Database Upgrade on Existing Hardware on page 7-103
6	Built-in	n/a	Installing a new ElectricFlow 7.x server	Use Case 6: Built-In Database Upgrade on New Hardware on page 7-116

A large ElectricFlow installation can have database upgrades in the size range of 10 GB to several tens of GB.

A small ElectricFlow installation has significantly less data in its database.

Use Case 1: Large Database Upgrade on Existing Hardware

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, the database must be upgraded. The database upgrade process includes creating a new database for ElectricFlow 7.x and copying the data from the source (original ElectricCommander 4.2.x) database to the new target (ElectricFlow 7.x) database.

This process is more involved than in previous releases because the database changes are more extensive than in previous ElectricFlow upgrades. You use the background upgrader in the upgrade bundle to upgrade the database.

IMPORTANT: You cannot upgrade the ElectricCommander 4.2.x H2 database. If you want to continue using a built-in database in ElectricFlow 7.x, follow the procedures in [Use Case 5: Built-In Database Upgrade on Existing Hardware](#) on page 7-103 or [Use Case 6: Built-In Database Upgrade on New Hardware](#) on page 7-116.

IMPORTANT: Before you start the database upgrade, you must create a new, empty database that is the same type as the existing source (original) database and is the target (ElectricFlow 7.x) database for the upgrade.

IMPORTANT: After you have run the ElectricFlow 7.x server, you cannot rerun the background upgrader against the target (ElectricFlow 6.x) database because ElectricFlow 7.x makes further database upgrades.

Step 1.1: Download and Extract the Upgrade Bundle

You use the upgrade bundle (upgrade-bundle.zip) to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x.

The upgrade-bundle.zip file has these files:

- `README.txt`: README file for Windows and Linux
- `migrateTemplates.pl`: Migration tool to convert `$(jobId)` and `$(workflowId)` to `jobCounter` and `workflowCounter` in the `jobNameTemplate` and `workflowName` template.
- `commander-upgrade-jar-with-dependencies.jar`: Background upgrader to update the database. You can run this tool or the background upgrader script before installing the latest ElectricFlow software.
- `commander-upgrade.pl`: Background upgrader script to update the database. You run this script or the background upgrader tool before installing the latest ElectricFlow software.
- `ElectricFlow-<version>`: ElectricFlow installer to install the latest software version.

The Windows installer for ElectricFlow 7.x is `ElectricFlow-<version>.exe`.

The corresponding Linux installer is `ElectricFlow-<version>`.

IMPORTANT: Only use the background upgrader when upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x. Do not use it when you are updating ElectricFlow 5.x to a later ElectricFlow version.

Perform these steps to download and extract the upgrade bundle:

1. Download the upgrade bundle (upgrade-bundle.zip) from the Electric Cloud site.
2. Extract the following files into the `$INSTALL_DIRECTORY` directory:
 - **README file:**
Put `README.txt` into any `$INSTALL_DIRECTORY` directory.
 - **Migration tool for UUIDs:**
Put `migrateTemplates.pl` into the `$INSTALL_DIRECTORY/bin` directory.
 - **Background upgrader:**
Put `commander-upgrade.pl` into the `$INSTALL_DIRECTORY/bin` directory.
Put `commander-upgrade-jar-with-dependencies.jar` into the `$INSTALL_DIRECTORY/Utils` directory.
3. Extract the `ElectricFlow-<version>` installer into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Linux Example:

If the default directory `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/`:

- Put `README.txt` into any `/opt/electriccloud/electriccommander` directory.
- Put `migrateTemplates.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `/opt/electriccloud/electriccommander/Utils` directory.
- Put `ElectricFlow-<version>` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Windows Example:

If the default directory `$INSTALL_DIRECTORY` is `C:\Program Files\Electric Cloud\ElectricCommander\`:

- Put `README.txt` into any `C:\Program Files\Electric Cloud\ElectricCommander` directory.
- Put `migrateTemplates.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `C:\Program Files\Electric Cloud\ElectricCommander\Utils` directory.
- Put `ElectricFlow-<version>.exe` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `C:\temp` directory.

Step 1.2: Run the Migration Tool to Convert `$(jobID)` and `$(workflowId)` References

This section describes how to convert `$(jobID)` and `$(workflowId)` references in your projects to `jobCounter` and `workflowCounter` references respectively. By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing *before* and *after* values). Before you

make the actual changes, make sure you examine the proposed changes thoroughly to avoid job name collision.

IMPORTANT: Before `migrateTemplates.pl` is invoked, the `admin` user must log into `ectool` using the following command and enter the password when prompted:

```
ectool --server xyz login admin
```

1. Go to the `$INSTALL_DIRECTORY/bin` directory.
2. To invoke the migration tool, enter the following command:

```
ec-perl migrateTemplates.pl --server <server name> [options]
```

where `--server <server name>` specifies the ElectricFlow server to migrate. The default server is *localhost*.

The following options are available:

Option	Description
<code>--counters <server project></code>	Specifies whether job counters are server-wide or project-specific. Default: project .
<code>--dryRun <1 0></code>	<ul style="list-style-type: none"> • 1 – Shows the prospective changes in a file that you can examine. Verify the changes before the migration tool makes them. • 0 – Actually makes the changes. Default: 1.
<code>--project <project></code>	Limits migration to the specified project.
<code>--startingJobId <n></code>	Initializes the job counter to the specified number. Default: the current greatest job ID.
<code>--startingWorkflowId <n></code>	Initializes the workflow counter to the specified number. Default: the current greatest workflow ID.

3. View the results.

The migration tool runs successfully when the script completes and output appears showing the changes that were made.

The tool fails when an error message appears.

Example 1

Before `migrateTemplates.pl` is invoked, the `admin` user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter across the entire ElectricFlow system and want the next job number to be 1001:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000
```

The above example will change `$(jobId)` to `$(/increment /server/jobCounter)` and initialize the job counter to 1000. This means the next job run on the server will be number 1001. The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000 --dryRun 0
```

Example 2

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter for each project and want the counters to continue from the current greatest job ID.

```
ec-perl migrateTemplates.pl --server Server1 --counters project
```

The above example will change `$(jobId)` to `$(/increment /myJob/project/jobCounter)` and use the current greatest job ID as the initial value for the job counter.

The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters project --dryRun 0
```

Step 1.3: Prepare for the Database Upgrade

The database upgrade process to ElectricFlow 7.x can be time-consuming. To minimize the duration of this process, the upgrade bundle has a background upgrader to allow your ElectricFlow system to build the ElectricFlow 7.x database while the ElectricCommander 4.2.x server is still running. You run the background upgrader before you run the installer, which can minimize downtime for the ElectricFlow environment and reduce the time required for the upgrade process.

Your DBA should review the following steps to determine which are necessary for your environment to prepare for the database upgrade:

1. Estimate how long the database upgrade may take.

It may take many hours to several days to upgrade the database, depending on the database hardware, database size, and the load on your database server. The background upgrader might put significant extra load on your database server while it runs in the background.

In our internal testing on moderately powerful database servers, database upgrades in the size range of 10 GB to several tens of GB usually took many hours on MySQL or Microsoft SQL Server, and several days on Oracle.

If you have a large database, you can run the background upgrader more than once to synchronize the source (original) and target (ElectricFlow 7.x) databases while the ElectricCommander 4.2.x server is still running.

To estimate the duration of the next database upgrade pass:

- a. Find the log file called `upgrade.log`.

When you run the background upgrader, the log file is in a subdirectory called `/logs` of the current working directory from which the background upgrader was run.

- b. In the log file, search for the phrase "End of pass <number> through tables."

- c. Calculate the duration of a database upgrade pass.

Look for the timestamp when the current upgrade ended and the timestamp when the previous upgrade ended.

The duration is the difference between the time when the last pass ended and the time when the previous pass ended.

The downtime for upgrading ElectricFlow 7.x is expected to take at least as long as the last pass, because it runs one more pass of the background upgrader. On MySQL, it can take many hours more than that, because MySQL requires entire tables to be copied when new columns are added to them.

IMPORTANT: Change tracking is enabled when you upgrade to ElectricFlow 5.3 and later. This can significantly increase the time (about an hour) to complete the upgrade. If you want to upgrade without enabling change tracking, add this line to the `database.properties` file before starting the software upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

```
COMMANDER_DB_AUDITING_ENABLED=false
```

2. Running the background upgrader puts additional load on the database. To minimize both the background upgrader runtime and your upgrade downtime:
 - If you usually run your ElectricFlow database server with your database located on a nonlocal disk such as a network file server, we recommend relocating both your ElectricCommander 4.2.x and new ElectricFlow databases to one or more local disks.
 - If your current database contains a lot of unused free space, we recommend compacting it. If it is fragmented, we recommend defragmenting it or moving it to a solid-state drive (SSD).
3. Estimate the amount of disk space for the database upgrade.

Keep the following in mind:

- You need to make sure that the database server has sufficient disk space for the database upgrade, including disk space for the data copied during the upgrade, log files, redo files, and undo files. The log space depends on the database type and the logging strategy.
 - We recommend that you have at least three times (3x) the size of your current ElectricCommander 4.2.x source (original) database of free disk space.
 - Some of this space is needed temporarily for the upgrade process to run. After the upgrade process is completed, the disk space for the new target (ElectricFlow 7.x) database is only a little larger than before.
4. (Optional) Delete old data in the database.

IMPORTANT: If your ElectricCommander 4.2.x source (original) database contains a lot of old job data, we *strongly* recommend that you delete as much of this data as possible before running the background upgrader, which will help to speed up the eventual upgrade. It will take time for this deletion effort to complete and for the background upgrader to run.

5. Perform the following tasks that are necessary for your environment:

Upgrade Testing

In most implementations, ElectricFlow is being used in an environment that affects many users. We recommend that you test your upgrade on a separate test server to understand all aspects of the upgrade process. This minimizes the potential impacts to downstream users.

We also strongly recommend that you make a copy of your database and walk through the database upgrade process on a test environment before performing the actual database upgrade on your production environment. The result of this is to understand the impact of performing the database upgrade in your environment and to understand the effect that the upgrade has on your environment. If you need a test license to perform such a test, contact support@electric-cloud.com to request a temporary license.

Backing Up Your Existing ElectricFlow Data

If you are upgrading an ElectricFlow server, it is *extremely* important that you back up your existing ElectricFlow data *before* upgrading. For more information about backups, see [ElectricFlow Server Backups](#) on page 11-1.

- Always back up the Plugins Directory. The default location is the `plugins` subdirectory within the data directory.
- Always back up the files that contain your configuration and custom settings. To ensure that all important settings are saved, back up the following subdirectories in `<data_dir>`:
 - The entire `conf` subdirectory (which contains the ElectricFlow server and agent configuration files)
 - Apache web server configuration files in the `apache/conf` subdirectory
 - The local MySQL database configuration file, `my.ini`, in the `mysql` subdirectory (if applicable)
- Always back up any other files where you have created custom configurations, specified other custom information, or created any type of modification.

Note: The number of ElectricFlow files you may have modified is too numerous to list, so you may want to back up the entire data directory and other miscellaneous files that may have changed.

- If you use an artifact repository, back up your ElectricFlow repository configuration files in the `conf/repository` subdirectory.
- Determine if any changes were made to the custom editor or preflight driver script properties (installed by default). Back up those files if changes were made.

These properties are stored in the server-level property sheet, which can be viewed in the platform UI by accessing the **Administration > Server** tab.

Custom editors are stored in the nested sheet named `ec_customEditors`. Preflight driver scripts are stored in the nested sheet named `ec_preflight`. The upgrade process overwrites default custom editor and preflight driver scripts with current versions. We recommend backing up any custom properties you created by renaming the property. For example, change `ec_preflight/clientDrivers/perforce` to `ec_preflight/clientDrivers/perforce_modified`.

Upgrade Installer Preservation

After you back up your ElectricFlow server, create a folder where you can download the ElectricFlow-<version> installation file.

Step 1.4: Run the Background Upgrader

To upgrade the database:

1. Create the new empty target (ElectricFlow 7.x) database with these characteristics:
 - The new database must be the same type as the existing source (original) database and is the target (ElectricFlow 7.x) database for the upgrade.
 - Your source (original) and target (ElectricFlow 7.x) databases must be on the same database server.
 - The text encoding settings of the source (original) and target (ElectricFlow 7.x) databases must match exactly.
- a. For MySQL databases, perform the following additional tasks:
 - You must give ElectricFlow permissions to create new databases and to add and delete tables at all times.
 - You must also give ElectricFlow DROP SCHEMA and CREATE SCHEMA permissions to create and delete databases during the duration of the upgrade process. This includes the intermediate database with the "_upgrade" suffix in its name.

IMPORTANT: If you do not give ElectricFlow the permissions previously listed, the database upgrade will fail with an error similar to the following:

```
2015-07-16T22:22:16.885 | ERROR | bootstrap | | |
schemaMaintenance | ServerStatus | Database upgrade failed:
Migration failed for change set
com/electriccloud/commander/upgrade/upgrade_5062_5063.xml::5062_
5063-0040::commander:
Reason: liquibase.exception.DatabaseException:
com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException:
Access denied for user 'admin'@'%' to database 'commander_
upgrade'
```

- These are the external MySQL database characteristics:
 - When upgrading ElectricFlow, ensure that `default-storage-engine=INNODB` is in the `[mysqld]` section in `my.cnf/my.ini`.
 - An install and upgrade log file named `installer.log` is created in the `logs` subdirectory in the data directory.

- Although the background upgrader does the bulk of the conversion of the MySQL server database from ElectricCommander 4.2.x to ElectricFlow 7.x, the remaining database conversion occurs during the ElectricFlow upgrade. For a large database, this might take many hours.

IMPORTANT: After starting the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, do not interrupt the upgrade process.

If the upgrade process is interrupted, you might corrupt your database. You have to restore your database from a previous backup.

- The following applies when the database is updated:
 - Upgrading saves the existing local MySQL database, if applicable.
 - Use the `ectool` to view the upgrade progress. On a command line, enter `ectool getServerStatus`.

b. For Oracle databases, do the following tasks:

- A database schema is a *user* in Oracle. One user has one schema.
- You must have a new user for the new empty target (ElectricFlow 7.x) database tables.
- We recommend putting the new user tables in their own workspace because this can make managing disk usage easier.
- Ensure that you have allocated adequate table space, temp space, and undo space.
- You must explicitly grant read access permissions to the target (ElectricFlow 7.x) database user so that this user can read information in the source (original) database user's tables.

Example: Enter the following SQL commands.

To specify the source (original) database, enter `GRANT "CONNECT" TO "COMMANDER42"` where `COMMANDER42` is the source (original) database schema or user.

To specify the destination (ElectricFlow 7.x) database, enter `GRANT "RESOURCE" TO "FLOW60"` where `FLOW60` is the destination (ElectricFlow 6.x) database schema or user.

To give the destination (ElectricFlow 7.x) database read access permissions to each table in the source (original) database, enter `GRANT SELECT ON "COMMANDER42"."EC_AGENT" TO "FLOW60"` where `COMMANDER42` is the source (original) database schema or user, `EC_AGENT` is a table in the source (original) database, and `FLOW60` is the destination (ElectricFlow 6.x) database schema or user.

Enter a command like this for each table in the source (original) database.

For a detailed example, see http://docs.electric-cloud.com/eflow_doc/5_4/PDF/grant_access.sql.

- The target (ElectricFlow 7.x) database user must have the same password as the source (original) database user.
- Ensure that the database `UNDO_RETENTION` parameter is set large enough (at least 12 to 24 hours).

2. Create or edit an upgrade.properties file.

The upgrade.properties file provides information to the database upgrade process about the target (ElectricFlow 7.x) and source (original) databases.

- In Linux, the file is usually in
`/opt/electriccloud/electriccommander/conf/upgrade.properties`
- In Windows, the file is usually in
`C:\ProgramData\Electric Cloud\ElectricCommander\conf\upgrade.properties`

Create or edit the file one of these ways:

- a. Use the wizard to help you edit the file.

- b. Create or edit a new `upgrade.properties` file with the `upgrade.properties` file template.
 - i. Copy the template content corresponding to your target (ElectricFlow 7.x) database type.
 - ii. Open a new empty file in a text editor.
 - iii. Paste the entire template contents into the new file.
 - iv. Replace the template values (after the = symbols) with the values from your existing ElectricFlow deployment.

Except for the `COMMANDER_DB_PASSWORD`, `COMMANDER_DB_BATCH_SIZE`, `COMMANDER_DB_NAME` (MySQL or Microsoft SQL Server), and `COMMANDER_DB_USER` (Oracle) values, use your existing database value in one of these locations:

- **Administration > Database Configuration** tab in your existing ElectricFlow server
- `$INSTALL_DIRECTORY/conf/database.properties` file in your existing ElectricFlow deployment for Linux (for example, `/opt/electriccloud/electriccommander/conf/database.properties`)
- `$INSTALL_DIRECTORY\conf\database.properties` file in your existing ElectricFlow deployment for Windows (for example, `C:\ProgramData\Electric Cloud\ElectricCommander\conf\database.properties`)

Make sure that you maintain the formatting of the values in the `database.properties` file.

- v. Save the file as `upgrade.properties`.
- vi. Because the `upgrade.properties` file contains the plaintext password for your ElectricFlow database, restrict read access to the file, or make sure that it is deleted when you no longer need it.

MySQL

```
# Example configuration file for mysql
COMMANDER_DB_TYPE=mysql

# The ip or dns of the mysql server
COMMANDER_DB_HOST=myDbHost.example.com

# The mysql port (defaults to 3306)
COMMANDER_DB_PORT=3306

# The name of the target (Commander 6.x) database
COMMANDER_DB_NAME=commander50

# The mysql username
COMMANDER_DB_USER=mySqlUser

# The mysql password
COMMANDER_DB_PASSWORD=mySqlPassword

# The mysql batch size
COMMANDER_DB_BATCH_SIZE=100
```

```
# The existing 4.2 database.  
# Must be on the same server as the new  
# database and readable by the mySqlUser  
OLD_COMMANDER_DB_NAME=commander42
```

Microsoft SQL Server

```
# Example configuration file for sqlserver  
COMMANDER_DB_TYPE=sqlserver  
  
# The ip or dns of the sqlserver server  
COMMANDER_DB_HOST=myDbHost.example.com  
  
# The sqlserver port (defaults to 1433)  
COMMANDER_DB_PORT=1433  
  
# The name of the target (Commander 6.x) database  
COMMANDER_DB_NAME=commander50  
  
# The sqlserver username  
COMMANDER_DB_USER=sqlServerUser  
  
# The sqlserver password  
COMMANDER_DB_PASSWORD=sqlServerPassword  
  
# The sqlserver batch size  
COMMANDER_DB_BATCH_SIZE=100  
  
# The existing 4.2 database.  
# Must be on the same server as the new  
# database and readable by the sqlServerUser  
OLD_COMMANDER_DB_NAME=commander42
```

Oracle

```
# Example configuration file for oracle  
COMMANDER_DB_TYPE=oracle  
  
# The ip or dns of the TNS listener  
COMMANDER_DB_HOST=myDbHost.example.com  
  
# The TNS listener port (defaults to 1521)  
COMMANDER_DB_PORT=1521  
  
# The sid of database  
COMMANDER_DB_NAME=commanderSID  
  
# The target (Commander 5.x) username/database  
COMMANDER_DB_USER=commander50  
  
# The oracle password  
COMMANDER_DB_PASSWORD=oraclePassword  
  
# The oracle batch size  
COMMANDER_DB_BATCH_SIZE=100  
  
# The existing 4.2 user/database.  
# Must be on the same server as the new
```

```
# database and readable by the target (Commander 6.x) user
OLD_COMMANDER_DB_USER=commander42
```

3. As the root user, run the background upgrader in the

```
$INSTALL_DIRECTORY/bin/commander-upgrade <path to your upgrade.properties file>
```

Note: If you are using MySQL, make sure you have put `mysql-connector-java.jar` in the `$INSTALL_DIRECTORY/server/lib` directory before you run the background upgrader.

- In Linux, enter `ec-perl /opt/electriccloud/electriccommander/bin/commander-upgrade.pl <path to your upgrade.properties file>` when the default directory is used.
- In Windows, enter `ec-perl C:\Program Files\Electric Cloud\ElectricCommander\bin\command-upgrade.pl <path to your upgrade.properties file>` when the default directory is used.

The background upgrader begins copying your existing database information into the new database.

During the database upgrade process:

- This may take a long time depending on the size of your database. If you have a very large database, it may take several days to copy your existing database information to the new database.

IMPORTANT: During the database upgrade, this step can be interrupted or stopped due to errors. If this happens, the background upgrader restarts the process to copy the database information, which may increase the time it takes to complete this step.

You can repeat this step more than once if needed.

- The background upgrader does not send output to the console. Depending on the size of your database, the database upgrade process may take hours or possibly days.
- Although the background upgrader does the bulk of the conversion of the database from ElectricCommander 4.2.x to ElectricFlow 7.x, the remaining database conversion occurs during the ElectricFlow upgrade. For a large database, this might take many hours.

IMPORTANT: After starting the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, do not interrupt the upgrade process.

If the upgrade process is interrupted, you might corrupt your database. You have to restore your database from a previous backup.

- Once you start the database upgrade process, ensure that the old ElectricFlow server is not shut down and then restarted. This avoids a situation where the background upgrader can fail to detect source database changes when trying to keep it synchronized with the target (ElectricFlow 7.x) database.

The following sequence of events rarely occurs, yet may cause the background upgrader to fail to detect source (original) database changes when the upgrade process is creating a target (ElectricFlow 7.x) database copy and keeping it in sync with the source (original) database.

- a. A run of one or more source (original) database entries with the greatest ID numbers in use have already been copied to the target (ElectricFlow 7.x) database.
- b. This run of entries is deleted from the source (original) database, which reduces the greatest ID number in use.
- c. All the deletions are successfully propagated to the target (ElectricFlow 7.x) database, and the old ElectricFlow version is shut down.
- d. While the old ElectricFlow version is down, the background upgrader process does not complete successfully, so the deletions still are not copied to the target (ElectricFlow 7.x) database.
- e. The old ElectricFlow version restarts and checks the source (original) database, finds the greatest ID currently in use, and sets that ID plus one as the ID of the next entry to be created.
- f. The old ElectricFlow server creates an entry reusing the ID of a deleted object that was not copied to the source (original) database, and by chance the entry:
 - Is in the same table as the deleted entry with the same ID
 - Has the same revision number, which is most likely if both were revision 1
 - Is a type that does not have a name, or else it has the same name as the deleted entry, and
 - Is not completely identical to the deleted entry

Based on all criteria available to the background upgrader, the two objects appear identical, so the background upgrader does not know that it must delete the target (ElectricFlow 7.x) database entry and replace it with the new entry from the source (original) database. The problem is that the two are not actually identical, and these differences are not propagated to the target (ElectricFlow 7.x) database.

- g. During the remaining background upgrader process, the background upgrader does not update the entry in the source (original) database.

To ensure that this does not happen, after you start the database upgrade process, *do not* shut down the old ElectricFlow server during the upgrade process. However, if you must do so, do one of the following:

- Run the background upgrader to completion before restarting the old ElectricFlow server

- If the previous action is not practicable (for example, it requires too much downtime for the old ElectricFlow server), wipe the partially-copied target (ElectricFlow 7.x) database and start the upgrade process again.
4. If you are using Linux, enter the `tail -f $INSTALL_DIRECTORY/logs/upgrade.log` to verify that the database upgrade is progressing by checking the end of the log (also referred to as tailing the log).

`$INSTALL_DIRECTORY` is the current working directory from which the background upgrader or installer runs.

5. Check the log file to see if the database upgrade process succeeded.

When this pass of the database upgrade is complete, the log file is in a subdirectory called *logs* in the current working directory from which the background upgrader runs.

If the process succeeds, it returns a zero (0) exit code, and the last few lines of its log file show that the process has succeeded.

6. (Optional) Run the background upgrader again one or more times.

If your ElectricCommander 4.2.x database has changed a lot since the background upgrader was last run, running the background upgrader again after its initial pass allows you to capture the ongoing changes to the database because the ElectricCommander 4.2.x server is still running. The background upgrader will replicate additions, deletions, and updates from the source (original) database to the target (ElectricFlow 7.x) database incrementally, which may shorten your downtime for the actual ElectricFlow 6.x installation.

When you run the background upgrader again, the background upgrader overwrites the log file after the database upgrade is completed.

Step 1.5: Troubleshooting the Database Upgrade

Use the following information to troubleshoot the database upgrade:

Issue	What to Do	Links to Help topics
What ectool options to use for troubleshooting	Use <code>--help</code> and <code>--version</code> .	Troubleshooting: Using ectool Options on page 7-62
Check the status of the database upgrade or determine why the upgrade failed.	Review the upgrade.log file.	Troubleshooting: upgrade.log File on page 7-60
The database user name and password have changed.	Go to Troubleshooting: When the Database User Name and Password Change .	Troubleshooting: When the Database User Name and Password Change on page 7-62
When the source (original) and target (ElectricFlow 7.x) databases are on different servers.	Go to Troubleshooting: When the Source and Target Databases Are on Different Servers on page 7-63.	Troubleshooting: When the Source and Target Databases Are on Different Servers on page 7-63

Issue	What to Do	Links to Help topics
Using the <code>--final</code> flag with the background upgrader.	Go to Troubleshooting: The --final Flag .	Troubleshooting: The --final Flag
The Oracle database upgrade fails.	Go to Troubleshooting: Oracle Database Upgrade on page 7-65.	Troubleshooting: Oracle Database Upgrade on page 7-65
The database does not have enough disk space for the database upgrade.	Go to Troubleshooting: Oracle Database Upgrade on page 7-65.	Troubleshooting: Oracle Database Upgrade on page 7-65
The background upgrader can fail to detect database changes.	Go to Troubleshooting: Background Upgrader Can Fail to Detect Database Changes on page 7-66.	Troubleshooting: Background Upgrader Can Fail to Detect Database Changes on page 7-66
Error messages	Go to Troubleshooting: Error Messages on page 7-61.	Troubleshooting: Error Messages on page 7-61

Troubleshooting the Background Upgrader

If the database upgrade process fails, check `upgrade.log` and examine the file for potential reasons for the failure.

- Look at the status entries and error messages in the log file.
- Look for incorrect, invalid, or missing values and update the `upgrade.properties` file with the correct values.
- Look for Java stack traces in the log file and on the screen of your workstation.
- Look for connection issues with the database server.
- If the background upgrader lost its connection to the database server, check its physical and network connections to the database server and try to reconnect.
- If the `upgrade.properties` file has incorrect or missing information, an error occurs and the installation fails.
- Compare the `upgrade.properties` file with the current `database.properties` file to find incorrect or missing values.
- Look for properties that could affect communication with the database server.

After resolving all the issues, restart the database upgrade process.

Troubleshooting: upgrade.log File

Review the `upgrade.log` file when you want to do the following:

- Check the status of the database upgrade because it does not seem like it is doing anything.

You can verify that the upgrade is progressing by using the `tail -f` command on the `upgrade.log` file in a separate terminal window.

Enter `tail -f $INSTALL_DIRECTORY/logs/upgrade.log` to check the status of the database upgrade. `$INSTALL_DIRECTORY` is the current working directory from which the background upgrader or installer runs. This command is available only for Linux platforms.

In Linux, if `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/bin/`, enter `tail -f ../logs/upgrade.log`.

If you are running the background upgrader, the log file is in a subdirectory called `logs` in the current working directory from which the background upgrader runs.

If the background upgrader is run automatically while the installer runs, the log file is in a subdirectory called `logs` in the `$INSTALL_DIRECTORY` directory.

Depending on the amount of data to copy from the source to the target (ElectricFlow 7.x), you may not see changes in the `upgrade.log` file for up to six to ten hours at a time when the database upgrade process is running.

- Determine why a database upgrade failed.

The `upgrade.log` file is located at `../logs/upgrade.log`, where `..` represents your current working directory for running the background upgrader (the database upgrade tool).

When the background upgrader is run by the ElectricFlow installer, the current working directory will be subdirectory of the temp directory.

Troubleshooting: Error Messages

These error messages may appear when you are upgrading ElectricCommander 4.2.x to ElectricFlow7.x.

Warning Regarding the ElectricCommander 4.2.x Database Schema

When the background upgrader is running, you need to check the `upgrade.log` file for errors. If you see the following warning near the end of the log, the upgrade is not yet fully complete.

```
2014-11-12T16:19:17.085 | WARN | main | | | MigrateDatabaseImpl | Due to issues with
Oracle support for the SERIALIZABLE isolation level, it is not possible on Oracle to
efficiently do an exact sync of the 5.0 schema to the 4.2 schema while the 4.2.x
Commander server is still running. The 5.0 schema is currently NOT fully synchronized
and NOT safe to use with 5.0, and its foreign key constraints are still disabled.
However, the bulk of the data should now have been copied across, so future upgrader
runs should be relatively quick. The 5.0 installer will automatically run the upgrader
again after shutting down the 4.2.x server: as long as that upgrader run succeeds, it
will bring the 5.0 schema into exact sync with the state of the 4.2 schema and
reenable the constraints.
```

Note that the error states that it is not possible on Oracle to efficiently do an exact sync of the 5.0 schema to the 4.2 schema while the 4.2.x Commander server is still running.

If you see this warning, continue following the upgrade instructions. When you run the background upgrader with the `--final` option or the ElectricFlow installer, the issue should be resolved.

For more information on the `--final` flag, go to [Troubleshooting: Using the --final Flag](#) on page 7-63

Table or View Does Not Exist

See also: [Troubleshooting: When the Source and Target Databases Are on Different Servers](#) on page 7-63

When the background upgrader is running, this error is in the upgrade.log file:

```
Unexpected error:org.hibernate.exception.SQLGrammarException: could not execute
statement : java.sql.SQLException: ORA-00942: table or view does not exist
```

Generally, this error occurs when the owner of the ElectricFlow 7.x database table owner does not have read access to the Commander 4.2.x database tables. If the user permissions that you are using to install ElectricFlow 7.x do not have read access to the Commander 4.2.x tables, the user (you) cannot run the background upgrader. It cannot read the information to which it needs from the Commander 4.2.x tables to do the conversion.

This error can also occur if the ElectricCommander 4.2.x database was not configured correctly in the database.properties file or if the wrong database sever was configured in the file.

To resolve this error:

1. Have your DBA give the ElectricFlow 7.x user access to the Commander 4.2 database tables.
2. Rerun the background upgrader.

Access Denied

This error may occur when the background upgrader is running:

```
2014-11-03T07:21:34.344 | DEBUG | CP-pool-watch-thread-commander | | | |
EcConnectionHook | Failed to acquire connection: java.sql.SQLException: Access denied
for user 'commander'@'localhost' (using password: YES).
```

To resolve this error:

1. Define the real password (not hash) for COMMANDER_DB_PASSWORD in upgrade.properties file.
2. Restart the background upgrader.

Trying to Connect to the Database

This error may occur when the ElectricFlow server tries to connect to the database server:

```
Unexpected error:java.lang.RuntimeException: Timed out while waiting to connect to the
database (possibly the database is unavailable, or one or more of the database
connection parameters was set incorrectly
```

To resolve this error:

1. Check the number of MySQL instances in the host.
2. Check the ports used by these MySQL instances and use the correct one.

Troubleshooting: Using ectool Options

Use these ectool options to troubleshoot the database upgrade:

Option	Description
--help	Displays this help information.
--version	Displays the version information

Troubleshooting: When the Database User Name and Password Change

Problem

While you are upgrading ElectricFlow with clustering, the database user name and password changes. The database upgrade cannot be successfully completed.

Solution

Do one of the following tasks:

- Update the `database.properties` file on the ElectricFlow server.
If this does not resolve the issue, find where the file is stored and update that file.
- If the `database.properties` file is stored with other configuration files on the Apache ZooKeeper server,
 1. Update the user name and password in the `database.properties` file.
 2. Run the following command to populate the ZooKeeper server with the new username and password:

```
COMMANDER_ZK_CONNECTION=<ZooKeeper_Server_IP>:2181
../jre/bin/java -jar
../server/bin/zk-config-tool-jar-with-dependencies.jar
com.electriccloud.commander.cluster.ZKConfigTool
--databasePropertiesFile database.properties
--keystoreFile keystore
--passkeyFile passkey
--commanderPropertiesFile commander.properties
```

Troubleshooting: When the Source and Target Databases Are on Different Servers**Problem**

The upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x was successfully completed at a site. After the upgrade, the data center was moved to a new site. However, the new MS SQL database could not be brought up at the new site, and a SQL error message about a "table or view not found" appears.

Solution

This is an permissions issue. At the new site, when the user was bringing the new database, the system did not recognize this user and did not allow the user to bring up the database. The system identified the user as a new user who did not have the proper permissions.

To resolve this issue, assign the user bringing up the database at the new site the proper permissions to complete the task. Make sure that the database at the new site is the same database instance that was at the original site.

Troubleshooting: Using the --final Flag**Problem**

The following error message appears indicating that there are issues with MS SQL server support for the SNAPSHOT isolation level:

```
Modifying table ec_property...
Unexpected error:org.hibernate.exception.SQLGrammarException: could not execute
statement : com.microsoft.sqlserver.jdbc.SQLServerException: The instance of the SQL
Server Database Engine cannot obtain a LOCK resource at this time. Rerun your
statement when there are fewer active users. Ask the database administrator to check
the lock and memory configuration for this instance, or to check for long-running
transactions. - could not execute statement :
org.hibernate.exception.SQLGrammarException: could not execute statement

at org.hibernate.exception.internal.SQLStateConversionDelegate.convert
```

```
(SQLStateConversionDelegate.java:123)
```

```
at org.hibernate.exception.internal.StandardSQLExceptionConverter.convert
(StandardSQLExceptionConverter.java:49)
```

...

This message appears when you enter

`commander-upgrade.pl --final` in one of the following scenarios:

- Scenario 1:

The ElectricCommander 4.2.x database server is running while the background upgrader is running. The background upgrader is capturing the ongoing updates to the source (ElectricCommander 4.2.x) database and prepopulating the target (ElectricFlow 7.x) database.

Before running the ElectricFlow installer, you run the background upgrader one last time while the ElectricCommander 4.2.x database server is running. After you enter `jcommander-upgrade.pl --final`, the database upgrade fails, and the error message appears.

- Scenario 2:

The target (ElectricFlow 7.x) database is on a different server than the source (original) database.

When you install ElectricFlow 7.x, the database upgrader automatically runs. The database upgrade fails, and the error message appears.

Solution

The error message means that while the ElectricCommander 4.2.x server is still running, the background upgrader was successfully run without the `--final` flag against the database.

However, it is not possible to perform an exact synchronization of the ElectricCommander 4.2.x database to the ElectricFlow 7.x database while the Commander 4.2.x server is still running. The ElectricFlow 7.x database is not yet fully synchronized and is not safe to use with ElectricFlow 7.x. The database foreign key constraints are still disabled. Most of the data should now have been copied from the source and target (ElectricFlow 7.x) database, and future upgrade runs should take less time.

During your troubleshooting, you may notice that the size of the database has decreased. The size of the database is expected to decrease before the last run of the background upgrader. It is common for an active database with many create and delete updates to have unused space in it. The upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x compacts the database. To verify that the ElectricCommander 4.2.x database has been upgraded to the ElectricFlow 7.x database, you can compare row counts for various tables in the ElectricCommander 4.2.x and ElectricFlow 7.x databases. Most of the tables should match exactly.

Do one of the following to complete the database upgrade:

- Scenario 1:

When you are installing the ElectricFlow 7.x server on the same machine as the ElectricCommander 4.2.x server, we recommend that you shut down the ElectricCommander 4.2.x server and run the ElectricFlow installer with the appropriate options. During the installation, the background upgrader runs automatically with the `--final` flag to complete the database upgrade.

- Shut down the Commander 4.2.x server before running the background upgrader the last time. Then run the background upgrader using `commander-upgrade.pl --final`. If the background upgrader runs successfully, the ElectricFlow 7.x database and Commander 4.2.x database are completely synchronized and the constraints are enabled again.

Troubleshooting: Oracle Database Upgrade

Problems

The Oracle database upgrade fails, and an error message appears in the log.

Possible causes include the following:

- The batch size affects how long the database upgrade takes. If the batch size is too large, the database upgrade process can fail.
- If the database does not have enough disk space for the database upgrade, one of the following occurs:
 - The background upgrader runs for at least 15 minutes and then fails. A log message about the database running out of temporary or undo space appears. The message includes an Oracle error such as ORA-00001, ORA-01652, ORA-08176, ORA-08177, or ORA-15555).
 - The background upgrader hangs for 12 or more hours without SQL messages being added to the log during that time. The database server has used up all its available storage space. It could have run out of disk space, or have run out of RAM and be disk-thrashing.

Solution

Your DBA reviews the log messages and may need to do the following tasks:

- Edit the `upgrade.properties` file.

If the batch size is too large, decrease the `COMMANDER_DB_BATCH_SIZE` from 100 to a smaller value, such as 50.

However, if you use a smaller batch size, this increases how long it takes to run the database upgrade process.

- Edit the timeout settings in Oracle.
- Make more space available in the database, use a lower value for the `COMMANDER_DB_BATCH_SIZE` setting in the `upgrader.conf` file, or do both tasks.

Then rerun the background upgrader.

- In general, smaller batches consume less database temporary and undo space.

The lower `COMMANDER_DB_BATCH_SIZE` values, however, are likely to make transferring a large database take longer, because smaller batches incur more overhead by resynchronizing to the updated state of the ElectricCommander 4.2 database more frequently.

- We recommend an initial value of 100 for `COMMANDER_DB_BATCH_SIZE`.

If that value fails, try 30, 10, 5, 2, and 1 in that order.

- Increase the `UNDO_RETENTION` value to 12 hours or one day.
- For most tables, the default batch row limit used is `COMMANDER_DB_BATCH_SIZE` x 100,000 rows per SQL query.

The background upgrader might temporarily reduce the row limit for some tables if it detects a problem.

- In general, it is not necessary (and is counterproductive) to wipe the ElectricFlow 7.x database before rerunning the background upgrader. The background upgrader copies differences from the ElectricCommander 4.2 to the ElectricFlow 7.x database to synchronize them (inserting, updating, or deleting rows from the ElectricFlow 7.x database as required). This means that a new run of the background upgrader continues building on any data that has already been transferred.

However, if the background upgrader repeatedly fails after only a few minutes with no data transferred and an error about the database being invalid, try wiping the ElectricFlow 7.x database and rerunning the background upgrader.

- If the background upgrader stops, or if you have to kill it, check that:
 - You are connected to your database.
 - The temporary resources that the background upgrader was using have been released. If not, have your DBA kill the background upgrader's connection.

Troubleshooting: Background Upgrader Can Fail to Detect Database Changes

The following sequence of events rarely occurs, but may cause the background upgrader to fail to detect existing source database changes when the upgrade process is creating a new target (ElectricFlow 7.x) database copy and keeping it synchronized with the source (original) database.

1. A run of one or more source (original) database entries with the greatest ID numbers in use have already been copied to the target (ElectricFlow 7.x) database.
2. This run of entries is deleted from the source (original) database, which reduces the greatest ID number in use.
3. Some or all of the deletions are not yet successfully propagated to the target (ElectricFlow 7.x) database, and the old ElectricFlow version is shut down.
4. While the old ElectricFlow version is down, the background upgrader process does not complete successfully, so the deletions still are not copied to the target (ElectricFlow 7.x) database.
5. The old ElectricFlow version restarts and checks the source (original) database, finds the greatest ID currently in use, and sets that ID plus one as the ID of the next entry to be created.
6. The old ElectricFlow server creates an entry reusing the ID of a deleted object that was not copied to the source (original) database, and by chance the entry:
 - Is in the same table as the deleted entry with the same ID
 - Has the same revision number (which is most likely if both were revision 1)
 - Is a type that does not have a name, or else it has the same name as the deleted entry, but
 - Is not completely identical to the deleted entry

Based on all criteria used by the background upgrader, the two objects appear identical, so the background upgrader does not know that it must delete the target (ElectricFlow 7.x) database entry and replace it with the new entry from the source (original) database. However, the two are not actually identical, and these differences are not propagated to the target (ElectricFlow 7.x) database.

7. During the remaining background upgrader process, ElectricFlow does not further update the entry in the source (original) database.

To ensure that this does not happen, after you start the database upgrade process, *do not* shut down the old ElectricFlow server during the upgrade process. If, for some reason, you must do so, then do one of the following:

- Run the background upgrader to completion before restarting the old ElectricFlow server.
- If this is not practicable (for example, it requires too much downtime for the old ElectricFlow server), wipe the partially-copied target (ElectricFlow 7.x) database and start the entire upgrade process again.

Step 1.6: Shut Down the Source (Original) Server

1. Shut down the ElectricCommander 4.2.x server.

Step 1.7: Run the Installer

To run the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:


```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Note: Although the ElectricFlow installer is in `$INSTALL_DIRECTORY/src`, do not launch it from there.

1. Review the information about the following methods:

These are the various installation methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms on page 2-1](#) and [Supported Agent Platforms on page 2-10](#).

User Interface Method

This method provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Method

This method provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is available only for Linux platforms.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version are not be removed or modified and remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Method

This method provides a non-interactive command-line upgrade for supported server platforms. You might prefer this installation method when upgrading multiple remote agents and servers.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server with an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall the software. You can install the repository server on a different machine to avoid uninstalling and reinstalling the software.

Stand-Alone Repository Server or Web Server Upgrade

You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting with ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server.

For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and the "Run the Installer" step for each use case.

Note: ElectricFlow 7.x web servers must use an ElectricFlow 5.x agent to retrieve log files to be displayed in the web interface. This *workspace proxy* agent is typically set to *localhost* but can be modified with the `config.php` file.

Non-Standard Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent. You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

2. Select a method:

User Interface Upgrade Method

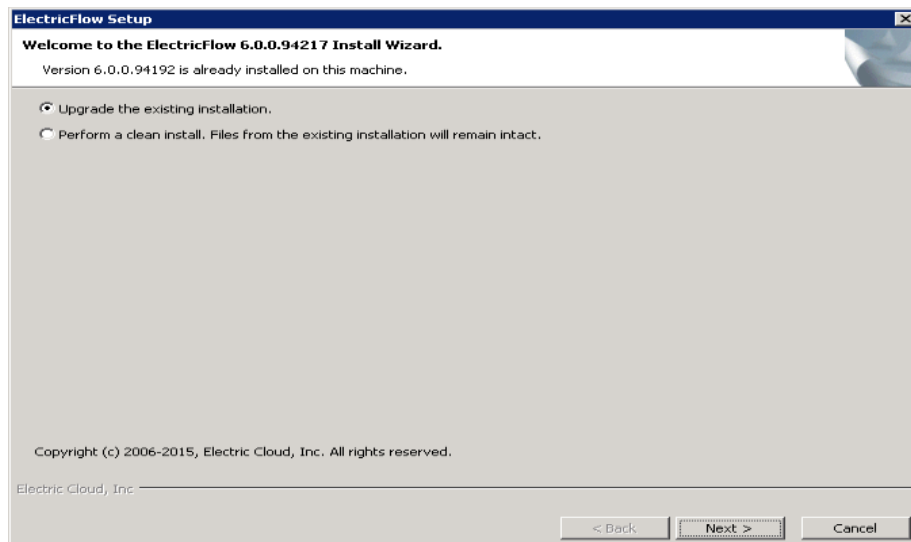
Use the following procedure to upgrade the ElectricFlow software. Review the "Prepare for the Database Upgrade" step in the use before performing this procedure.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 7.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database, if applicable.

1. Open the installer and double-click the `ElectricFlow-<version>` file to begin the installation process.

The Welcome to the ElectricFlow Installer screen appears.



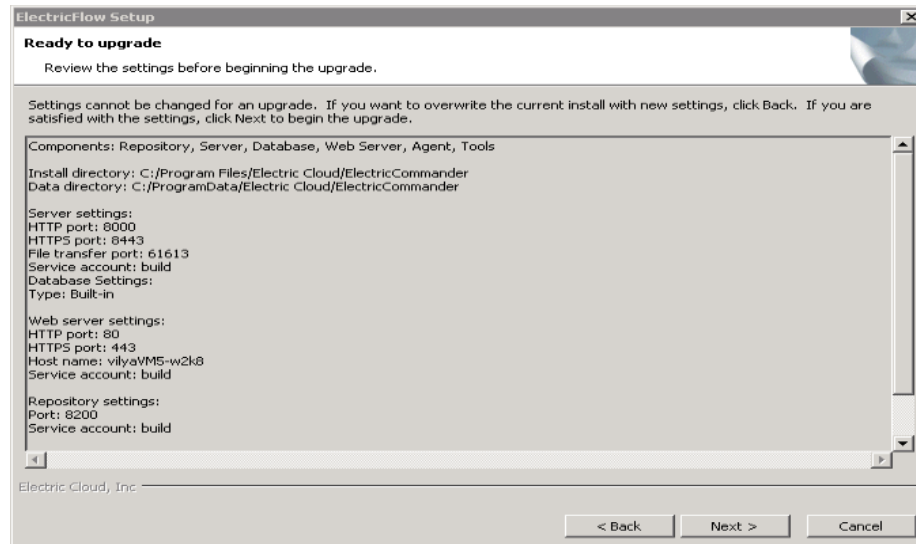
2. Choose one of the following options:

- Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
- Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next**.
4. Provide your **New Database Name** (non-Oracle) or **New Database Username** (Oracle) and the corresponding **Password** and click **Next** to continue.
5. Click **Next** to upgrade the existing installation.

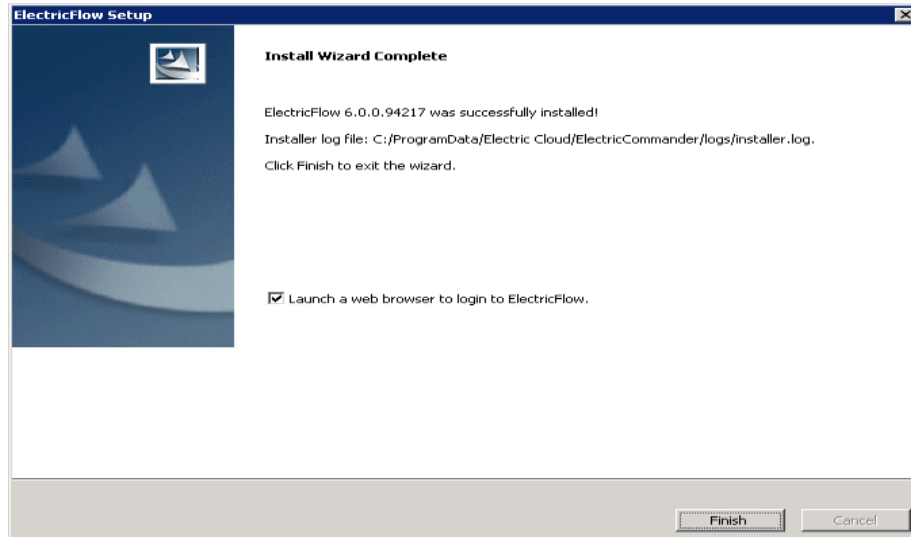
The Ready to Install screen appears.



6. Review the upgrade settings. Use the Back button to change your selections if necessary.
7. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. This process can take up to fifteen minutes to complete. After this process is complete, the new

ElectricFlow version is installed.

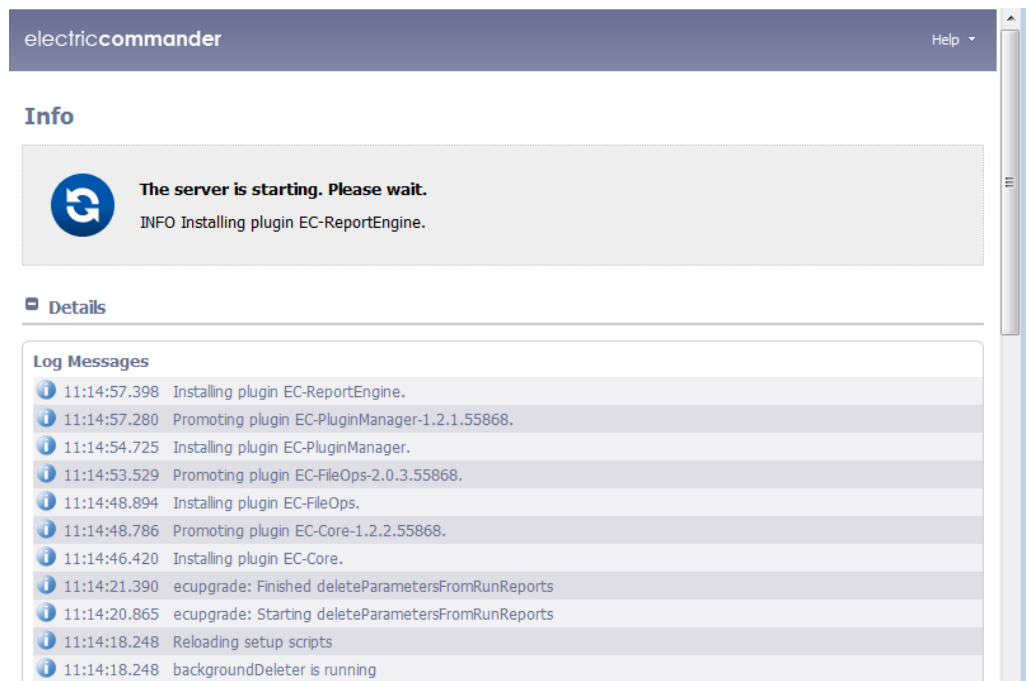


8. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
9. Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line: `ectool getServerStatus`.

After clicking **Finish**, you might see a web page similar to the following screen if the upgrade

is still in progress.



Interactive Command-line Upgrade Method

Use the following procedure to complete the command-line upgrade of a Linux platform. Review the "Preparing for the Database Upgrade" step in the use case before performing this procedure.

When you upgrade ElectricFlow from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database.

1. Change directories to \$INSTALL_DIRECTORY/src.
2. Go to the command line and enter one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer automatically brings up the graphical user interface. To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
This will install ElectricFlow on your computer. Continue? [n/Y]
```

3. Enter `y`.

The system displays the following message:

```
Upgrade the existing <version> installation to version
<version>? [n/Y]
```

4. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter: `y`.
- Enter `n` to exit the installer.

The system displays the following message:

For a non-Oracle database: New database name and database password.

For an Oracle database: New database username and database password.

Then the system displays the following message:

```
Installing agent...
Installing server...
Copied log file to "/opt/electriccloud/electriccommander/logs"
ElectricFlow <version> was successfully installed!
Installer log file: /opt/electriccloud/electriccommander/logs/installer
.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in silent (unattended) mode with no user interface for either Windows or Linux.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to ElectricFlow 7.x saves the existing database.

To upgrade ElectricFlow in silent mode:

1. Connect your ElectricFlow server to the database.
2. Go to the command line and enter one of the following commands from a command line to begin the silent upgrade (for non-Oracle and Oracle databases, respectively):

```
./ElectricFlow-<version> --mode silent --upgradeNewDBName <newdatabasename> --upgradeDBPassword <password> --haveRunMigrationUtility
```

```
./ElectricFlow-<version> --mode silent --upgradeNewDBUserName <newdatabaseusername> --upgradeDBPassword <password> --haveRunMigrationUtility
```

where `<newdatabasename>` is the name of the new database, `<newdatabaseusername>` is the name of database user account in `database.properties`, `<password>` is the password for the database user account in `database.properties`, and `haveRunMigrationUtility` confirms that you have run the migration tool. The database user name and password in `database.properties` may be different than the ones used for the background upgrader.

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (`<datadir>/repository-data`) is:

- **UNIX** - `/opt/electriccloud/electriccommander/repository-data`
- **Windows** - `C:\ProgramData\Electric Cloud\ElectricCommander\repository-data`

Use Case 2: Large Database Upgrade on New Hardware

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, the database must be upgraded. The database upgrade process includes creating a new database for ElectricFlow 7.x and copying the data from the source (original ElectricCommander 4.2.x) database to the new target (ElectricFlow 7.x) database.

This process is more involved than in previous releases because the database changes are more extensive than in previous ElectricFlow upgrades. You use the background upgrader in the upgrade bundle to upgrade the database.

IMPORTANT: You cannot upgrade the ElectricCommander 4.2.x H2 database. If you want to continue using a built-in database in ElectricFlow 7.x, follow the procedures in [Use Case 5: Built-In Database Upgrade on Existing Hardware](#) on page 7-103 or [Use Case 6: Built-In Database Upgrade on New Hardware](#) on page 7-116.

IMPORTANT: Before you start the database upgrade, you must create a new, empty database that is the same type as the existing source (original) database and is the target (ElectricFlow 7.x) database for the upgrade.

IMPORTANT: After you have run the ElectricFlow 7.x server, you cannot rerun the background upgrader against the target (ElectricFlow 6.x) database because ElectricFlow 7.x makes further database upgrades.

When you are performing an upgrade on existing hardware, you do not need to manually run the `commander-upgrade.pl` command with the `--final` flag, because this command is run automatically when you execute the installer to upgrade your existing ElectricFlow server to the new version.. You run the background upgrader against your source database, and then run the installer to install the latest software version. Running the installer will run the background upgrader with the `--final` flag, ensuring that your database is finalized for the upgrade.

However, when installing new hardware (specifically, a new server that is not pointing at the newly upgraded database) while performing the upgrade, the `--final` flag will not run automatically for the background upgrader, and you will need to run it manually. Before running the installer, you need to run the background upgrader one more time with `--final` flag as an option in this command:

```
commander-upgrade.pl --final.
```

This will mark the database as being finalized and allow ElectricFlow to work properly.

If you are installing on new hardware while upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x, perform the following steps.

Step 2.1: Download and Extract the Upgrade Bundle

You use the upgrade bundle (`upgrade-bundle.zip`) to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x.

The `upgrade-bundle.zip` file has these files:

- `README.txt`: README file for Windows and Linux
- `migrateTemplates.pl`: Migration tool to convert `$(jobId)` and `$(workflowId)` to `jobCounter` and `workflowCounter` in the `jobNameTemplate` and `workflowName` template.
- `commander-upgrade-jar-with-dependencies.jar`: Background upgrader to update the database. You can run this tool or the background upgrader script before installing the latest ElectricFlow software.
- `commander-upgrade.pl`: Background upgrader script to update the database. You run this script or the background upgrader tool before installing the latest ElectricFlow software.
- `ElectricFlow-<version>`: ElectricFlow installer to install the latest software version.

The Windows installer for ElectricFlow 7.x is `ElectricFlow-<version>.exe`.

The corresponding Linux installer is `ElectricFlow-<version>`.

IMPORTANT: Only use the background upgrader when upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x. Do not use it when you are updating ElectricFlow 5.x to a later ElectricFlow version.

Perform these steps to download and extract the upgrade bundle:

1. Download the upgrade bundle (upgrade-bundle.zip) from the Electric Cloud site.
2. Extract the following files into the `$INSTALL_DIRECTORY` directory:
 - **README file:**
Put `README.txt` into any `$INSTALL_DIRECTORY` directory.
 - **Migration tool for UUIDs:**
Put `migrateTemplates.pl` into the `$INSTALL_DIRECTORY/bin` directory.
 - **Background upgrader:**
Put `commander-upgrade.pl` into the `$INSTALL_DIRECTORY/bin` directory.
Put `commander-upgrade-jar-with-dependencies.jar` into the `$INSTALL_DIRECTORY/Utils` directory.
3. Extract the `ElectricFlow-<version>` installer into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Linux Example:

If the default directory `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/`:

- Put `README.txt` into any `/opt/electriccloud/electriccommander` directory.
- Put `migrateTemplates.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `/opt/electriccloud/electriccommander/Utils` directory.
- Put `ElectricFlow-<version>` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Windows Example:

If the default directory `$INSTALL_DIRECTORY` is `C:\Program Files\Electric Cloud\ElectricCommander\`:

- Put `README.txt` into any `C:\Program Files\Electric Cloud\ElectricCommander` directory.
- Put `migrateTemplates.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `C:\Program Files\Electric Cloud\ElectricCommander\Utils` directory.
- Put `ElectricFlow-<version>.exe` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `C:\temp` directory.

Step 2.2: Run the Migration Tool to Convert `jobID` and `workflowId` References

This section describes how to convert the IDs for entity types to automatically-generated universally unique identifiers (UUIDs), 36-character strings. The most common entity changes that affect you are the `jobId` and `workflowId` properties.

The migration tool is an ec-perl script called *migrateTemplates.pl* that converts *jobNameTemplate* and *workflowNameTemplate* properties containing `$(jobId)` and `$(workflowId)` to use a counter instead (by converting those references to *jobCounter* and *workflowCounter* respectively).

This section describes how to convert `$(jobId)` and `$(workflowId)` references in your projects to *jobCounter* and *workflowCounter* references respectively. By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing *before* and *after* values). Before you make the actual changes, make sure you examine the proposed changes thoroughly to avoid job name collision.

IMPORTANT: Before *migrateTemplates.pl* is invoked, the `admin` user must log into ectool using the following command and enter the password when prompted:

```
ectool --server xyz login admin
```

1. Go to the `$INSTALL_DIRECTORY/bin` directory.
2. To invoke the migration tool, enter the following command:

```
ec-perl migrateTemplates.pl --server <server name> [options]
```

where `--server <server name>` specifies the ElectricFlow server to migrate. The default server is *localhost*.

The following options are available:

Option	Description
<code>--counters <server project></code>	Specifies whether job counters are server-wide or project-specific. Default: project .
<code>--dryRun <1 0></code>	<ul style="list-style-type: none"> • 1 – Shows the prospective changes in a file that you can examine. Verify the changes before the migration tool makes them. • 0 – Actually makes the changes. Default: 1.
<code>--project <project></code>	Limits migration to the specified project.
<code>--startingJobId <n></code>	Initializes the job counter to the specified number. Default: the current greatest job ID.
<code>--startingWorkflowId <n></code>	Initializes the workflow counter to the specified number. Default: the current greatest workflow ID.

3. View the results.

The migration tool runs successfully when the script completes and output appears showing the changes that were made.

The tool fails when an error message appears.

Example 1

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter across the entire ElectricFlow system and want the next job number to be 1001:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000
```

The above example will change `$(jobId)` to `[/increment /server/jobCounter]` and initialize the job counter to 1000. This means the next job run on the server will be number 1001. The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000 --dryRun 0
```

Example 2

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter for each project and want the counters to continue from the current greatest job ID.

```
ec-perl migrateTemplates.pl --server Server1 --counters project
```

The above example will change `$(jobId)` to `[/increment /myJob/project/jobCounter]` and use the current greatest job ID as the initial value for the job counter.

The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters project --dryRun 0
```

Step 2.3: Prepare for the Upgrade

The database upgrade process to ElectricFlow 7.x can be time-consuming. To minimize the duration of this process, the upgrade bundle has a background upgrader to allow your ElectricFlow system to build the ElectricFlow 7.x database while the ElectricCommander 4.2.x server is still running. You run the background upgrader before you run the installer, which can minimize downtime for the ElectricFlow environment and reduce the time required for the upgrade process.

Your DBA should review the following steps to determine which are necessary for your environment to prepare for the database upgrade:

1. Estimate how long the database upgrade may take.

It may take many hours to several days to upgrade the database, depending on the database hardware, database size, and the load on your database server. The background upgrader might put significant extra load on your database server while it runs in the background.

In our internal testing on moderately powerful database servers, database upgrades in the size range of 10 GB to several tens of GB usually took many hours on MySQL or Microsoft SQL Server, and several days on Oracle.

If you have a large database, you can run the background upgrader more than once to synchronize the source (original) and target (ElectricFlow 7.x) databases while the ElectricCommander 4.2.x server is still running.

To estimate the duration of the next database upgrade pass:

- a. Find the log file called upgrade.log.

When you run the background upgrader, the log file is in a subdirectory called *logs* of the current working directory from which the background upgrader was run.

- b. In the log file, search for the phrase "End of pass <number> through tables."

- c. Calculate the duration of a database upgrade pass.

Look for the timestamp when the current upgrade ended and the timestamp when the previous upgrade ended.

The duration is the difference between the time when the last pass ended and the time when the previous pass ended.

The downtime for upgrading ElectricFlow 7.x is expected to take at least as long as the last pass, because it runs one more pass of the background upgrader. On MySQL, it can take many hours more than that, because MySQL requires entire tables to be copied when new columns are added to them.

IMPORTANT: Change tracking is enabled when you upgrade to ElectricFlow 5.3 and later. This can significantly increase the time (about an hour) to complete the upgrade. If you want to upgrade without enabling change tracking, add this line to the `database.properties` file before starting the software upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

```
COMMANDER_DB_AUDITING_ENABLED=false
```

2. Running the background upgrader puts additional load on the database. To minimize both the background upgrader runtime and your upgrade downtime:

- If you usually run your ElectricFlow database server with your database located on a nonlocal disk such as a network file server, we recommend relocating both your ElectricCommander 4.2.x and new ElectricFlow databases to one or more local disks.
- If your current database contains a lot of unused free space, we recommend compacting it. If it is fragmented, we recommend defragmenting it or moving it to a solid-state drive (SSD).

3. Estimate the amount of disk space for the database upgrade.

Keep the following in mind:

- You need to make sure that the database server has sufficient disk space for the database upgrade, including disk space for the data copied during the upgrade, log files, redo files, and undo files. The log space depends on the database type and the logging strategy.
- We recommend that you have at least three times (3x) the size of your current ElectricCommander 4.2.x source (original) database of free disk space.
- Some of this space is needed temporarily for the upgrade process to run. After the upgrade process is completed, the disk space for the new target (ElectricFlow 7.x) database is only a little larger than before.

4. (Optional) Delete old data in the database.

IMPORTANT: If your ElectricCommander 4.2 .x source (original) database contains a lot of old job data, we *strongly* recommend that you delete as much of this data as possible before running the background upgrader, which will help to speed up the eventual upgrade. It will take time for this deletion effort to complete and for the background upgrader to run.

5. Perform the following tasks that are necessary for your environment:

Upgrade Testing

In most implementations, ElectricFlow is being used in an environment that affects many users. We recommend that you test your upgrade on a separate test server to understand all aspects of the upgrade process. This minimizes the potential impacts to downstream users.

We also strongly recommend that you make a copy of your database and walk through the database upgrade process on a test environment before performing the actual database upgrade on your production environment. The result of this is to understand the impact of performing the database upgrade in your environment and to understand the effect that the upgrade has on your environment. If you need a test license to perform such a test, contact support@electric-cloud.com to request a temporary license.

Backing Up Your Existing ElectricFlow Data

If you are upgrading an ElectricFlow server, it is *extremely* important that you back up your existing ElectricFlow data *before* upgrading. For more information about backups, see [ElectricFlow Server Backups](#) on page 11-1.

- Always back up the Plugins Directory. The default location is the `plugins` subdirectory within the data directory.
- Always back up the files that contain your configuration and custom settings. To ensure that all important settings are saved, back up the following subdirectories in `<data_dir>`:
 - The entire `conf` subdirectory (which contains the ElectricFlow server and agent configuration files)
 - Apache web server configuration files in the `apache/conf` subdirectory
 - The local MySQL database configuration file, `my.ini`, in the `mysql` subdirectory (if applicable)
- Always back up any other files where you have created custom configurations, specified other custom information, or created any type of modification.

Note: The number of ElectricFlow files you may have modified is too numerous to list, so you may want to back up the entire data directory and other miscellaneous files that may have changed.

- If you use an artifact repository, back up your ElectricFlow repository configuration files in the `conf/repository` subdirectory.
- Determine if any changes were made to the custom editor or preflight driver script properties (installed by default). Back up those files if changes were made.

These properties are stored in the server-level property sheet, which can be viewed in the platform UI by accessing the **Administration > Server** tab.

Custom editors are stored in the nested sheet named `ec_customEditors`. Preflight driver scripts are stored in the nested sheet named `ec_preflight`. The upgrade process overwrites default custom editor and preflight driver scripts with current versions. We recommend backing up any custom properties you created by renaming the property. For example, change `ec_preflight/clientDrivers/perforce` to `ec_preflight/clientDrivers/perforce_modified`.

Upgrade Installer Preservation

After you back up your ElectricFlow server, create a folder where you can download the ElectricFlow-<version> installation file.

Step 2.4: Run the Background Upgrader

To upgrade the database:

1. Create the new empty target (ElectricFlow 7.x) database with these characteristics:
 - The new database must be the same type as the existing source (original) database and is the target (ElectricFlow 7.x) database for the upgrade.
 - Your source (original) and target (ElectricFlow 7.x) databases must be on the same database server.
 - The text encoding settings of the source (original) and target (ElectricFlow 7.x) databases must match exactly.
- a. For MySQL databases, perform the following additional tasks:
 - You must give ElectricFlow permissions to create new databases and to add and delete tables at all times.
 - You must also give ElectricFlow DROP SCHEMA and CREATE SCHEMA permissions to create and delete databases during the duration of the upgrade process. This includes the intermediate database with the "_upgrade" suffix in its name.

IMPORTANT: If you do not give ElectricFlow the permissions previously listed, the database upgrade will fail with an error similar to the following:

```
2015-07-16T22:22:16.885 | ERROR | bootstrap | | |
schemaMaintenance | ServerStatus | Database upgrade failed:
Migration failed for change set
com/electriccloud/commander/upgrade/upgrade_5062_5063.xml::5062_
5063-0040::commander:
Reason: liquibase.exception.DatabaseException:
com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException:
Access denied for user 'admin'@'%' to database 'commander_
upgrade'
```

- These are the external MySQL database characteristics:
 - When upgrading ElectricFlow, ensure that `default-storage-engine=INNODB` is in the `[mysqld]` section in `my.cnf/my.ini`.
 - An install and upgrade log file named `installer.log` is created in the `logs` subdirectory in the data directory.

- Although the background upgrader does the bulk of the conversion of the MySQL server database from ElectricCommander 4.2.x to ElectricFlow 7.x, the remaining database conversion occurs during the ElectricFlow upgrade. For a large database, this might take many hours.

IMPORTANT: After starting the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, do not interrupt the upgrade process.

If the upgrade process is interrupted, you might corrupt your database. You have to restore your database from a previous backup.

- The following applies when the database is updated:
 - Upgrading saves the existing local MySQL database, if applicable.
 - Use the `ectool` to view the upgrade progress. On a command line, enter `ectool getServerStatus`.

b. For Oracle databases, do the following tasks:

- A database schema is a *user* in Oracle. One user has one schema.
- You must have a new user for the new empty target (ElectricFlow 7.x) database tables.
- We recommend putting the new user tables in their own workspace because this can make managing disk usage easier.
- Ensure that you have allocated adequate table space, temp space, and undo space.
- You must explicitly grant read access permissions to the target (ElectricFlow 7.x) database user so that this user can read information in the source (original) database user's tables.

Example: Enter the following SQL commands.

To specify the source (original) database, enter `GRANT "CONNECT" TO "COMMANDER42"` where `COMMANDER42` is the source (original) database schema or user.

To specify the destination (ElectricFlow 7.x) database, enter `GRANT "RESOURCE" TO "FLOW60"` where `FLOW60` is the destination (ElectricFlow 6.x) database schema or user.

To give the destination (ElectricFlow 7.x) database read access permissions to each table in the source (original) database, enter `GRANT SELECT ON "COMMANDER42"."EC_AGENT" TO "FLOW60"` where `COMMANDER42` is the source (original) database schema or user, `EC_AGENT` is a table in the source (original) database, and `FLOW60` is the destination (ElectricFlow 6.x) database schema or user.

Enter a command like this for each table in the source (original) database.

For a detailed example, see http://docs.electric-cloud.com/eflow_doc/5_4/PDF/grant_access.sql.

- The target (ElectricFlow 7.x) database user must have the same password as the source (original) database user.
- Ensure that the database `UNDO_RETENTION` parameter is set large enough (at least 12 to 24 hours).

2. Create or edit an upgrade.properties file.

The upgrade.properties file provides information to the database upgrade process about the target (ElectricFlow 7.x) and source (original) databases.

- In Linux, the file is usually in
`/opt/electriccloud/electriccommander/conf/upgrade.properties`
- In Windows, the file is usually in
`C:\ProgramData\Electric Cloud\ElectricCommander\conf\upgrade.properties`

Create or edit the file one of these ways:

- a. Use the wizard to help you edit the file.

- b. Create or edit a new `upgrade.properties` file with the `upgrade.properties` file template.
 - i. Copy the template content corresponding to your target (ElectricFlow 7.x) database type.
 - ii. Open a new empty file in a text editor.
 - iii. Paste the entire template contents into the new file.
 - iv. Replace the template values (after the = symbols) with the values from your existing ElectricFlow deployment.

Except for the `COMMANDER_DB_PASSWORD`, `COMMANDER_DB_BATCH_SIZE`, `COMMANDER_DB_NAME` (MySQL or Microsoft SQL Server), and `COMMANDER_DB_USER` (Oracle) values, use your existing database value in one of these locations:

- **Administration > Database Configuration** tab in your existing ElectricFlow server
- `$INSTALL_DIRECTORY/conf/database.properties` file in your existing ElectricFlow deployment for Linux (for example, `/opt/electriccloud/electriccommander/conf/database.properties`)
- `$INSTALL_DIRECTORY\conf\database.properties` file in your existing ElectricFlow deployment for Windows (for example, `C:\ProgramData\ElectricCloud\ElectricCommander\conf\database.properties`)

Make sure that you maintain the formatting of the values in the `database.properties` file.

- v. Save the file as `upgrade.properties`.
- vi. Because the `upgrade.properties` file contains the plaintext password for your ElectricFlow database, restrict read access to the file, or make sure that it is deleted when you no longer need it.

MySQL

```
# Example configuration file for mysql
COMMANDER_DB_TYPE=mysql

# The ip or dns of the mysql server
COMMANDER_DB_HOST=myDbHost.example.com

# The mysql port (defaults to 3306)
COMMANDER_DB_PORT=3306

# The name of the target (Commander 6.x) database
COMMANDER_DB_NAME=commander50

# The mysql username
COMMANDER_DB_USER=mySqlUser

# The mysql password
COMMANDER_DB_PASSWORD=mySqlPassword

# The mysql batch size
COMMANDER_DB_BATCH_SIZE=100
```

```
# The existing 4.2 database.  
# Must be on the same server as the new  
# database and readable by the mySqlUser  
OLD_COMMANDER_DB_NAME=commander42
```

Microsoft SQL Server

```
# Example configuration file for sqlserver  
COMMANDER_DB_TYPE=sqlserver  
  
# The ip or dns of the sqlserver server  
COMMANDER_DB_HOST=myDbHost.example.com  
  
# The sqlserver port (defaults to 1433)  
COMMANDER_DB_PORT=1433  
  
# The name of the target (Commander 6.x) database  
COMMANDER_DB_NAME=commander50  
  
# The sqlserver username  
COMMANDER_DB_USER=sqlServerUser  
  
# The sqlserver password  
COMMANDER_DB_PASSWORD=sqlServerPassword  
  
# The sqlserver batch size  
COMMANDER_DB_BATCH_SIZE=100  
  
# The existing 4.2 database.  
# Must be on the same server as the new  
# database and readable by the sqlServerUser  
OLD_COMMANDER_DB_NAME=commander42
```

Oracle

```
# Example configuration file for oracle  
COMMANDER_DB_TYPE=oracle  
  
# The ip or dns of the TNS listener  
COMMANDER_DB_HOST=myDbHost.example.com  
  
# The TNS listener port (defaults to 1521)  
COMMANDER_DB_PORT=1521  
  
# The sid of database  
COMMANDER_DB_NAME=commanderSID  
  
# The target (Commander 5.x) username/database  
COMMANDER_DB_USER=commander50  
  
# The oracle password  
COMMANDER_DB_PASSWORD=oraclePassword  
  
# The oracle batch size  
COMMANDER_DB_BATCH_SIZE=100  
  
# The existing 4.2 user/database.  
# Must be on the same server as the new
```

```
# database and readable by the target (Commander 6.x) user
OLD_COMMANDER_DB_USER=commander42
```

3. As the root user, run the background upgrader in the

```
$INSTALL_DIRECTORY/bin/commander-upgrade <path to your upgrade.properties file>
```

Note: If you are using MySQL, make sure you have put `mysql-connector-java.jar` in the `$INSTALL_DIRECTORY/server/lib` directory before you run the background upgrader.

- In Linux, enter `ec-perl /opt/electriccloud/electriccommander/bin/commander-upgrade.pl <path to your upgrade.properties file>` when the default directory is used.
- In Windows, enter `ec-perl C:\Program Files\Electric Cloud\ElectricCommander\bin\command-upgrade.pl <path to your upgrade.properties file>` when the default directory is used.

The background upgrader begins copying your existing database information into the new database.

During the database upgrade process:

- This may take a long time depending on the size of your database. If you have a very large database, it may take several days to copy your existing database information to the new database.

IMPORTANT: During the database upgrade, this step can be interrupted or stopped due to errors. If this happens, the background upgrader restarts the process to copy the database information, which may increase the time it takes to complete this step.

You can repeat this step more than once if needed.

- The background upgrader does not send output to the console. Depending on the size of your database, the database upgrade process may take hours or possibly days.
- Although the background upgrader does the bulk of the conversion of the database from ElectricCommander 4.2.x to ElectricFlow 7.x, the remaining database conversion occurs during the ElectricFlow upgrade. For a large database, this might take many hours.

IMPORTANT: After starting the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, do not interrupt the upgrade process.

If the upgrade process is interrupted, you might corrupt your database. You have to restore your database from a previous backup.

- Once you start the database upgrade process, ensure that the old ElectricFlow server is not shut down and then restarted. This avoids a situation where the background upgrader can fail to detect source database changes when trying to keep it synchronized with the target (ElectricFlow 7.x) database.

The following sequence of events rarely occurs, yet may cause the background upgrader to fail to detect source (original) database changes when the upgrade process is creating a target (ElectricFlow 7.x) database copy and keeping it in sync with the source (original) database.

- a. A run of one or more source (original) database entries with the greatest ID numbers in use have already been copied to the target (ElectricFlow 7.x) database.
- b. This run of entries is deleted from the source (original) database, which reduces the greatest ID number in use.
- c. All the deletions are successfully propagated to the target (ElectricFlow 7.x) database, and the old ElectricFlow version is shut down.
- d. While the old ElectricFlow version is down, the background upgrader process does not complete successfully, so the deletions still are not copied to the target (ElectricFlow 7.x) database.
- e. The old ElectricFlow version restarts and checks the source (original) database, finds the greatest ID currently in use, and sets that ID plus one as the ID of the next entry to be created.
- f. The old ElectricFlow server creates an entry reusing the ID of a deleted object that was not copied to the source (original) database, and by chance the entry:
 - Is in the same table as the deleted entry with the same ID
 - Has the same revision number, which is most likely if both were revision 1
 - Is a type that does not have a name, or else it has the same name as the deleted entry, and
 - Is not completely identical to the deleted entry

Based on all criteria available to the background upgrader, the two objects appear identical, so the background upgrader does not know that it must delete the target (ElectricFlow 7.x) database entry and replace it with the new entry from the source (original) database. The problem is that the two are not actually identical, and these differences are not propagated to the target (ElectricFlow 7.x) database.

- g. During the remaining background upgrader process, the background upgrader does not update the entry in the source (original) database.

To ensure that this does not happen, after you start the database upgrade process, *do not* shut down the old ElectricFlow server during the upgrade process. However, if you must do so, do one of the following:

- Run the background upgrader to completion before restarting the old ElectricFlow server

- If the previous action is not practicable (for example, it requires too much downtime for the old ElectricFlow server), wipe the partially-copied target (ElectricFlow 7.x) database and start the upgrade process again.
4. If you are using Linux, enter the `tail -f $INSTALL_DIRECTORY/logs/upgrade.log` to verify that the database upgrade is progressing by checking the end of the log (also referred to as tailing the log).

`$INSTALL_DIRECTORY` is the current working directory from which the background upgrader or installer runs.

5. Check the log file to see if the database upgrade process succeeded.

When this pass of the database upgrade is complete, the log file is in a subdirectory called *logs* in the current working directory from which the background upgrader runs.

If the process succeeds, it returns a zero (0) exit code, and the last few lines of its log file show that the process has succeeded.

6. (Optional) Run the background upgrader again one or more times.

If your ElectricCommander 4.2.x database has changed a lot since the background upgrader was last run, running the background upgrader again after its initial pass allows you to capture the ongoing changes to the database because the ElectricCommander 4.2.x server is still running. The background upgrader will replicate additions, deletions, and updates from the source (original) database to the target (ElectricFlow 7.x) database incrementally, which may shorten your downtime for the actual ElectricFlow 6.x installation.

When you run the background upgrader again, the background upgrader overwrites the log file after the database upgrade is completed.

Step 2.5: Troubleshooting the Database Upgrade

Use the following information to troubleshoot the database upgrade:

Issue	What to Do	Links to Help topics
What ectool options to use for troubleshooting	Use <code>--help</code> and <code>--version</code> .	Troubleshooting: Using ectool Options on page 7-62
Check the status of the database upgrade or determine why the upgrade failed.	Review the upgrade.log file.	Troubleshooting: upgrade.log File on page 7-60
The database user name and password have changed.	Go to Troubleshooting: When the Database User Name and Password Change .	Troubleshooting: When the Database User Name and Password Change on page 7-62
When the source (original) and target (ElectricFlow 7.x) databases are on different servers.	Go to Troubleshooting: When the Source and Target Databases Are on Different Servers on page 7-63.	Troubleshooting: When the Source and Target Databases Are on Different Servers on page 7-63

Issue	What to Do	Links to Help topics
Using the <code>--final</code> flag with the background upgrader.	Go to Troubleshooting: The --final Flag .	Troubleshooting: The --final Flag
The Oracle database upgrade fails.	Go to Troubleshooting: Oracle Database Upgrade on page 7-65.	Troubleshooting: Oracle Database Upgrade on page 7-65
The database does not have enough disk space for the database upgrade.	Go to Troubleshooting: Oracle Database Upgrade on page 7-65.	Troubleshooting: Oracle Database Upgrade on page 7-65
The background upgrader can fail to detect database changes.	Go to Troubleshooting: Background Upgrader Can Fail to Detect Database Changes on page 7-66.	Troubleshooting: Background Upgrader Can Fail to Detect Database Changes on page 7-66
Error messages	Go to Troubleshooting: Error Messages on page 7-61.	Troubleshooting: Error Messages on page 7-61

Troubleshooting the Background Upgrader

If the database upgrade process fails, check `upgrade.log` and examine the file for potential reasons for the failure.

- Look at the status entries and error messages in the log file.
- Look for incorrect, invalid, or missing values and update the `upgrade.properties` file with the correct values.
- Look for Java stack traces in the log file and on the screen of your workstation.
- Look for connection issues with the database server.
- If the background upgrader lost its connection to the database server, check its physical and network connections to the database server and try to reconnect.
- If the `upgrade.properties` file has incorrect or missing information, an error occurs and the installation fails.
- Compare the `upgrade.properties` file with the current `database.properties` file to find incorrect or missing values.
- Look for properties that could affect communication with the database server.

After resolving all the issues, restart the database upgrade process.

Troubleshooting: upgrade.log File

Review the `upgrade.log` file when you want to do the following:

- Check the status of the database upgrade because it does not seem like it is doing anything.

You can verify that the upgrade is progressing by using the `tail -f` command on the `upgrade.log` file in a separate terminal window.

Enter `tail -f $INSTALL_DIRECTORY/logs/upgrade.log` to check the status of the database upgrade. `$INSTALL_DIRECTORY` is the current working directory from which the background upgrader or installer runs. This command is available only for Linux platforms.

In Linux, if `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/bin/`, enter `tail -f ../logs/upgrade.log`.

If you are running the background upgrader, the log file is in a subdirectory called `logs` in the current working directory from which the background upgrader runs.

If the background upgrader is run automatically while the installer runs, the log file is in a subdirectory called `logs` in the `$INSTALL_DIRECTORY` directory.

Depending on the amount of data to copy from the source to the target (ElectricFlow 7.x), you may not see changes in the `upgrade.log` file for up to six to ten hours at a time when the database upgrade process is running.

- Determine why a database upgrade failed.

The `upgrade.log` file is located at `../logs/upgrade.log`, where `..` represents your current working directory for running the background upgrader (the database upgrade tool).

When the background upgrader is run by the ElectricFlow installer, the current working directory will be subdirectory of the temp directory.

Troubleshooting: Error Messages

These error messages may appear when you are upgrading ElectricCommander 4.2.x to ElectricFlow7.x.

Warning Regarding the ElectricCommander 4.2.x Database Schema

When the background upgrader is running, you need to check the `upgrade.log` file for errors. If you see the following warning near the end of the log, the upgrade is not yet fully complete.

```
2014-11-12T16:19:17.085 | WARN | main | | | MigrateDatabaseImpl | Due to issues with
Oracle support for the SERIALIZABLE isolation level, it is not possible on Oracle to
efficiently do an exact sync of the 5.0 schema to the 4.2 schema while the 4.2.x
Commander server is still running. The 5.0 schema is currently NOT fully synchronized
and NOT safe to use with 5.0, and its foreign key constraints are still disabled.
However, the bulk of the data should now have been copied across, so future upgrader
runs should be relatively quick. The 5.0 installer will automatically run the upgrader
again after shutting down the 4.2.x server: as long as that upgrader run succeeds, it
will bring the 5.0 schema into exact sync with the state of the 4.2 schema and
reenable the constraints.
```

Note that the error states that it is not possible on Oracle to efficiently do an exact sync of the 5.0 schema to the 4.2 schema while the 4.2.x Commander server is still running.

If you see this warning, continue following the upgrade instructions. When you run the background upgrader with the `--final` option or the ElectricFlow installer, the issue should be resolved.

For more information on the `--final` flag, go to [Troubleshooting: Using the --final Flag](#) on page 7-63

Table or View Does Not Exist

See also: [Troubleshooting: When the Source and Target Databases Are on Different Servers](#) on page 7-63

When the background upgrader is running, this error is in the upgrade.log file:

```
Unexpected error:org.hibernate.exception.SQLGrammarException: could not execute
statement : java.sql.SQLException: ORA-00942: table or view does not exist
```

Generally, this error occurs when the owner of the ElectricFlow 7.x database table owner does not have read access to the Commander 4.2.x database tables. If the user permissions that you are using to install ElectricFlow 7.x do not have read access to the Commander 4.2.x tables, the user (you) cannot run the background upgrader. It cannot read the information to which it needs from the Commander 4.2.x tables to do the conversion.

This error can also occur if the ElectricCommander 4.2.x database was not configured correctly in the database.properties file or if the wrong database sever was configured in the file.

To resolve this error:

1. Have your DBA give the ElectricFlow 7.x user access to the Commander 4.2 database tables.
2. Rerun the background upgrader.

Access Denied

This error may occur when the background upgrader is running:

```
2014-11-03T07:21:34.344 | DEBUG | CP-pool-watch-thread-commander | | | |
EcConnectionHook | Failed to acquire connection: java.sql.SQLException: Access denied
for user 'commander'@'localhost' (using password: YES).
```

To resolve this error:

1. Define the real password (not hash) for COMMANDER_DB_PASSWORD in upgrade.properties file.
2. Restart the background upgrader.

Trying to Connect to the Database

This error may occur when the ElectricFlow server tries to connect to the database server:

```
Unexpected error:java.lang.RuntimeException: Timed out while waiting to connect to the
database (possibly the database is unavailable, or one or more of the database
connection parameters was set incorrectly
```

To resolve this error:

1. Check the number of MySQL instances in the host.
2. Check the ports used by these MySQL instances and use the correct one.

Troubleshooting: Using ectool Options

Use these ectool options to troubleshoot the database upgrade:

Option	Description
--help	Displays this help information.
--version	Displays the version information

Troubleshooting: When the Database User Name and Password Change

Problem

While you are upgrading ElectricFlow with clustering, the database user name and password changes. The database upgrade cannot be successfully completed.

Solution

Do one of the following tasks:

- Update the `database.properties` file on the ElectricFlow server.
If this does not resolve the issue, find where the file is stored and update that file.
- If the `database.properties` file is stored with other configuration files on the Apache ZooKeeper server,
 1. Update the user name and password in the `database.properties` file.
 2. Run the following command to populate the ZooKeeper server with the new username and password:

```
COMMANDER_ZK_CONNECTION=<ZooKeeper_Server_IP>:2181
../jre/bin/java -jar
../server/bin/zk-config-tool-jar-with-dependencies.jar
com.electriccloud.commander.cluster.ZKConfigTool
--databasePropertiesFile database.properties
--keystoreFile keystore
--passkeyFile passkey
--commanderPropertiesFile commander.properties
```

Troubleshooting: When the Source and Target Databases Are on Different Servers

Problem

The upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x was successfully completed at a site. After the upgrade, the data center was moved to a new site. However, the new MS SQL database could not be brought up at the new site, and a SQL error message about a "table or view not found" appears.

Solution

This is an permissions issue. At the new site, when the user was bringing the new database, the system did not recognize this user and did not allow the user to bring up the database. The system identified the user as a new user who did not have the proper permissions.

To resolve this issue, assign the user bringing up the database at the new site the proper permissions to complete the task. Make sure that the database at the new site is the same database instance that was at the original site.

Troubleshooting: Using the --final Flag

Problem

The following error message appears indicating that there are issues with MS SQL server support for the SNAPSHOT isolation level:

```
Modifying table ec_property...
Unexpected error:org.hibernate.exception.SQLGrammarException: could not execute
statement : com.microsoft.sqlserver.jdbc.SQLServerException: The instance of the SQL
Server Database Engine cannot obtain a LOCK resource at this time. Rerun your
statement when there are fewer active users. Ask the database administrator to check
the lock and memory configuration for this instance, or to check for long-running
transactions. - could not execute statement :
org.hibernate.exception.SQLGrammarException: could not execute statement

at org.hibernate.exception.internal.SQLStateConversionDelegate.convert
```

```
(SQLStateConversionDelegate.java:123)
```

```
at org.hibernate.exception.internal.StandardSQLExceptionConverter.convert  
(StandardSQLExceptionConverter.java:49)
```

...

This message appears when you enter

`commander-upgrade.pl --final` in one of the following scenarios:

- Scenario 1:

The ElectricCommander 4.2.x database server is running while the background upgrader is running. The background upgrader is capturing the ongoing updates to the source (ElectricCommander 4.2.x) database and prepopulating the target (ElectricFlow 7.x) database.

Before running the ElectricFlow installer, you run the background upgrader one last time while the ElectricCommander 4.2.x database server is running. After you enter `jcommander-upgrade.pl --final`, the database upgrade fails, and the error message appears.

- Scenario 2:

The target (ElectricFlow 7.x) database is on a different server than the source (original) database.

When you install ElectricFlow 7.x, the database upgrader automatically runs. The database upgrade fails, and the error message appears.

Solution

The error message means that while the ElectricCommander 4.2.x server is still running, the background upgrader was successfully run without the `--final` flag against the database.

However, it is not possible to perform an exact synchronization of the ElectricCommander 4.2.x database to the ElectricFlow 7.x database while the Commander 4.2.x server is still running. The ElectricFlow 7.x database is not yet fully synchronized and is not safe to use with ElectricFlow 7.x. The database foreign key constraints are still disabled. Most of the data should now have been copied from the source and target (ElectricFlow 7.x) database, and future upgrade runs should be take less time.

During your troubleshooting, you may notice that the size of the database has decreased. The size of the database is expected to decrease before the last run of the background upgrader. It is common for an active database with many create and delete updates to have unused space in it. The upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x compacts the database. To verify that the ElectricCommander 4.2.x database has been upgraded to the ElectricFlow 7.x database, you can compare row counts for various tables in the ElectricCommander 4.2.x and ElectricFlow 7.x databases. Most of the tables should match exactly.

Do one of the following to complete the database upgrade:

- Scenario 1:

When you are installing the ElectricFlow 7.x server on the same machine as the ElectricCommander 4.2.x server, we recommend that you shut down the ElectricCommander 4.2.x server and run the ElectricFlow installer with the appropriate options. During the installation, the background upgrader runs automatically with the `--final` flag to complete the database upgrade.

- Shut down the Commander 4.2.x server before running the background upgrader the last time. Then run the background upgrader using `commander-upgrade.pl --final`. If the background upgrader runs successfully, the ElectricFlow 7.x database and Commander 4.2.x database are completely synchronized and the constraints are enabled again.

Troubleshooting: Oracle Database Upgrade

Problems

The Oracle database upgrade fails, and an error message appears in the log.

Possible causes include the following:

- The batch size affects how long the database upgrade takes. If the batch size is too large, the database upgrade process can fail.
- If the database does not have enough disk space for the database upgrade, one of the following occurs:
 - The background upgrader runs for at least 15 minutes and then fails. A log message about the database running out of temporary or undo space appears. The message includes an Oracle error such as ORA-00001, ORA-01652, ORA-08176, ORA-08177, or ORA-15555).
 - The background upgrader hangs for 12 or more hours without SQL messages being added to the log during that time. The database server has used up all its available storage space. It could have run out of disk space, or have run out of RAM and be disk-thrashing.

Solution

Your DBA reviews the log messages and may need to do the following tasks:

- Edit the `upgrade.properties` file.

If the batch size is too large, decrease the `COMMANDER_DB_BATCH_SIZE` from 100 to a smaller value, such as 50.

However, if you use a smaller batch size, this increases how long it takes to run the database upgrade process.

- Edit the timeout settings in Oracle.
- Make more space available in the database, use a lower value for the `COMMANDER_DB_BATCH_SIZE` setting in the `upgrader.conf` file, or do both tasks.

Then rerun the background upgrader.

- In general, smaller batches consume less database temporary and undo space.

The lower `COMMANDER_DB_BATCH_SIZE` values, however, are likely to make transferring a large database take longer, because smaller batches incur more overhead by resynchronizing to the updated state of the ElectricCommander 4.2 database more frequently.

- We recommend an initial value of 100 for `COMMANDER_DB_BATCH_SIZE`.

If that value fails, try 30, 10, 5, 2, and 1 in that order.

- Increase the `UNDO_RETENTION` value to 12 hours or one day.
- For most tables, the default batch row limit used is `COMMANDER_DB_BATCH_SIZE` x 100,000 rows per SQL query.

The background upgrader might temporarily reduce the row limit for some tables if it detects a problem.

- In general, it is not necessary (and is counterproductive) to wipe the ElectricFlow 7.x database before rerunning the background upgrader. The background upgrader copies differences from the ElectricCommander 4.2 to the ElectricFlow 7.x database to synchronize them (inserting, updating, or deleting rows from the ElectricFlow 7.x database as required). This means that a new run of the background upgrader continues building on any data that has already been transferred.

However, if the background upgrader repeatedly fails after only a few minutes with no data transferred and an error about the database being invalid, try wiping the ElectricFlow 7.x database and rerunning the background upgrader.

- If the background upgrader stops, or if you have to kill it, check that:
 - You are connected to your database.
 - The temporary resources that the background upgrader was using have been released. If not, have your DBA kill the background upgrader's connection.

Troubleshooting: Background Upgrader Can Fail to Detect Database Changes

The following sequence of events rarely occurs, but may cause the background upgrader to fail to detect existing source database changes when the upgrade process is creating a new target (ElectricFlow 7.x) database copy and keeping it synchronized with the source (original) database.

1. A run of one or more source (original) database entries with the greatest ID numbers in use have already been copied to the target (ElectricFlow 7.x) database.
2. This run of entries is deleted from the source (original) database, which reduces the greatest ID number in use.
3. Some or all of the deletions are not yet successfully propagated to the target (ElectricFlow 7.x) database, and the old ElectricFlow version is shut down.
4. While the old ElectricFlow version is down, the background upgrader process does not complete successfully, so the deletions still are not copied to the target (ElectricFlow 7.x) database.
5. The old ElectricFlow version restarts and checks the source (original) database, finds the greatest ID currently in use, and sets that ID plus one as the ID of the next entry to be created.
6. The old ElectricFlow server creates an entry reusing the ID of a deleted object that was not copied to the source (original) database, and by chance the entry:
 - Is in the same table as the deleted entry with the same ID
 - Has the same revision number (which is most likely if both were revision 1)
 - Is a type that does not have a name, or else it has the same name as the deleted entry, but
 - Is not completely identical to the deleted entry

Based on all criteria used by the background upgrader, the two objects appear identical, so the background upgrader does not know that it must delete the target (ElectricFlow 7.x) database entry and replace it with the new entry from the source (original) database. However, the two are not actually identical, and these differences are not propagated to the target (ElectricFlow 7.x) database.

7. During the remaining background upgrader process, ElectricFlow does not further update the entry in the source (original) database.

To ensure that this does not happen, after you start the database upgrade process, *do not* shut down the old ElectricFlow server during the upgrade process. If, for some reason, you must do so, then do one of the following:

- Run the background upgrader to completion before restarting the old ElectricFlow server.
- If this is not practicable (for example, it requires too much downtime for the old ElectricFlow server), wipe the partially-copied target (ElectricFlow 7.x) database and start the entire upgrade process again.

Step 2.6: Run the Installer

To run the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
```

```
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Note: Although the ElectricFlow installer is in `$INSTALL_DIRECTORY/src`, do not launch it from there.

1. Review the information about the following methods:

These are the various installation methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms on page 2-1](#) and [Supported Agent Platforms on page 2-10](#).

User Interface Method

This method provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Method

This method provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is available only for Linux platforms.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version are not be removed or modified and remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Method

This method provides a non-interactive command-line upgrade for supported server platforms. You might prefer this installation method when upgrading multiple remote agents and servers.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server with an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall the software. You can install the repository server on a different machine to avoid uninstalling and reinstalling the software.

Stand-Alone Repository Server or Web Server Upgrade

You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting with ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server.

For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and the "Run the Installer" step for each use case.

Note: ElectricFlow 7.x web servers must use an ElectricFlow 5.x agent to retrieve log files to be displayed in the web interface. This *workspace proxy* agent is typically set to *localhost* but can be modified with the `config.php` file.

Non-Standard Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent. You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

2. Select a method:

User Interface Upgrade Method

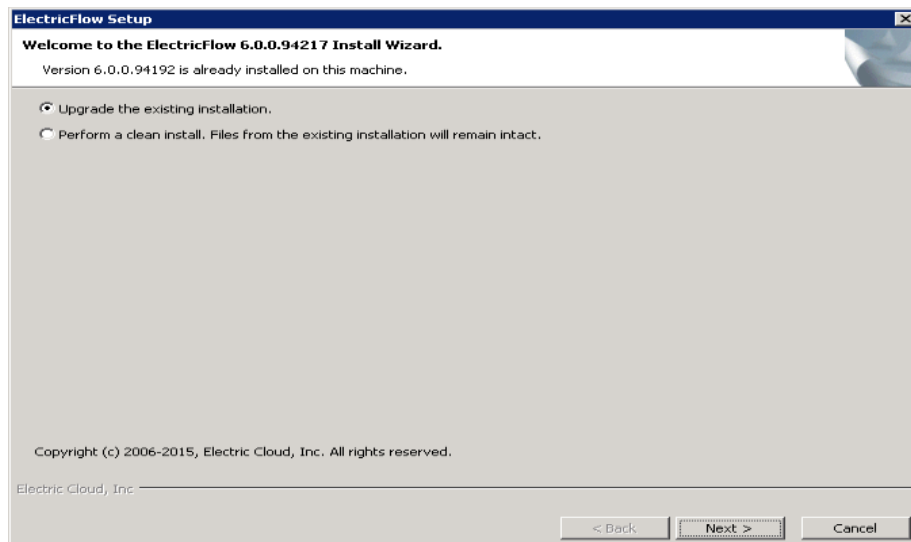
Use the following procedure to upgrade the ElectricFlow software. Review the "Prepare for the Database Upgrade" step in the use before performing this procedure.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 7.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database, if applicable.

1. Open the installer and double-click the `ElectricFlow-<version>` file to begin the installation process.

The Welcome to the ElectricFlow Installer screen appears.

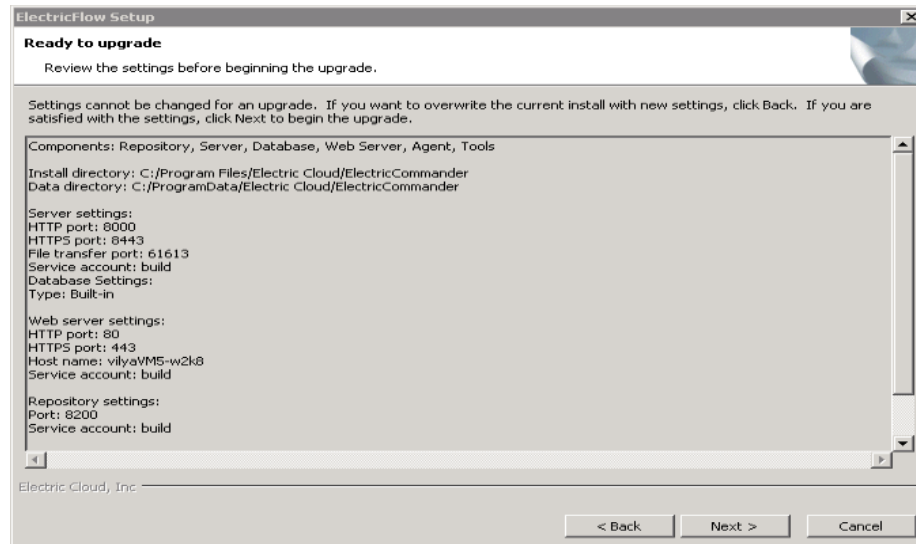


2. Choose one of the following options:
 - Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
 - Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next**.
4. Provide your **New Database Name** (non-Oracle) or **New Database Username** (Oracle) and the corresponding **Password** and click **Next** to continue.
5. Click **Next** to upgrade the existing installation.

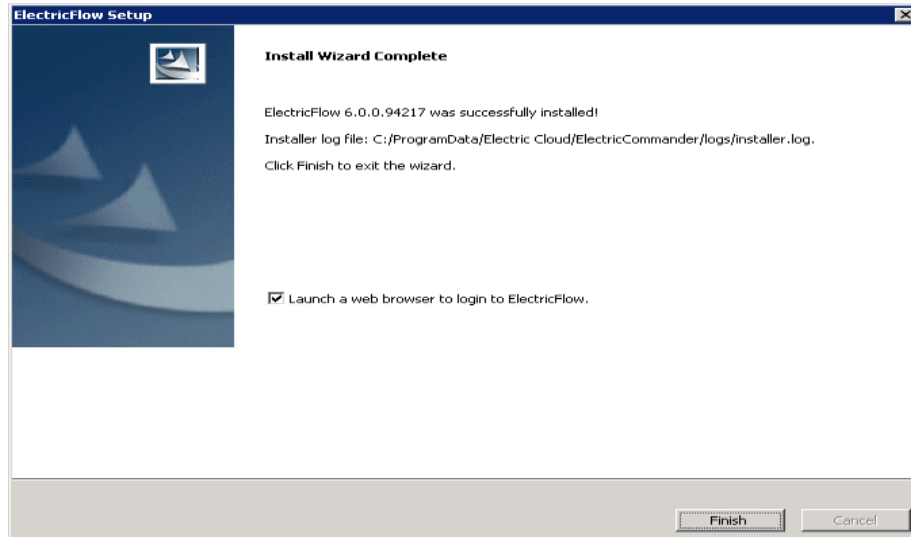
The Ready to Install screen appears.



6. Review the upgrade settings. Use the Back button to change your selections if necessary.
7. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. This process can take up to fifteen minutes to complete. After this process is complete, the new

ElectricFlow version is installed.

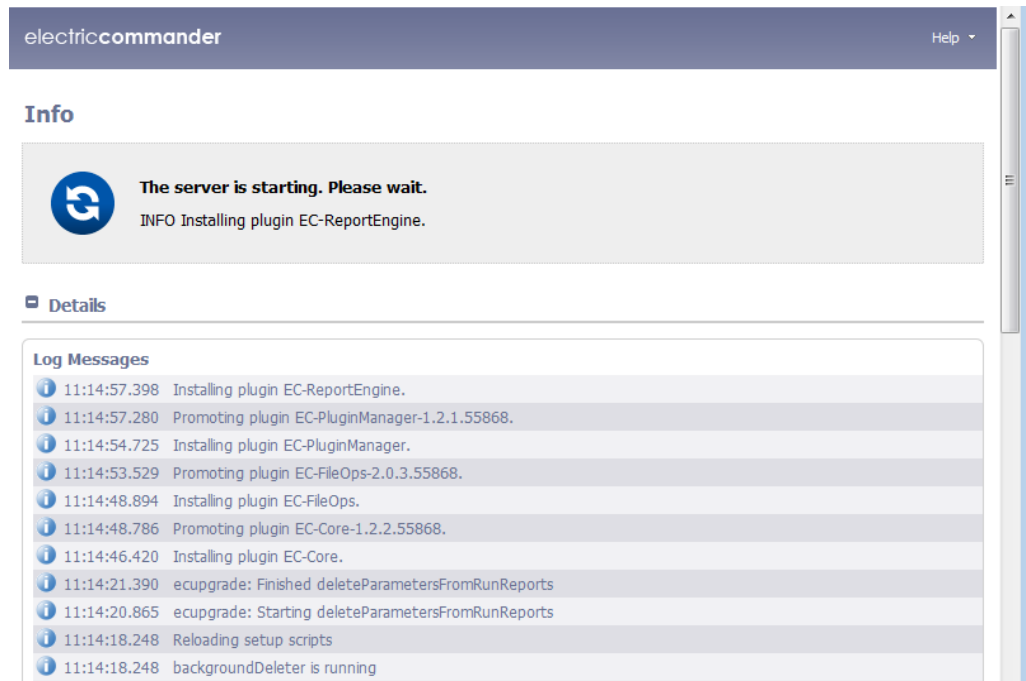


8. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
9. Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line: `ectool getServerStatus`.

After clicking **Finish**, you might see a web page similar to the following screen if the upgrade

is still in progress.



Interactive Command-line Upgrade Method

Use the following procedure to complete the command-line upgrade of a Linux platform. Review the "Preparing for the Database Upgrade" step in the use case before performing this procedure.

When you upgrade ElectricFlow from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database.

1. Change directories to `$INSTALL_DIRECTORY/src`.
2. Go to the command line and enter one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer automatically brings up the graphical user interface. To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
This will install ElectricFlow on your computer. Continue? [n/Y]
```

3. Enter `y`.

The system displays the following message:

```
Upgrade the existing <version> installation to version
<version>? [n/Y]
```

4. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter: `y`.
- Enter `n` to exit the installer.

The system displays the following message:

For a non-Oracle database: New database name and database password.

For an Oracle database: New database username and database password.

Then the system displays the following message:

```
Installing agent...
Installing server...
Copied log file to "/opt/electriccloud/electriccommander/logs"
ElectricFlow <version> was successfully installed!
Installer log file: /opt/electriccloud/electriccommander/logs/installer
.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in silent (unattended) mode with no user interface for either Windows or Linux.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to ElectricFlow 7.x saves the existing database.

To upgrade ElectricFlow in silent mode:

1. Connect your ElectricFlow server to the database.
2. Go to the command line and enter one of the following commands from a command line to begin the silent upgrade (for non-Oracle and Oracle databases, respectively):

```
./ElectricFlow-<version> --mode silent --upgradeNewDBName <newdatabasename> --upgradeDBPassword <password> --haveRunMigrationUtility
```

```
./ElectricFlow-<version> --mode silent --upgradeNewDBUserName <newdatabaseusername> --upgradeDBPassword <password> --haveRunMigrationUtility
```

where `<newdatabasename>` is the name of the new database, `<newdatabaseusername>` is the name of database user account in `database.properties`, `<password>` is the password for the database user account in `database.properties`, and `haveRunMigrationUtility` confirms that you have run the migration tool. The database user name and password in `database.properties` may be different than the ones used for the background upgrader.

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (`<datadir>/repository-data`) is:

- **UNIX** - `/opt/electriccloud/electriccommander/repository-data`
- **Windows** - `C:\ProgramData\Electric Cloud\ElectricCommander\repository-data`

Step 2.7: Shut Down the Existing Server

1. Shut down the ElectricCommander 4.2.x server.

Step 2.8: Complete the Upgrade

1. Point the new ElectricFlow 7.x server to the new target (ElectricFlow 7.x) database, either through the UI or entering `ectool setDatabaseConfiguration`.

Use Case 3: Small Database Upgrade on Existing Hardware

Note that this use case omits the background upgrader step, which is normally used only for large databases or where downtime is an issue.

Perform the following steps to upgrade ElectricFlow on existing hardware:

Step 3.1: Download and Extract the Upgrade Bundle

You use the upgrade bundle (`upgrade-bundle.zip`) to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x.

The `upgrade-bundle.zip` file has these files:

- `README.txt`: README file for Windows and Linux
- `migrateTemplates.pl`: Migration tool to convert `$(jobId)` and `$(workflowId)` to `jobCounter` and `workflowCounter` in the `jobNameTemplate` and `workflowName` template.
- `commander-upgrade-jar-with-dependencies.jar`: Background upgrader to update the database. You can run this tool or the background upgrader script before installing the latest ElectricFlow software.
- `commander-upgrade.pl`: Background upgrader script to update the database. You run this script or the background upgrader tool before installing the latest ElectricFlow software.
- `ElectricFlow-<version>`: ElectricFlow installer to install the latest software version.

The Windows installer for ElectricFlow 7.x is `ElectricFlow-<version>.exe`.

The corresponding Linux installer is `ElectricFlow-<version>`.

IMPORTANT: Only use the background upgrader when upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x. Do not use it when you are updating ElectricFlow 5.x to a later ElectricFlow version.

Perform these steps to download and extract the upgrade bundle:

1. Download the upgrade bundle (`upgrade-bundle.zip`) from the Electric Cloud site.
2. Extract the following files into the `$INSTALL_DIRECTORY` directory:
 - README file:
Put `README.txt` into any `$INSTALL_DIRECTORY` directory.
 - Migration tool for UUIDs:
Put `migrateTemplates.pl` into the `$INSTALL_DIRECTORY/bin` directory.
 - Background upgrader:
Put `commander-upgrade.pl` into the `$INSTALL_DIRECTORY/bin` directory.
Put `commander-upgrade-jar-with-dependencies.jar` into the `$INSTALL_DIRECTORY/Utils` directory.
3. Extract the `ElectricFlow-<version>` installer into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Linux Example:

If the default directory `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/`:

- Put `README.txt` into any `/opt/electriccloud/electriccommander` directory.
- Put `migrateTemplates.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `/opt/electriccloud/electriccommander/Utils` directory.
- Put `ElectricFlow-<version>` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Windows Example:

If the default directory `$INSTALL_DIRECTORY` is `C:\Program Files\Electric Cloud\ElectricCommander\`:

- Put `README.txt` into any `C:\Program Files\Electric Cloud\ElectricCommander` directory.
- Put `migrateTemplates.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `C:\Program Files\Electric Cloud\ElectricCommander\utils` directory.
- Put `ElectricFlow-<version>.exe` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `C:\temp` directory.

Step 3.2: Run the Migration Tool to Convert `$(jobID)` and `$(workflowId)` References

This section describes how to convert the IDs for entity types to automatically-generated universally unique identifiers (UUIDs), 36-character strings. The most common entity changes that affect you are the `jobId` and `workflowId` properties.

The migration tool is an ec-perl script called *migrateTemplates.pl* that converts `jobNameTemplate` and *workflowNameTemplate* properties containing `$(jobId)` and `$(workflowId)` to use a counter instead (by converting those references to `jobCounter` and `workflowCounter` respectively).

This section describes how to convert `$(jobID)` and `$(workflowId)` references in your projects to `jobCounter` and `workflowCounter` references respectively. By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing *before* and *after* values). Before you make the actual changes, make sure you examine the proposed changes thoroughly to avoid job name collision.

IMPORTANT: Before `migrateTemplates.pl` is invoked, the `admin` user must log into `ectool` using the following command and enter the password when prompted:

```
ectool --server xyz login admin
```

1. Go to the `$INSTALL_DIRECTORY/bin` directory.
2. To invoke the migration tool, enter the following command:

```
ec-perl migrateTemplates.pl --server <server name> [options]
```

where `--server <server name>` specifies the ElectricFlow server to migrate. The default server is *localhost*.

The following options are available:

Option	Description
<code>--counters <server project></code>	Specifies whether job counters are server-wide or project-specific. Default: project .
<code>--dryRun <1 0></code>	<ul style="list-style-type: none">• 1 – Shows the prospective changes in a file that you can examine. Verify the changes before the migration tool makes them.• 0 – Actually makes the changes. Default: 1.
<code>--project <project></code>	Limits migration to the specified project.
<code>--startingJobId <n></code>	Initializes the job counter to the specified number. Default: the current greatest job ID.
<code>--startingWorkflowId <n></code>	Initializes the workflow counter to the specified number. Default: the current greatest workflow ID.

3. View the results.

The migration tool runs successfully when the script completes and output appears showing the changes that were made.

The tool fails when an error message appears.

Example 1

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter across the entire ElectricFlow system and want the next job number to be 1001:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000
```

The above example will change `$(jobId)` to `$(/increment /server/jobCounter)` and initialize the job counter to 1000. This means the next job run on the server will be number 1001. The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000 --dryRun 0
```

Example 2

Before `migrateTemplates.pl` is invoked, the `admin` user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter for each project and want the counters to continue from the current greatest job ID.

```
ec-perl migrateTemplates.pl --server Server1 --counters project
```

The above example will change `$(jobId)` to `[/increment /myJob/project/jobCounter]` and use the current greatest job ID as the initial value for the job counter.

The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters project --dryRun 0
```

Step 3.3: Prepare for the Upgrade

Review the following information before you upgrade the ElectricFlow software.

Upgrade Testing

In most implementations, ElectricFlow is being used in an environment that affects many users. We recommend that you test your upgrade on a separate test server to understand all aspects of the upgrade process. This minimizes the potential impacts to downstream users.

We also strongly recommend that you make a copy of your database and walk through the database upgrade process on a test environment before performing the actual database upgrade on your production environment. The result of this is to understand the impact of performing the database upgrade in your environment and to understand the effect that the upgrade has on your environment. If you need a test license to perform such a test, contact support@electric-cloud.com to request a temporary license.

Backing Up Your Existing ElectricFlow Data

If you are upgrading an ElectricFlow server, it is *extremely* important that you back up your existing ElectricFlow data *before* upgrading. For more information about backups, see [ElectricFlow Server Backups](#) on page 11-1.

- Always back up the Plugins Directory. The default location is the `plugins` subdirectory within the data directory.
- Always back up the files that contain your configuration and custom settings. To ensure that all important settings are saved, back up the following subdirectories in `<data_dir>`:
 - The entire `conf` subdirectory (which contains the ElectricFlow server and agent configuration files)
 - Apache web server configuration files in the `apache/conf` subdirectory
 - The local MySQL database configuration file, `my.ini`, in the `mysql` subdirectory (if applicable)
- Always back up any other files where you have created custom configurations, specified other custom information, or created any type of modification.

Note: The number of ElectricFlow files you may have modified is too numerous to list, so you may want to back up the entire data directory and other miscellaneous files that may have changed.

- If you use an artifact repository, back up your ElectricFlow repository configuration files in the `conf/repository` subdirectory.
- Determine if any changes were made to the custom editor or preflight driver script properties (installed by default). Back up those files if changes were made.

These properties are stored in the server-level property sheet, which can be viewed in the web UI by accessing the **Administration > Server** tab.

Custom editors are stored in the nested sheet named `ec_customEditors`. Preflight driver scripts are stored in the nested sheet named `ec_preflight`. The upgrade process overwrites default custom editor and preflight driver scripts with current versions. We recommend backing up any custom properties you created by renaming the property. For example, change `ec_preflight/clientDrivers/perforce` to `ec_preflight/clientDrivers/perforce_modified`.

Upgrade Installer Preservation

After you back up your ElectricFlow server, create a folder where you can download the `ElectricFlow-<version>` installation file.

MySQL Upgrades

Although the background upgrader does the bulk of the conversion of the MySQL server database from ElectricCommander 4.2.x to ElectricFlow 7.x, the remaining database conversion occurs during the ElectricFlow upgrade. For a large database, this might take many hours.

IMPORTANT: After starting the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, do not interrupt the upgrade process.

- If the upgrade process is interrupted, you might corrupt your database. You have to restore your database from a previous backup.

The following applies when the database is updated:

- Upgrading saves the existing local MySQL database, if applicable.
- Use the `ectool` to view the upgrade progress. On a command line, enter `ectool getServerStatus`.

External MySQL Database Characteristics

When upgrading ElectricFlow, ensure that `default-storage-engine=INNODB` is in the `[mysqld]` section in `my.cnf/my.ini`.

An install and upgrade log file named `installer.log` is created in the `logs` subdirectory in the data directory.

Step 3.4: Shut Down the Source (Original) Server

1. Shut down the ElectricCommander 4.2.x server.

Step 3.5: Run the Installer

To run the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Note: Although the ElectricFlow installer is in `$INSTALL_DIRECTORY/src`, do not launch it from there.

1. Review the information about the following methods:

These are the various installation methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms on page 2-1](#) and [Supported Agent Platforms on page 2-10](#).

User Interface Method

This method provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Method

This method provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is available only for Linux platforms.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version are not be removed or modified and remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Method

This method provides a non-interactive command-line upgrade for supported server platforms. You might prefer this installation method when upgrading multiple remote agents and servers.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server with an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall the software. You can install the repository server on a different machine to avoid uninstalling and reinstalling the software.

Stand-Alone Repository Server or Web Server Upgrade

You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting with ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server.

For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and the "Run the Installer" step for each use case.

Note: ElectricFlow 7.x web servers must use an ElectricFlow 5.x agent to retrieve log files to be displayed in the web interface. This *workspace proxy* agent is typically set to *localhost* but can be modified with the config.php file.

Non-Standard Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent. You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

2. Select a method:

User Interface Upgrade Method

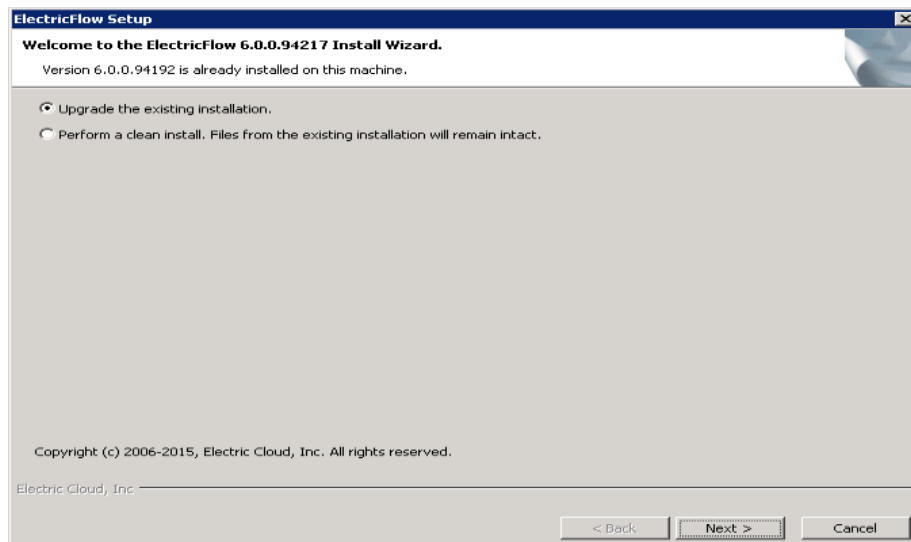
Use the following procedure to upgrade the ElectricFlow software. Review the "Prepare for the Database Upgrade" step in the use before performing this procedure.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 7.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database, if applicable.

1. Open the installer and double-click the `ElectricFlow-<version>` file to begin the installation process.

The Welcome to the ElectricFlow Installer screen appears.

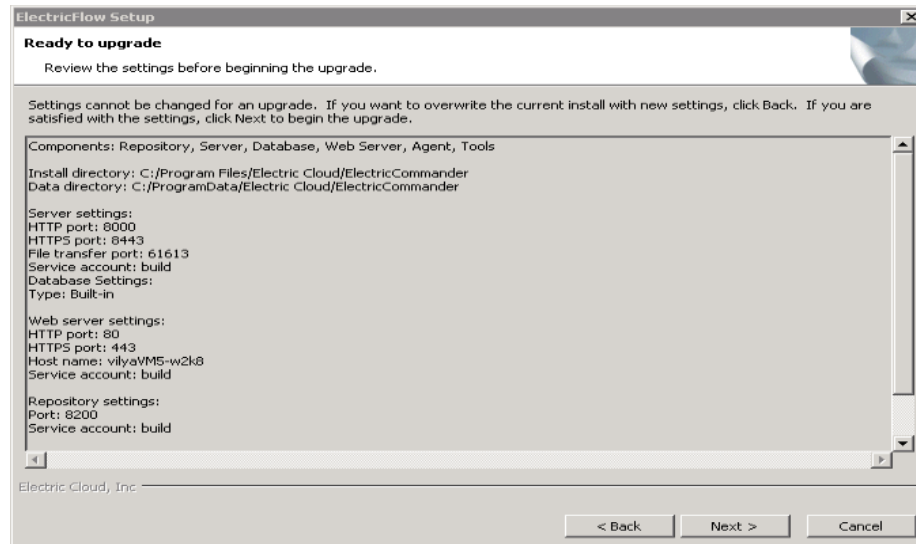


2. Choose one of the following options:
 - Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
 - Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next**.
4. Provide your **New Database Name** (non-Oracle) or **New Database Username** (Oracle) and the corresponding **Password** and click **Next** to continue.
5. Click **Next** to upgrade the existing installation.

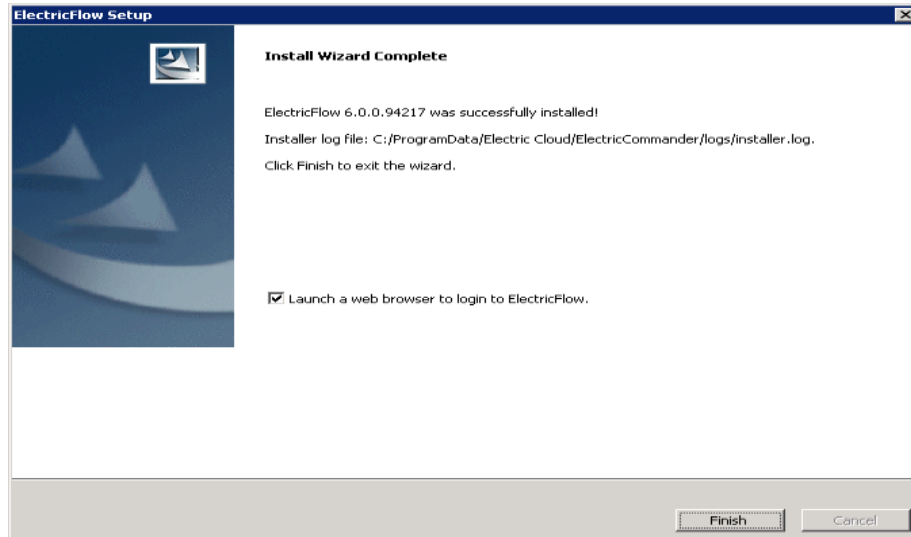
The Ready to Install screen appears.



6. Review the upgrade settings. Use the Back button to change your selections if necessary.
7. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. This process can take up to fifteen minutes to complete. After this process is complete, the new

ElectricFlow version is installed.

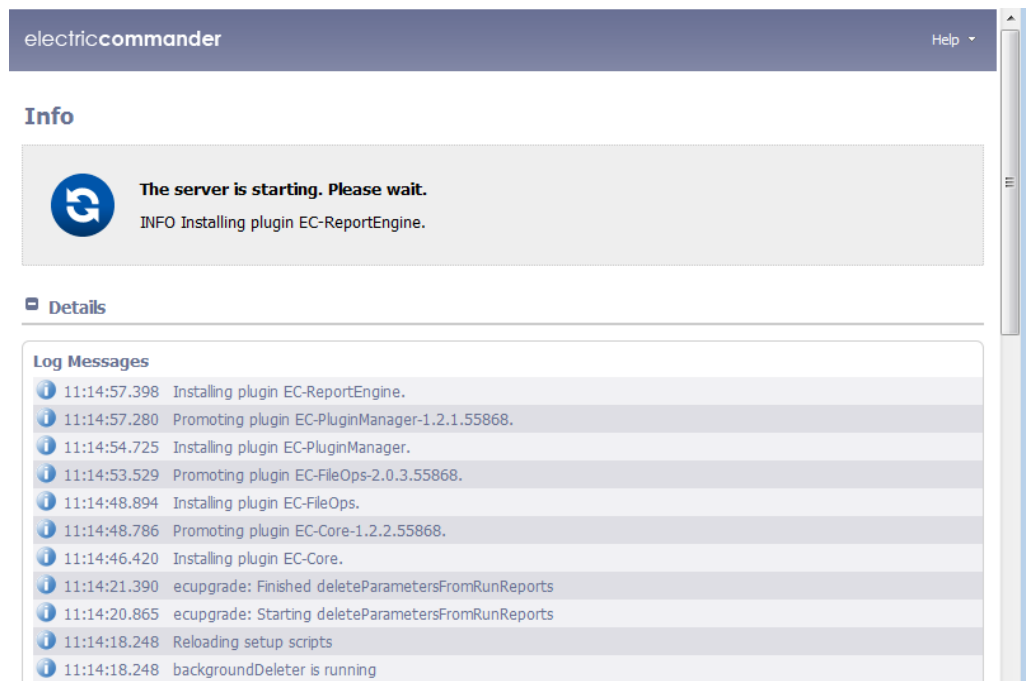


8. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
9. Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line: `ectool getServerStatus`.

After clicking **Finish**, you might see a web page similar to the following screen if the upgrade

is still in progress.



Interactive Command-line Upgrade Method

Use the following procedure to complete the command-line upgrade of a Linux platform. Review the "Preparing for the Database Upgrade" step in the use case before performing this procedure.

When you upgrade ElectricFlow from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database.

1. Change directories to \$INSTALL_DIRECTORY/src.
2. Go to the command line and enter one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer automatically brings up the graphical user interface. To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
This will install ElectricFlow on your computer. Continue? [n/Y]
```

3. Enter `y`.

The system displays the following message:

```
Upgrade the existing <version> installation to version
<version>? [n/Y]
```

4. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter: `y`.
- Enter `n` to exit the installer.

The system displays the following message:

For a non-Oracle database: New database name and database password.

For an Oracle database: New database username and database password.

Then the system displays the following message:

```
Installing agent...
Installing server...
Copied log file to "/opt/electriccloud/electriccommander/logs"
ElectricFlow <version> was successfully installed!
Installer log file: /opt/electriccloud/electriccommander/logs/installer
.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in silent (unattended) mode with no user interface for either Windows or Linux.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to ElectricFlow 7.x saves the existing database.

To upgrade ElectricFlow in silent mode:

1. Connect your ElectricFlow server to the database.
2. Go to the command line and enter one of the following commands from a command line to begin the silent upgrade (for non-Oracle and Oracle databases, respectively):

```
./ElectricFlow-<version> --mode silent --upgradeNewDBName <newdatabasename> --upgradeDBPassword <password> --haveRunMigrationUtility
```

```
./ElectricFlow-<version> --mode silent --upgradeNewDBUserName <newdatabaseusername> --upgradeDBPassword <password> --haveRunMigrationUtility
```

where `<newdatabasename>` is the name of the new database, `<newdatabaseusername>` is the name of database user account in `database.properties`, `<password>` is the password for the database user account in `database.properties`, and `haveRunMigrationUtility` confirms that you have run the migration tool. The database user name and password in `database.properties` may be different than the ones used for the background upgrader.

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (`<datadir>/repository-data`) is:

- **UNIX** - `/opt/electriccloud/electriccommander/repository-data`
- **Windows** - `C:\ProgramData\Electric Cloud\ElectricCommander\repository-data`

Use Case 4: Small Database Upgrade on New Hardware

Note that this use case omits the background upgrader step, which is normally used only for large databases or where downtime is an issue.

If you are installing on new hardware while upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x, perform the following steps.

Step 4.1: Download and Extract the Upgrade Bundle

You use the upgrade bundle (`upgrade-bundle.zip`) to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x.

The `upgrade-bundle.zip` file has these files:

- `README.txt`: README file for Windows and Linux
- `migrateTemplates.pl`: Migration tool to convert `$(jobId)` and `$(workflowId)` to `jobCounter` and `workflowCounter` in the `jobNameTemplate` and `workflowName` template.
- `commander-upgrade-jar-with-dependencies.jar`: Background upgrader to update the database. You can run this tool or the background upgrader script before installing the latest ElectricFlow software.

- `commander-upgrade.pl`: Background upgrader script to update the database. You run this script or the background upgrader tool before installing the latest ElectricFlow software.
- `ElectricFlow-<version>`: ElectricFlow installer to install the latest software version.

The Windows installer for ElectricFlow 7.x is `ElectricFlow-<version>.exe`.

The corresponding Linux installer is `ElectricFlow-<version>`.

IMPORTANT: Only use the background upgrader when upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x. Do not use it when you are updating ElectricFlow 5.x to a later ElectricFlow version.

Perform these steps to download and extract the upgrade bundle:

1. Download the upgrade bundle (`upgrade-bundle.zip`) from the Electric Cloud site.
2. Extract the following files into the `$INSTALL_DIRECTORY` directory:
 - **README file:**
Put `README.txt` into any `$INSTALL_DIRECTORY` directory.
 - **Migration tool for UUIDs:**
Put `migrateTemplates.pl` into the `$INSTALL_DIRECTORY/bin` directory.
 - **Background upgrader:**
Put `commander-upgrade.pl` into the `$INSTALL_DIRECTORY/bin` directory.
Put `commander-upgrade-jar-with-dependencies.jar` into the `$INSTALL_DIRECTORY/Utils` directory.
3. Extract the `ElectricFlow-<version>` installer into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Linux Example:

If the default directory `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/`:

- Put `README.txt` into any `/opt/electriccloud/electriccommander` directory.
- Put `migrateTemplates.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `/opt/electriccloud/electriccommander/Utils` directory.
- Put `ElectricFlow-<version>` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Windows Example:

If the default directory `$INSTALL_DIRECTORY` is `C:\Program Files\Electric Cloud\ElectricCommander\`:

- Put `README.txt` into any `C:\Program Files\Electric Cloud\ElectricCommander` directory.
- Put `migrateTemplates.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.

- Put `commander-upgrade.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `C:\Program Files\Electric Cloud\ElectricCommander\utils` directory.
- Put `ElectricFlow-<version>.exe` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `C:\temp` directory.

Step 4.2: Run the Migration Tool to Convert `$(jobID)` and `$(workflowId)` References

This section describes how to convert the IDs for entity types to automatically-generated universally unique identifiers (UUIDs), 36-character strings. The most common entity changes that affect you are the `jobId` and `workflowId` properties.

The migration tool is an ec-perl script called *migrateTemplates.pl* that converts `jobNameTemplate` and *workflowNameTemplate* properties containing `$(jobId)` and `$(workflowId)` to use a counter instead (by converting those references to `jobCounter` and `workflowCounter` respectively).

This section describes how to convert `$(jobID)` and `$(workflowId)` references in your projects to `jobCounter` and `workflowCounter` references respectively. By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing *before* and *after* values). Before you make the actual changes, make sure you examine the proposed changes thoroughly to avoid job name collision.

IMPORTANT: Before `migrateTemplates.pl` is invoked, the `admin` user must log into ectool using the following command and enter the password when prompted:

```
ectool --server xyz login admin
```

1. Go to the `$INSTALL_DIRECTORY/bin` directory.
2. To invoke the migration tool, enter the following command:

```
ec-perl migrateTemplates.pl --server <server name> [options]
```

where `--server <server name>` specifies the ElectricFlow server to migrate. The default server is *localhost*.

The following options are available:

Option	Description
<code>--counters <server project></code>	Specifies whether job counters are server-wide or project-specific. Default: project .
<code>--dryRun <1 0></code>	<ul style="list-style-type: none"> • 1 – Shows the prospective changes in a file that you can examine. Verify the changes before the migration tool makes them. • 0 – Actually makes the changes. Default: 1.
<code>--project <project></code>	Limits migration to the specified project.
<code>--startingJobId <n></code>	Initializes the job counter to the specified number. Default: the current greatest job ID.
<code>--startingWorkflowId <n></code>	Initializes the workflow counter to the specified number. Default: the current greatest workflow ID.

3. View the results.

The migration tool runs successfully when the script completes and output appears showing the changes that were made.

The tool fails when an error message appears.

Example 1

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter across the entire ElectricFlow system and want the next job number to be 1001:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000
```

The above example will change `$(jobId)` to `$(/increment /server/jobCounter)` and initialize the job counter to 1000. This means the next job run on the server will be number 1001. The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000 --dryRun 0
```

Example 2

Before `migrateTemplates.pl` is invoked, the `admin` user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter for each project and want the counters to continue from the current greatest job ID.

```
ec-perl migrateTemplates.pl --server Server1 --counters project
```

The above example will change `$(jobId)` to `[/increment /myJob/project/jobCounter]` and use the current greatest job ID as the initial value for the job counter.

The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters project --dryRun 0
```

Step 4.3: Prepare for the Upgrade

Review the following information before you upgrade the ElectricFlow software.

Upgrade Testing

In most implementations, ElectricFlow is being used in an environment that affects many users. We recommend that you test your upgrade on a separate test server to understand all aspects of the upgrade process. This minimizes the potential impacts to downstream users.

We also strongly recommend that you make a copy of your database and walk through the database upgrade process on a test environment before performing the actual database upgrade on your production environment. The result of this is to understand the impact of performing the database upgrade in your environment and to understand the effect that the upgrade has on your environment. If you need a test license to perform such a test, contact support@electric-cloud.com to request a temporary license.

Backing Up Your Existing ElectricFlow Data

If you are upgrading an ElectricFlow server, it is *extremely* important that you back up your existing ElectricFlow data *before* upgrading. For more information about backups, see [ElectricFlow Server Backups](#) on page 11-1.

- Always back up the Plugins Directory. The default location is the `plugins` subdirectory within the data directory.
- Always back up the files that contain your configuration and custom settings. To ensure that all important settings are saved, back up the following subdirectories in `<data_dir>`:
 - The entire `conf` subdirectory (which contains the ElectricFlow server and agent configuration files)
 - Apache web server configuration files in the `apache/conf` subdirectory
 - The local MySQL database configuration file, `my.ini`, in the `mysql` subdirectory (if applicable)
- Always back up any other files where you have created custom configurations, specified other custom information, or created any type of modification.

Note: The number of ElectricFlow files you may have modified is too numerous to list, so you may want to back up the entire data directory and other miscellaneous files that may have changed.

- If you use an artifact repository, back up your ElectricFlow repository configuration files in the `conf/repository` subdirectory.
- Determine if any changes were made to the custom editor or preflight driver script properties (installed by default). Back up those files if changes were made.

These properties are stored in the server-level property sheet, which can be viewed in the web UI by accessing the **Administration > Server** tab.

Custom editors are stored in the nested sheet named `ec_customEditors`. Preflight driver scripts are stored in the nested sheet named `ec_preflight`. The upgrade process overwrites default custom editor and preflight driver scripts with current versions. We recommend backing up any custom properties you created by renaming the property. For example, change `ec_preflight/clientDrivers/perforce` to `ec_preflight/clientDrivers/perforce_modified`.

Upgrade Installer Preservation

After you back up your ElectricFlow server, create a folder where you can download the `ElectricFlow-<version>` installation file.

MySQL Upgrades

Although the background upgrader does the bulk of the conversion of the MySQL server database from ElectricCommander 4.2.x to ElectricFlow 7.x, the remaining database conversion occurs during the ElectricFlow upgrade. For a large database, this might take many hours.

IMPORTANT: After starting the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, do not interrupt the upgrade process.

- If the upgrade process is interrupted, you might corrupt your database. You have to restore your database from a previous backup.

The following applies when the database is updated:

- Upgrading saves the existing local MySQL database, if applicable.
- Use the `ectool` to view the upgrade progress. On a command line, enter `ectool getServerStatus`.

External MySQL Database Characteristics

When upgrading ElectricFlow, ensure that `default-storage-engine=INNODB` is in the `[mysqld]` section in `my.cnf/my.ini`.

An install and upgrade log file named `installer.log` is created in the `logs` subdirectory in the data directory.

Step 4.4: Run the Installer on the New Server

To run the installer to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x:

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not

need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Note: Although the ElectricFlow installer is in `$INSTALL_DIRECTORY/src`, do not launch it from there.

1. Review the information about the following methods:

These are the various installation methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms on page 2-1](#) and [Supported Agent Platforms on page 2-10](#).

User Interface Method

This method provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Method

This method provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is available only for Linux platforms.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version are not be removed or modified and remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Method

This method provides a non-interactive command-line upgrade for supported server platforms. You might prefer this installation method when upgrading multiple remote agents and servers.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server with an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall the software. You can install the repository server on a different machine to avoid uninstalling and reinstalling the software.

Stand-Alone Repository Server or Web Server Upgrade

You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting with ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server.

For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and the "Run the Installer" step for each use case.

Note: ElectricFlow 7.x web servers must use an ElectricFlow 5.x agent to retrieve log files to be displayed in the web interface. This *workspace proxy* agent is typically set to *localhost* but can be modified with the config.php file.

Non-Standard Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent. You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

2. Select a method:

User Interface Upgrade Method

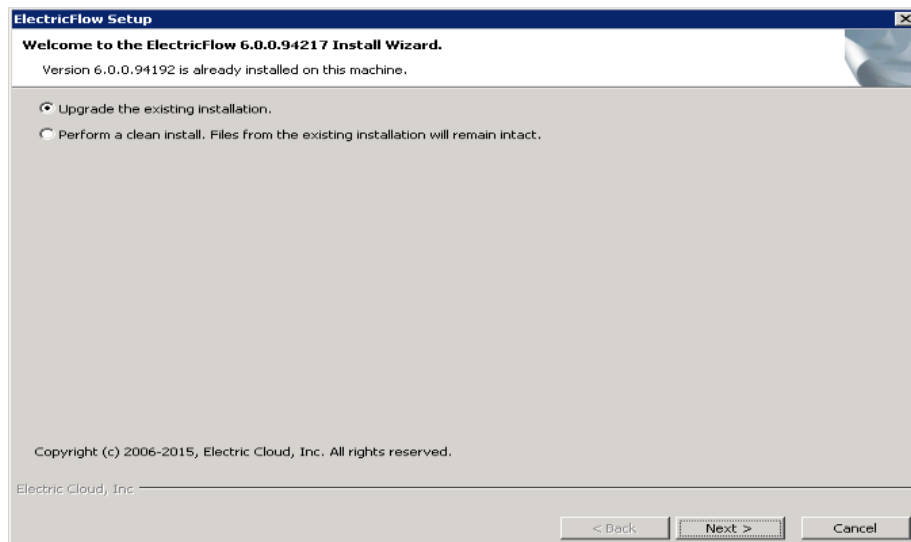
Use the following procedure to upgrade the ElectricFlow software. Review the "Prepare for the Database Upgrade" step in the use before performing this procedure.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 7.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database, if applicable.

1. Open the installer and double-click the `ElectricFlow-<version>` file to begin the installation process.

The Welcome to the ElectricFlow Installer screen appears.



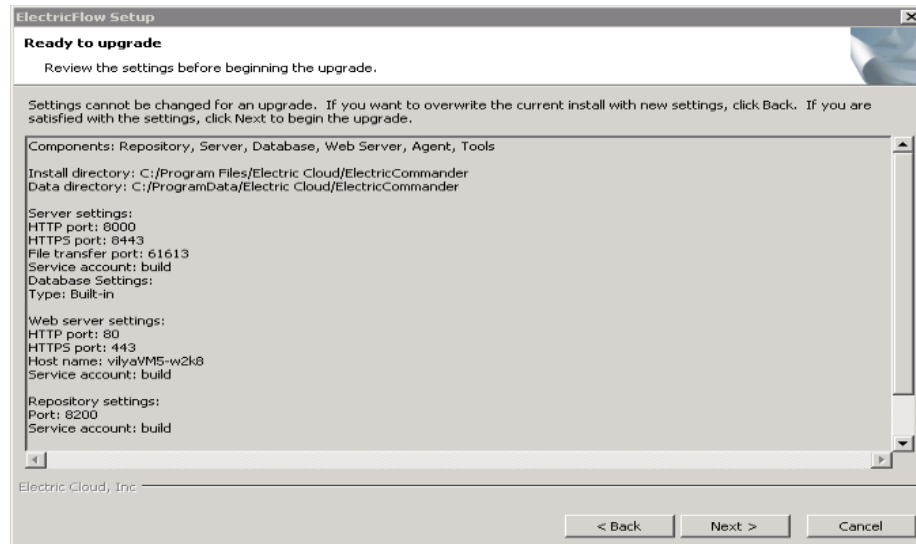
2. Choose one of the following options:

- Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
- Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next**.
4. Provide your **New Database Name** (non-Oracle) or **New Database Username** (Oracle) and the corresponding **Password** and click **Next** to continue.
5. Click **Next** to upgrade the existing installation.

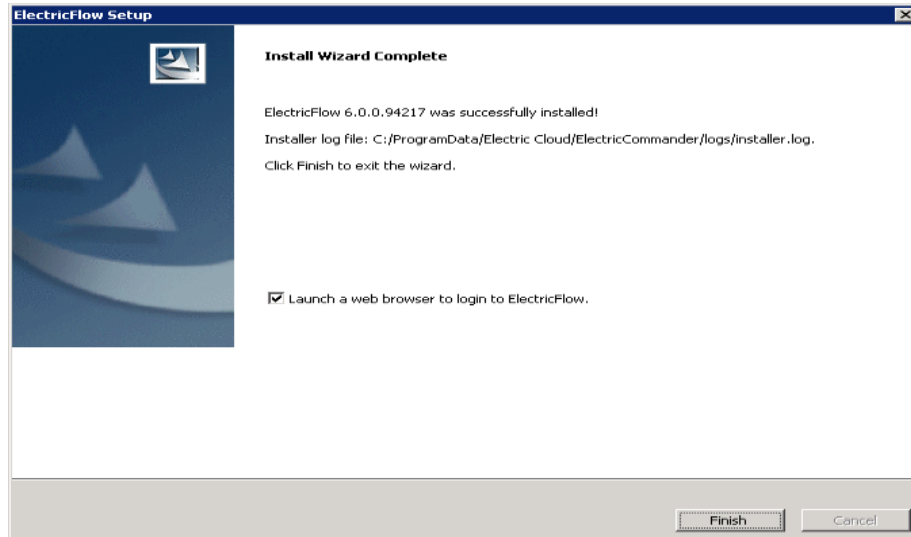
The Ready to Install screen appears.



6. Review the upgrade settings. Use the Back button to change your selections if necessary.
7. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. This process can take up to fifteen minutes to complete. After this process is complete, the new

ElectricFlow version is installed.

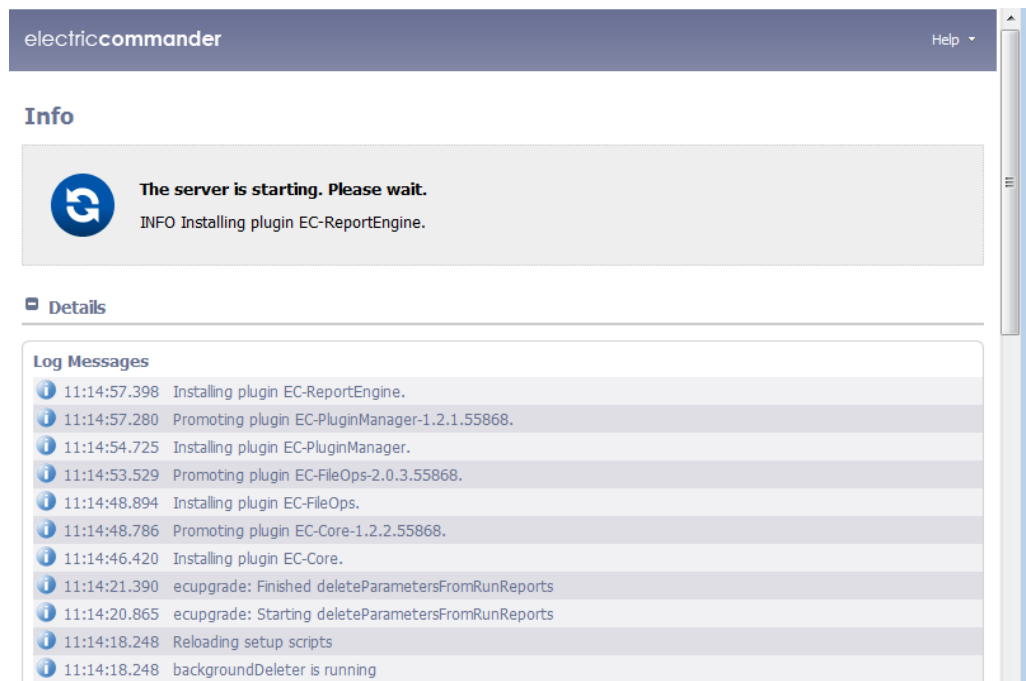


8. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
9. Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line: `ectool getServerStatus`.

After clicking **Finish**, you might see a web page similar to the following screen if the upgrade

is still in progress.



Interactive Command-line Upgrade Method

Use the following procedure to complete the command-line upgrade of a Linux platform. Review the "Preparing for the Database Upgrade" step in the use case before performing this procedure.

When you upgrade ElectricFlow from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database.

1. Change directories to `$INSTALL_DIRECTORY/src`.
2. Go to the command line and enter one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer automatically brings up the graphical user interface. To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
This will install ElectricFlow on your computer. Continue? [n/Y]
```

3. Enter `y`.

The system displays the following message:

```
Upgrade the existing <version> installation to version
<version>? [n/Y]
```

4. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter: `y`.
- Enter `n` to exit the installer.

The system displays the following message:

For a non-Oracle database: New database name and database password.

For an Oracle database: New database username and database password.

Then the system displays the following message:

```
Installing agent...
Installing server...
Copied log file to "/opt/electriccloud/electriccommander/logs"
ElectricFlow <version> was successfully installed!
Installer log file: /opt/electriccloud/electriccommander/logs/installer
.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in silent (unattended) mode with no user interface for either Windows or Linux.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to ElectricFlow 7.x saves the existing database.

To upgrade ElectricFlow in silent mode:

1. Connect your ElectricFlow server to the database.
2. Go to the command line and enter one of the following commands from a command line to begin the silent upgrade (for non-Oracle and Oracle databases, respectively):

```
./ElectricFlow-<version> --mode silent --upgradeNewDBName <newdatabasename> --upgradeDBPassword <password> --haveRunMigrationUtility
```

```
./ElectricFlow-<version> --mode silent --upgradeNewDBUserName <newdatabaseusername> --upgradeDBPassword <password> --haveRunMigrationUtility
```

where `<newdatabasename>` is the name of the new database, `<newdatabaseusername>` is the name of database user account in `database.properties`, `<password>` is the password for the database user account in `database.properties`, and `haveRunMigrationUtility` confirms that you have run the migration tool. The database user name and password in `database.properties` may be different than the ones used for the background upgrader.

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (`<datadir>/repository-data`) is:

- **UNIX** - `/opt/electriccloud/electriccommander/repository-data`
- **Windows** - `C:\ProgramData\Electric Cloud\ElectricCommander\repository-data`

Step 4.5: Shut Down the Existing Server

1. Shut down the ElectricCommander 4.2.x server.

Step 4.6: Complete the Upgrade

1. Point the new ElectricFlow 7.x server to the new target (ElectricFlow 7.x) database, either through the UI or entering `ectool setDatabaseConfiguration`.

Use Case 5: Built-In Database Upgrade on Existing Hardware

Note: In ElectricFlow 5.0, the built-in database changed from H2 to HSQLDB (also referred to as HyperSQL Database).

If you are using the built-in database in ElectricCommander 4.2.x and want to continue using the built-in database in ElectricFlow 7.x, perform the following steps:

Step 5.1: Download and Extract the Upgrade Bundle

You use the upgrade bundle (`upgrade-bundle.zip`) to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x.

The `upgrade-bundle.zip` file has these files:

- `README.txt`: README file for Windows and Linux
- `migrateTemplates.pl`: Migration tool to convert `$(jobId)` and `$(workflowId)` to `jobCounter` and `workflowCounter` in the `jobNameTemplate` and `workflowName` template.
- `commander-upgrade-jar-with-dependencies.jar`: Background upgrader to update the database. You can run this tool or the background upgrader script before installing the latest ElectricFlow software.
- `commander-upgrade.pl`: Background upgrader script to update the database. You run this script or the background upgrader tool before installing the latest ElectricFlow software.
- `ElectricFlow-<version>`: ElectricFlow installer to install the latest software version.

The Windows installer for ElectricFlow 7.x is `ElectricFlow-<version>.exe`.

The corresponding Linux installer is `ElectricFlow-<version>`.

IMPORTANT: Only use the background upgrader when upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x. Do not use it when you are updating ElectricFlow 5.x to a later ElectricFlow version.

Perform these steps to download and extract the upgrade bundle:

1. Download the upgrade bundle (`upgrade-bundle.zip`) from the Electric Cloud site.
2. Extract the following files into the `$INSTALL_DIRECTORY` directory:
 - README file:
Put `README.txt` into any `$INSTALL_DIRECTORY` directory.
 - Migration tool for UUIDs:
Put `migrateTemplates.pl` into the `$INSTALL_DIRECTORY/bin` directory.
 - Background upgrader:
Put `commander-upgrade.pl` into the `$INSTALL_DIRECTORY/bin` directory.
Put `commander-upgrade-jar-with-dependencies.jar` into the `$INSTALL_DIRECTORY/Utils` directory.
3. Extract the `ElectricFlow-<version>` installer into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Linux Example:

If the default directory `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/`:

- Put `README.txt` into any `/opt/electriccloud/electriccommander` directory.
- Put `migrateTemplates.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `/opt/electriccloud/electriccommander/Utils` directory.
- Put `ElectricFlow-<version>` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Windows Example:

If the default directory `$INSTALL_DIRECTORY` is `C:\Program Files\Electric Cloud\ElectricCommander\`:

- Put `README.txt` into any `C:\Program Files\Electric Cloud\ElectricCommander` directory.
- Put `migrateTemplates.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `C:\Program Files\Electric Cloud\ElectricCommander\utils` directory.
- Put `ElectricFlow-<version>.exe` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `C:\temp` directory.

Step 5.2: Run the Migration Tool to Convert `$(jobID)` and `$(workflowId)` References

This section describes how to convert the IDs for entity types to automatically-generated universally unique identifiers (UUIDs), 36-character strings. The most common entity changes that affect you are the `jobId` and `workflowId` properties.

The migration tool is an ec-perl script called *migrateTemplates.pl* that converts `jobNameTemplate` and *workflowNameTemplate* properties containing `$(jobId)` and `$(workflowId)` to use a counter instead (by converting those references to `jobCounter` and `workflowCounter` respectively).

This section describes how to convert `$(jobID)` and `$(workflowId)` references in your projects to `jobCounter` and `workflowCounter` references respectively. By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing *before* and *after* values). Before you make the actual changes, make sure you examine the proposed changes thoroughly to avoid job name collision.

IMPORTANT: Before `migrateTemplates.pl` is invoked, the `admin` user must log into `ectool` using the following command and enter the password when prompted:

```
ectool --server xyz login admin
```

1. Go to the `$INSTALL_DIRECTORY/bin` directory.
2. To invoke the migration tool, enter the following command:

```
ec-perl migrateTemplates.pl --server <server name> [options]
```

where `--server <server name>` specifies the ElectricFlow server to migrate. The default server is *localhost*.

The following options are available:

Option	Description
<code>--counters <server project></code>	Specifies whether job counters are server-wide or project-specific. Default: project .
<code>--dryRun <1 0></code>	<ul style="list-style-type: none"> • 1 – Shows the prospective changes in a file that you can examine. Verify the changes before the migration tool makes them. • 0 – Actually makes the changes. Default: 1.
<code>--project <project></code>	Limits migration to the specified project.
<code>--startingJobId <n></code>	Initializes the job counter to the specified number. Default: the current greatest job ID.
<code>--startingWorkflowId <n></code>	Initializes the workflow counter to the specified number. Default: the current greatest workflow ID.

3. View the results.

The migration tool runs successfully when the script completes and output appears showing the changes that were made.

The tool fails when an error message appears.

Example 1

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter across the entire ElectricFlow system and want the next job number to be 1001:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000
```

The above example will change `$(jobId)` to `$(/increment /server/jobCounter)` and initialize the job counter to 1000. This means the next job run on the server will be number 1001. The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000 --dryRun 0
```


Example 2

Before `migrateTemplates.pl` is invoked, the `admin` user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter for each project and want the counters to continue from the current greatest job ID.

```
ec-perl migrateTemplates.pl --server Server1 --counters project
```

The above example will change `$(jobId)` to `[/increment /myJob/project/jobCounter]` and use the current greatest job ID as the initial value for the job counter.

The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters project --dryRun 0
```

Step 5.3: Create a Database Backup of the ElectricCommander 4.2.x Database

1. Create a complete XML database backup of your ElectricCommander 4.2.x database.

Review the following methods and determine which method is appropriate for your environment:

- We recommend that you run `ectool export` on an inactive system to prevent this process from failing due to conflicts, which may be created in a running system.
- You can also create a backup of your ElectricCommander 4.2.x database by exporting without jobs to limit the time the system is inactive. However, the backup will not have history data.
- See [Data Backup Methods on page 11-1](#) and the “Using `ectool` and the ElectricFlow API” help topic in the ElectricFlow online help system for more information about this process.

Step 5.4: Shut Down the Source (Original) Server

1. Shut down the ElectricCommander 4.2.x server.

Step 5.5: Run the Installer

To run the installer to upgrade an ElectricFlow 7.x server that includes the built-in database from ElectricCommander 4.2.x to ElectricFlow 7.x:

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not

need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Note: Although the ElectricFlow installer is in `$INSTALL_DIRECTORY/src`, do not launch it from there.

1. Review the information about the following methods:

These are the various installation methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms on page 2-1](#) and [Supported Agent Platforms on page 2-10](#).

User Interface Method

This method provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Method

This method provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is available only for Linux platforms.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version are not be removed or modified and remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Method

This method provides a non-interactive command-line upgrade for supported server platforms. You might prefer this installation method when upgrading multiple remote agents and servers.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server with an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall the software. You can install the repository server on a different machine to avoid uninstalling and reinstalling the software.

Stand-Alone Repository Server or Web Server Upgrade

You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting with ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server.

For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and the "Run the Installer" step for each use case.

Note: ElectricFlow 7.x web servers must use an ElectricFlow 5.x agent to retrieve log files to be displayed in the web interface. This *workspace proxy* agent is typically set to *localhost* but can be modified with the config.php file.

Non-Standard Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent. You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

2. Select a method:

User Interface Upgrade Method

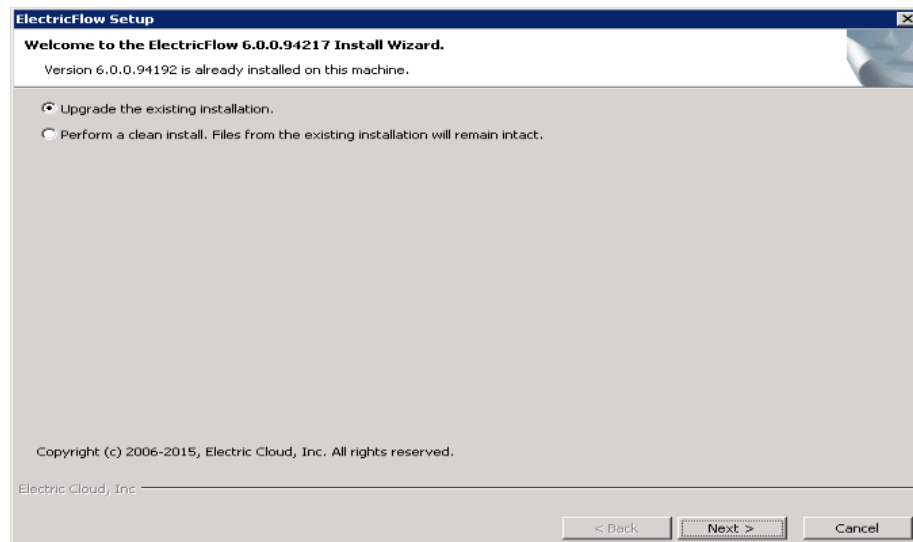
Use the following procedure to upgrade the ElectricFlow software. Review the "Prepare for the Database Upgrade" step in the use before performing this procedure.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 7.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database, if applicable.

1. Open the installer and double-click the `ElectricFlow-<version>` file to begin the installation process.

The Welcome to the ElectricFlow Installer screen appears.

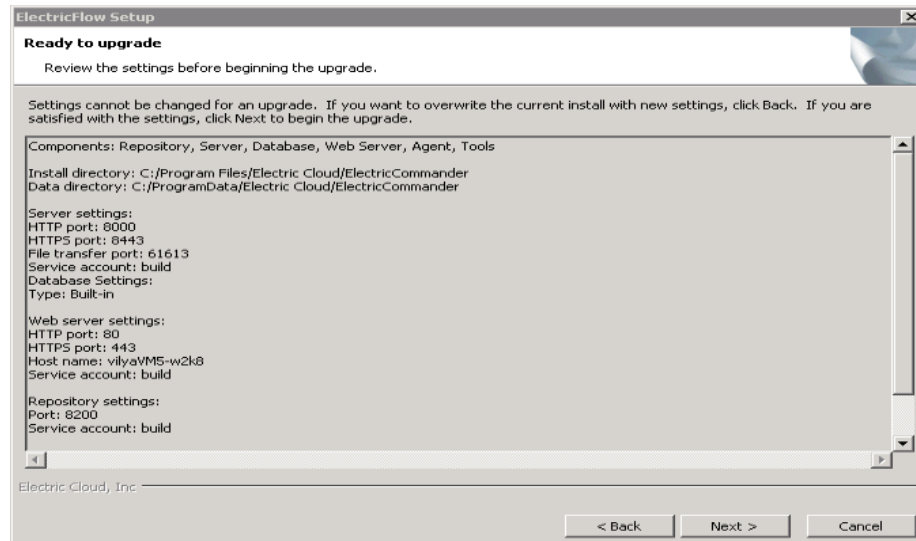


2. Choose one of the following options:
 - Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
 - Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next**.
4. Provide your **New Database Name** (non-Oracle) or **New Database Username** (Oracle) and the corresponding **Password** and click **Next** to continue.
5. Click **Next** to upgrade the existing installation.

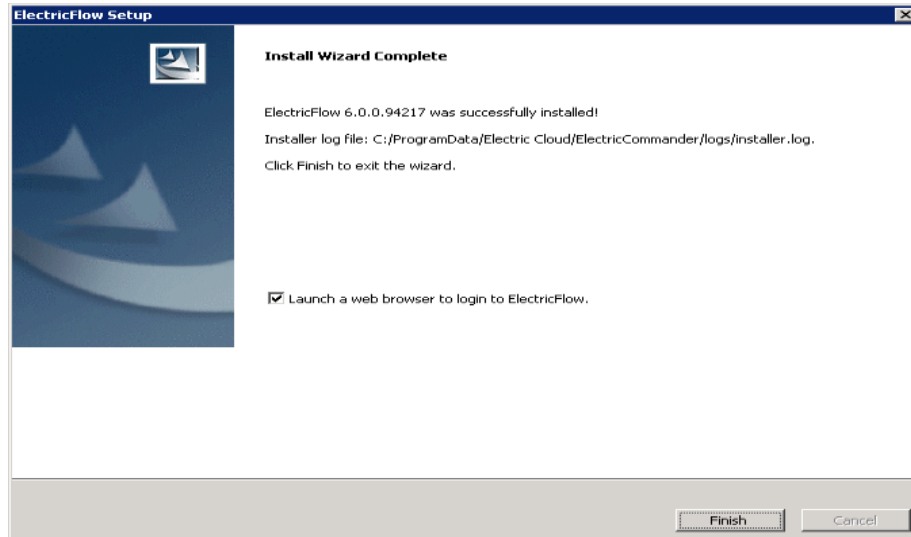
The Ready to Install screen appears.



6. Review the upgrade settings. Use the Back button to change your selections if necessary.
7. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. This process can take up to fifteen minutes to complete. After this process is complete, the new

ElectricFlow version is installed.

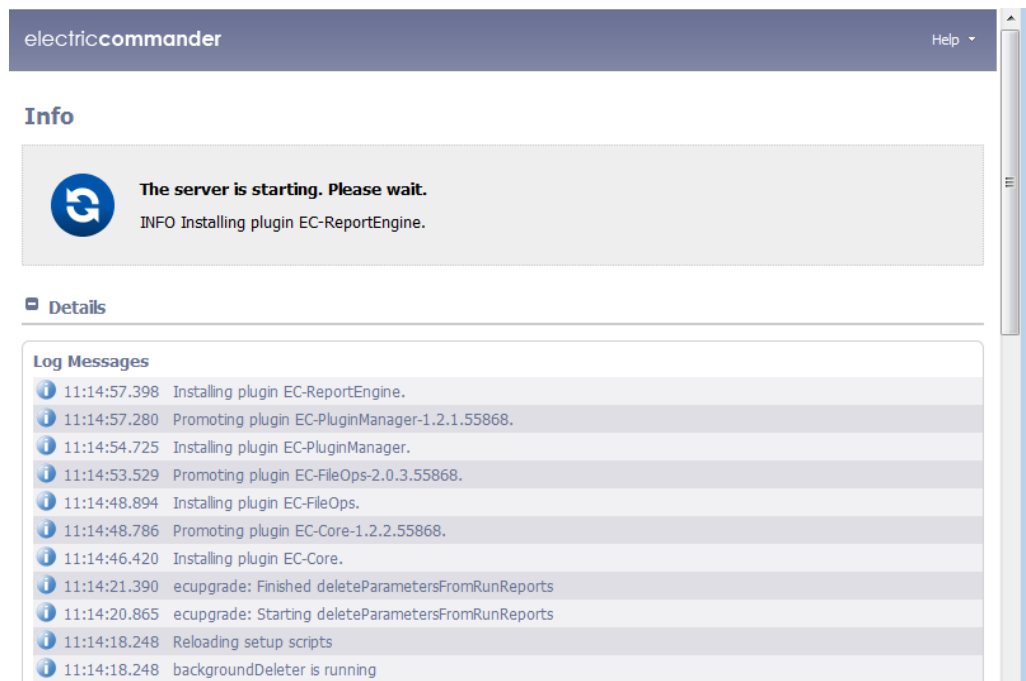


8. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
9. Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line: `ectool getServerStatus`.

After clicking **Finish**, you might see a web page similar to the following screen if the upgrade

is still in progress.



Interactive Command-line Upgrade Method

Use the following procedure to complete the command-line upgrade of a Linux platform. Review the "Preparing for the Database Upgrade" step in the use case before performing this procedure.

When you upgrade ElectricFlow from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database.

1. Change directories to \$INSTALL_DIRECTORY/src.
2. Go to the command line and enter one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer automatically brings up the graphical user interface. To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
This will install ElectricFlow on your computer. Continue? [n/Y]
```


3. Enter `y`.

The system displays the following message:

```
Upgrade the existing <version> installation to version
<version>? [n/Y]
```

4. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter: `y`.
- Enter `n` to exit the installer.

The system displays the following message:

For a non-Oracle database: New database name and database password.

For an Oracle database: New database username and database password.

Then the system displays the following message:

```
Installing agent...
Installing server...
Copied log file to "/opt/electriccloud/electriccommander/logs"
ElectricFlow <version> was successfully installed!
Installer log file: /opt/electriccloud/electriccommander/logs/installer
.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in silent (unattended) mode with no user interface for either Windows or Linux.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to ElectricFlow 7.x saves the existing database.

To upgrade ElectricFlow in silent mode:

1. Connect your ElectricFlow server to the database.
2. Go to the command line and enter one of the following commands from a command line to begin the silent upgrade (for non-Oracle and Oracle databases, respectively):

```
./ElectricFlow-<version> --mode silent --upgradeNewDBName <newdatabasename> --upgradeDBPassword <password> --haveRunMigrationUtility
```

```
./ElectricFlow-<version> --mode silent --upgradeNewDBUserName <newdatabaseusername> --upgradeDBPassword <password> --haveRunMigrationUtility
```

where <newdatabasename> is the name of the new database, <newdatabaseusername> is the name of database user account in database.properties, <password> is the password for the database user account in database.properties, and haveRunMigrationUtility confirms that you have run the migration tool. The database user name and password in database.properties may be different than the ones used for the background upgrader.

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (<datadir>/repository-data) is:

- **UNIX** - /opt/electriccloud/electriccommander/repository-data
- **Windows** - C:\ProgramData\Electric Cloud\ElectricCommander\repository-data

Step 5.6: Import the ElectricCommander 4.2.x Database into ElectricFlow 7.x

1. Enter `ectool --server <server name> login "<user name>" "<password>"` to log in to the ElectricFlow 6.x server.

For example: `ectool --server bldglserver login "Ellen Ernst" "ee123"`

2. Enter `ectool import <path to file>/<database filename>`.

Use Case 6: Built-In Database Upgrade on New Hardware

If you are installing on new hardware while upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x, perform the following steps.

Step 6.1: Download and Extract the Upgrade Bundle to the Existing Server

You use the upgrade bundle (upgrade-bundle.zip) to upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x.

The upgrade-bundle.zip file has these files:

- `README.txt`: README file for Windows and Linux
- `migrateTemplates.pl`: Migration tool to convert `$(jobId)` and `$(workflowId)` to `jobCounter` and `workflowCounter` in the `jobNameTemplate` and `workflowName` template.
- `commander-upgrade-jar-with-dependencies.jar`: Background upgrader to update the database. You can run this tool or the background upgrader script before installing the latest ElectricFlow software.
- `commander-upgrade.pl`: Background upgrader script to update the database. You run this script or the background upgrader tool before installing the latest ElectricFlow software.
- `ElectricFlow-<version>`: ElectricFlow installer to install the latest software version.

The Windows installer for ElectricFlow 7.x is `ElectricFlow-<version>.exe`.

The corresponding Linux installer is `ElectricFlow-<version>`.

IMPORTANT: Only use the background upgrader when upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x. Do not use it when you are updating ElectricFlow 5.x to a later ElectricFlow version.

Perform these steps to download and extract the upgrade bundle:

1. Download the upgrade bundle (`upgrade-bundle.zip`) from the Electric Cloud site.
2. Extract the following files into the `$INSTALL_DIRECTORY` directory:
 - README file:
Put `README.txt` into any `$INSTALL_DIRECTORY` directory.
 - Migration tool for UUIDs:
Put `migrateTemplates.pl` into the `$INSTALL_DIRECTORY/bin` directory.
 - Background upgrader:
Put `commander-upgrade.pl` into the `$INSTALL_DIRECTORY/bin` directory.
Put `commander-upgrade-jar-with-dependencies.jar` into the `$INSTALL_DIRECTORY/Utils` directory.
3. Extract the `ElectricFlow-<version>` installer into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Linux Example:

If the default directory `$INSTALL_DIRECTORY` is `/opt/electriccloud/electriccommander/`:

- Put `README.txt` into any `/opt/electriccloud/electriccommander` directory.
- Put `migrateTemplates.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade.pl` into the `/opt/electriccloud/electriccommander/bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `/opt/electriccloud/electriccommander/Utils` directory.
- Put `ElectricFlow-<version>` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `/tmp` directory.

Windows Example:

If the default directory `$INSTALL_DIRECTORY` is `C:\Program Files\Electric Cloud\ElectricCommander\`:

- Put `README.txt` into any `C:\Program Files\Electric Cloud\ElectricCommander` directory.
- Put `migrateTemplates.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade.pl` into the `C:\Program Files\Electric Cloud\ElectricCommander\bin` directory.
- Put `commander-upgrade-jar-with-dependencies.jar` into the `C:\Program Files\Electric Cloud\ElectricCommander\utils` directory.
- Put `ElectricFlow-<version>.exe` into any directory except the `$INSTALL_DIRECTORY` directory, such as the `C:\temp` directory.

Step 6.2: Run the Migration Tool to Convert `$(jobID)` and `$(workflowId)` References

This section describes how to convert the IDs for entity types to automatically-generated universally unique identifiers (UUIDs), 36-character strings. The most common entity changes that affect you are the `jobId` and `workflowId` properties.

The migration tool is an ec-perl script called *migrateTemplates.pl* that converts `jobNameTemplate` and *workflowNameTemplate* properties containing `$(jobId)` and `$(workflowId)` to use a counter instead (by converting those references to `jobCounter` and `workflowCounter` respectively).

This section describes how to convert `$(jobID)` and `$(workflowId)` references in your projects to `jobCounter` and `workflowCounter` references respectively. By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing *before* and *after* values). Before you make the actual changes, make sure you examine the proposed changes thoroughly to avoid job name collision.

IMPORTANT: Before `migrateTemplates.pl` is invoked, the `admin` user must log into `ectool` using the following command and enter the password when prompted:

```
ectool --server xyz login admin
```

1. Go to the `$INSTALL_DIRECTORY/bin` directory.
2. To invoke the migration tool, enter the following command:

```
ec-perl migrateTemplates.pl --server <server name> [options]
```

where `--server <server name>` specifies the ElectricFlow server to migrate. The default server is *localhost*.

The following options are available:

Option	Description
<code>--counters <server project></code>	Specifies whether job counters are server-wide or project-specific. Default: project .
<code>--dryRun <1 0></code>	<ul style="list-style-type: none"> • 1 – Shows the prospective changes in a file that you can examine. Verify the changes before the migration tool makes them. • 0 – Actually makes the changes. Default: 1.
<code>--project <project></code>	Limits migration to the specified project.
<code>--startingJobId <n></code>	Initializes the job counter to the specified number. Default: the current greatest job ID.
<code>--startingWorkflowId <n></code>	Initializes the workflow counter to the specified number. Default: the current greatest workflow ID.

3. View the results.

The migration tool runs successfully when the script completes and output appears showing the changes that were made.

The tool fails when an error message appears.

Example 1

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter across the entire ElectricFlow system and want the next job number to be 1001:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000
```

The above example will change `$(jobId)` to `$(/increment /server/jobCounter)` and initialize the job counter to 1000. This means the next job run on the server will be number 1001. The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000 --dryRun 0
```

Example 2

Before `migrateTemplates.pl` is invoked, the `admin` user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter for each project and want the counters to continue from the current greatest job ID.

```
ec-perl migrateTemplates.pl --server Server1 --counters project
```

The above example will change `$(jobId)` to `$(/increment /myJob/project/jobCounter)` and use the current greatest job ID as the initial value for the job counter.

The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters project --dryRun 0
```

This section describes how to convert `$(jobID)` and `$(workflowId)` references in your projects to `jobCounter` and `workflowCounter` references respectively. By default, the migration tool runs in *dry run* mode. Dry run mode lists the changes the script would make (showing *before* and *after* values). Before you make the actual changes, make sure you examine the proposed changes thoroughly to avoid job name collision.

IMPORTANT: Before `migrateTemplates.pl` is invoked, the `admin` user must log into `ectool` using the following command and enter the password when prompted:

```
ectool --server xyz login admin
```

1. Go to the `$INSTALL_DIRECTORY/bin` directory.
2. To invoke the migration tool, enter the following command:

```
ec-perl migrateTemplates.pl --server <server name> [options]
```

where `--server <server name>` specifies the ElectricFlow server to migrate. The default server is *localhost*.

The following options are available:

Option	Description
<code>--counters <server project></code>	Specifies whether job counters are server-wide or project-specific. Default: project .
<code>--dryRun <1 0></code>	<ul style="list-style-type: none"> • 1 – Shows the prospective changes in a file that you can examine. Verify the changes before the migration tool makes them. • 0 – Actually makes the changes. Default: 1.
<code>--project <project></code>	Limits migration to the specified project.
<code>--startingJobId <n></code>	Initializes the job counter to the specified number. Default: the current greatest job ID.
<code>--startingWorkflowId <n></code>	Initializes the workflow counter to the specified number. Default: the current greatest workflow ID.

3. View the results.

The migration tool runs successfully when the script completes and output appears showing the changes that were made.

The tool fails when an error message appears.

Example 1

Before `migrateTemplates.pl` is invoked, the admin user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter across the entire ElectricFlow system and want the next job number to be 1001:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000
```

The above example will change `$(jobId)` to `$(/increment /server/jobCounter)` and initialize the job counter to 1000. This means the next job run on the server will be number 1001. The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters server --startingJobId 1000 --dryRun 0
```

Example 2

Before `migrateTemplates.pl` is invoked, the `admin` user logs into `ectool` using the `ectool --server xyz login admin` command and enters the password.

You want a unique job counter for each project and want the counters to continue from the current greatest job ID.

```
ec-perl migrateTemplates.pl --server Server1 --counters project
```

The above example will change `$(jobId)` to `[/increment /myJob/project/jobCounter]` and use the current greatest job ID as the initial value for the job counter.

The following actually makes the proposed changes:

```
ec-perl migrateTemplates.pl --server Server1 --counters project --dryRun 0
```

Step 6.3: Create a Database Backup of the ElectricCommander 4.2.x Database

1. Create a complete XML database backup of your ElectricCommander 4.2.x database.

Review the following methods and determine which method is appropriate for your environment:

- We recommend that you run `ectool export` on an inactive system to prevent this process from failing due to conflicts, which may be created in a running system.
- You can also create a backup of your ElectricCommander 4.2.x database by exporting without jobs to limit the time the system is inactive. However, the backup will not have history data.
- See [Data Backup Methods on page 11-1](#) and the “Using `ectool` and the ElectricFlow API” help topic in the ElectricFlow online help system for more information about this process.

Step 6.4: Install ElectricFlow 6.x on the New Server

To install ElectricFlow 7.x on the new server:

IMPORTANT: When installing ElectricFlow on RHEL 6.x:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Red Hat. The ElectricFlow installer needs these libraries to run successfully.

When you run the one of these installers on a 64-bit RHEL 6.x environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit RHEL 6.x machines:

- Run the `yum install libstdc++.i686` command. If you do not run this command, the ElectricFlow Apache server will not start and the installer silently fails for any type of ElectricFlow installation.

- Run the `yum install libuuid.i686` command if you are performing an ElectricFlow installation that includes an Apache server. If you are installing ElectricFlow agents only, without a web server, you do not need to run this command on each agent machine.

- Run the `yum install nss-pam-ldapd*.i686` command (to install 32-bit NSS packages) if using an LDAP account for ownership of the server/web/repository services. If you do not run this command, the ElectricFlow Apache server fails to start.

IMPORTANT: When installing ElectricFlow on Ubuntu 10.04, Ubuntu 12.04, and Ubuntu 14.04:

For the full installer, 32-bit-agent-only installer, 64-bit-agent-only installer on unsupported platforms, or when installing the 64-bit agent without an internet connection, you must install some 32-bit libraries that were omitted by Ubuntu. The ElectricFlow installer needs these libraries to run successfully.

When you run one of these installers on a 64-bit Ubuntu environment without 32-bit libraries, the ElectricFlow installer exits immediately because it cannot find the required 32-bit libraries. No error message is displayed, and the log file has no information about the error.

To resolve this issue, we recommend running *all* of the following commands on the 64-bit Ubuntu machines before installing ElectricFlow to install Ubuntu's 32-bit compatibility layer:

- On Ubuntu 10.04 and Ubuntu 12.04, enter these commands:

```
sudo apt-get update
sudo apt-get install lib32bz2-1.0
sudo apt-get install libuuid1:i386
```

- On Ubuntu 14.04, enter these commands:

```
sudo apt-get update
sudo dpkg --add-architecture i386
sudo apt-get install lib32bz2-1.0
sudo apt-get update
sudo apt-get install libuuid1:i386
```

- If you plan to use an LDAP account for ownership of the server/web/repository services with 64-bit Ubuntu, you must run `sudo apt-get update && sudo apt-get install libnss-ldap:i386`. This command installs 32-bit NSS packages. If you do not run this command, the ElectricFlow Apache server fails to start.

Note: Although the ElectricFlow installer is in `$INSTALL_DIRECTORY/src`, do not launch it from there.

1. Review the information about the following methods:

These are the various installation methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms on page 2-1](#) and [Supported Agent Platforms on page 2-10](#).

User Interface Method

This method provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Method

This method provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is available only for Linux platforms.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version are not be removed or modified and remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Method

This method provides a non-interactive command-line upgrade for supported server platforms. You might prefer this installation method when upgrading multiple remote agents and servers.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, reruns the background upgrader, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server with an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall the software. You can install the repository server on a different machine to avoid uninstalling and reinstalling the software.

Stand-Alone Repository Server or Web Server Upgrade

You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting with ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server.

For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and the "Run the Installer" step for each use case.

Note: ElectricFlow 7.x web servers must use an ElectricFlow 5.x agent to retrieve log files to be displayed in the web interface. This *workspace proxy* agent is typically set to *localhost* but can be modified with the config.php file.

Non-Standard Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent. You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

2. Select a method:

User Interface Upgrade Method

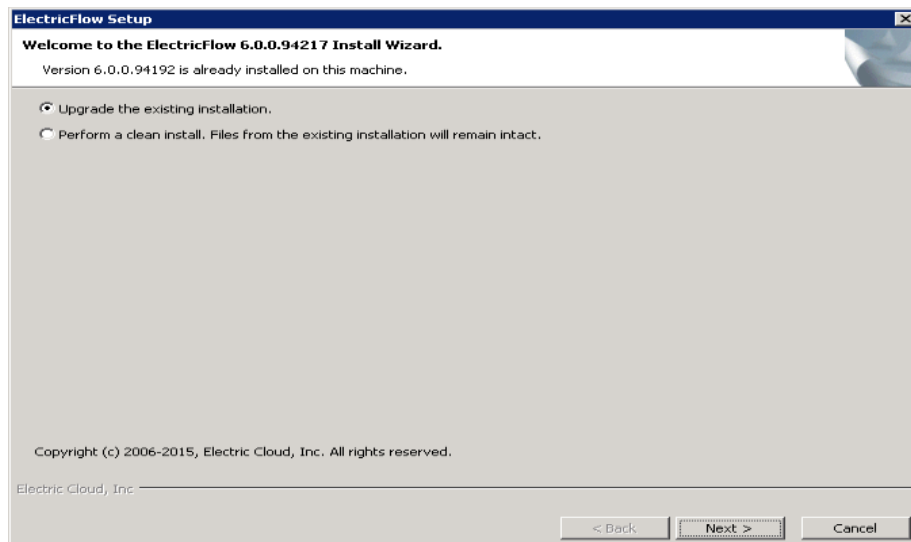
Use the following procedure to upgrade the ElectricFlow software. Review the "Prepare for the Database Upgrade" step in the use before performing this procedure.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 7.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database, if applicable.

1. Open the installer and double-click the `ElectricFlow-<version>` file to begin the installation process.

The Welcome to the ElectricFlow Installer screen appears.

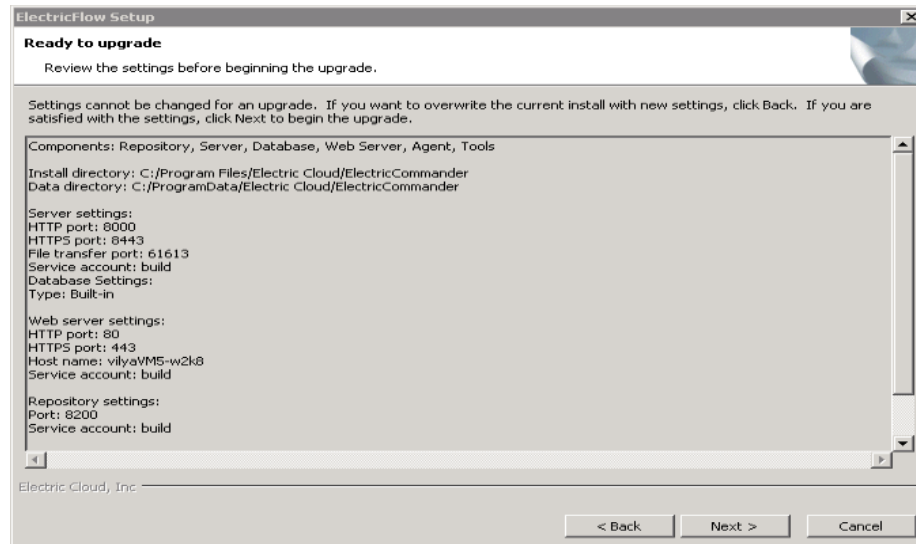


2. Choose one of the following options:
 - Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
 - Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next**.
4. Provide your **New Database Name** (non-Oracle) or **New Database Username** (Oracle) and the corresponding **Password** and click **Next** to continue.
5. Click **Next** to upgrade the existing installation.

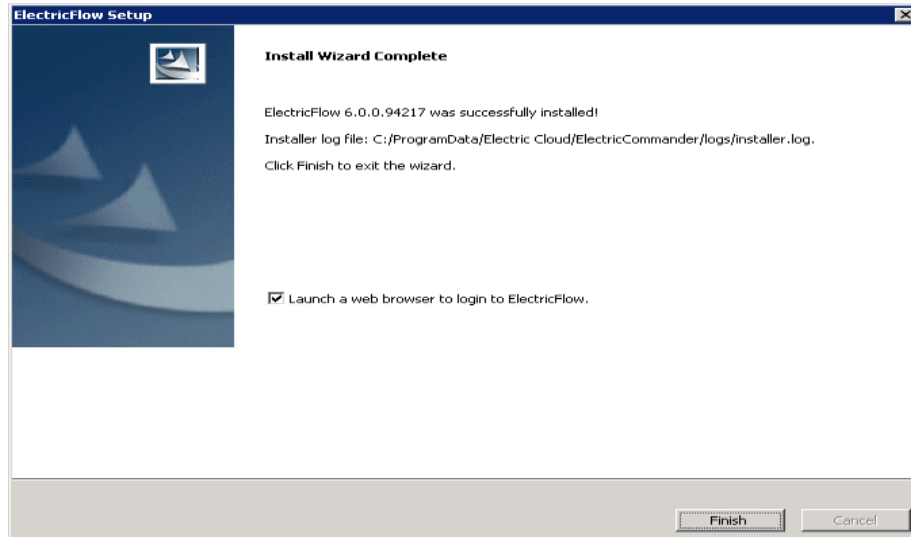
The Ready to Install screen appears.



6. Review the upgrade settings. Use the Back button to change your selections if necessary.
7. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. This process can take up to fifteen minutes to complete. After this process is complete, the new

ElectricFlow version is installed.

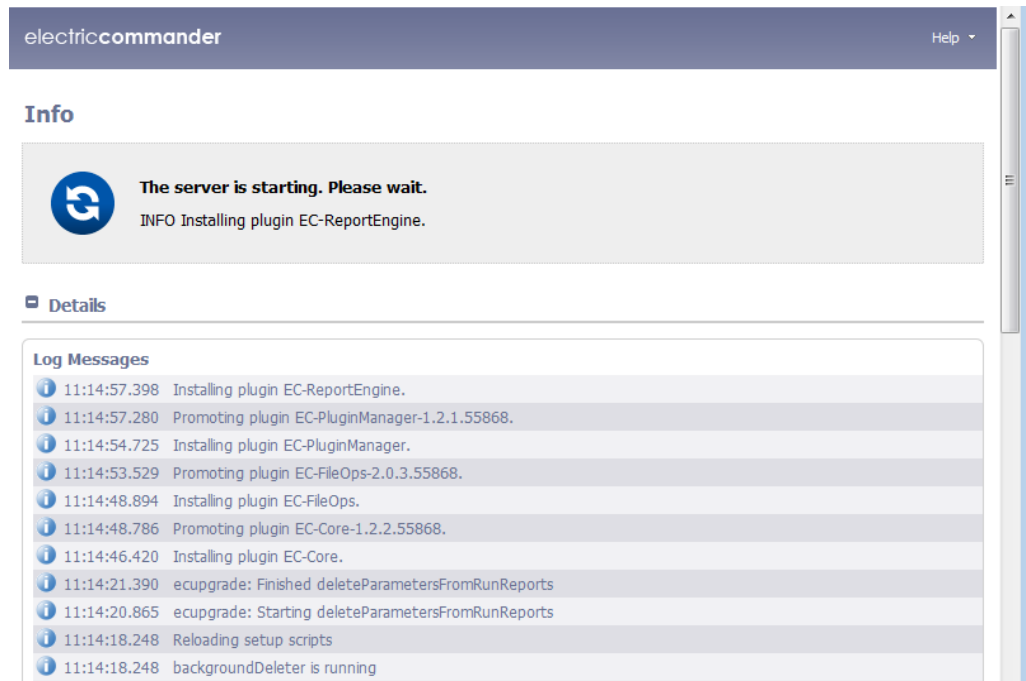


8. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
9. Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line: `ectool getServerStatus`.

After clicking **Finish**, you might see a web page similar to the following screen if the upgrade

is still in progress.



Interactive Command-line Upgrade Method

Use the following procedure to complete the command-line upgrade of a Linux platform. Review the "Preparing for the Database Upgrade" step in the use case before performing this procedure.

When you upgrade ElectricFlow from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to Commander 4.2 or later saves the existing database.

1. Change directories to `$INSTALL_DIRECTORY/src`.
2. Go to the command line and enter one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer automatically brings up the graphical user interface. To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
This will install ElectricFlow on your computer. Continue? [n/Y]
```

3. Enter `y`.

The system displays the following message:

```
Upgrade the existing <version> installation to version
<version>? [n/Y]
```

4. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter: `y`.
- Enter `n` to exit the installer.

The system displays the following message:

For a non-Oracle database: New database name and database password.

For an Oracle database: New database username and database password.

Then the system displays the following message:

```
Installing agent...
Installing server...
Copied log file to "/opt/electriccloud/electriccommander/logs"
ElectricFlow <version> was successfully installed!
Installer log file: /opt/electriccloud/electriccommander/logs/installer
.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in silent (unattended) mode with no user interface for either Windows or Linux.

When you upgrade from ElectricCommander 4.2.x to ElectricFlow 7.x, if your database is large, we recommend that you run the background upgrader and complete the database upgrade before upgrading to ElectricFlow 5.x because the upgrade process requires an extensive database upgrade.

Note: Upgrading to ElectricFlow 7.x saves the existing database.

To upgrade ElectricFlow in silent mode:

1. Connect your ElectricFlow server to the database.
2. Go to the command line and enter one of the following commands from a command line to begin the silent upgrade (for non-Oracle and Oracle databases, respectively):

```
./ElectricFlow-<version> --mode silent --upgradeNewDBName <newdatabasename> --upgradeDBPassword <password> --haveRunMigrationUtility
```

```
./ElectricFlow-<version> --mode silent --upgradeNewDBUserName <newdatabaseusername> --upgradeDBPassword <password> --haveRunMigrationUtility
```

where <newdatabasename> is the name of the new database, <newdatabaseusername> is the name of database user account in database.properties, <password> is the password for the database user account in database.properties, and haveRunMigrationUtility confirms that you have run the migration tool. The database user name and password in database.properties may be different than the ones used for the background upgrader.

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (<datadir>/repository-data) is:

- **UNIX** - /opt/electriccloud/electriccommander/repository-data
- **Windows** - C:\ProgramData\Electric Cloud\ElectricCommander\repository-data

Step 6.5: Shut Down the Existing Server

1. Shut down the ElectricCommander 4.2.x server.

Step 6.6: Import the ElectricCommander 4.2.x Database into ElectricFlow 6.x

1. Enter `ectool --server <server name> login "<user name>" "<password>"` to log in to the ElectricFlow 6.x server.

For example: `ectool --server bldg1server login "Ellen Ernst" "ee123"`

2. Enter `ectool import <path to file>/<database filename>`.

Chapter 8: Use Case 7: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version in a Clustered Environment

This section describes how to upgrade the software from ElectricFlow 5.x or 6.x to latest ElectricFlow version and upgrade cluster configurations at the same time. The procedure is the same as when you upgrade from ElectricFlow 5.x or 6.x to the latest version except that you need to perform additional tasks to upgrade the cluster.

In this section, the term "*ElectricFlow 6.x*" refers to the latest ElectricFlow version. For example, "from ElectricFlow 5.x to ElectricFlow 6.x" means from ElectricFlow 5.x or 6.x (the current version installed in your system) to the latest version (a newer version). In addition, ElectricCommander 5.x is referred to as ElectricFlow 5.x. Anything that refers to ElectricCommander 5.x also refers to ElectricFlow 5.x.

IMPORTANT: This upgrade process applies only to clusters, so it does not apply if your system is running ElectricCommander 4.2.x. You must use the upgrade process described in [Upgrading from ElectricCommander 4.2.x to ElectricFlow 7.x \(Use Cases 1 to 6\)](#) on page 7-1.

IMPORTANT: When upgrading from ElectricCommander 4.2.x to ElectricFlow 6.x, use the migration tool, background upgrader, and installer in the upgrade bundle.

When upgrading from ElectricFlow 5.x to ElectricFlow 6.x, use only the ElectricFlow installer (`ElectricFlow-<version>`), which works the same way as the installers in previous releases. Do not use the migration tool or the background upgrader.

To upgrade from ElectricFlow 5.x to ElectricFlow 6.x, use the `ElectricFlow-<version>` installer, which collects the ElectricFlow service account credentials, uninstalls the current release, installs the latest ElectricFlow release, configures the system with all property values mined, and restores custom files and data.

After preparing for the upgrade, make sure to shut down the ElectricFlow server service before installing ElectricFlow 7.x.

Agent Configuration Settings Preserved After an Upgrade

The following agent configuration settings are saved during a software upgrade. These settings are reloaded into ElectricFlow after the upgrade.

IMPORTANT: Because custom configuration settings will be lost during the upgrade, Electric Cloud recommends that you save them before upgrading.

Properties in `<data_dir>/conf/agent.conf`

port
proto
pluginsPath
artifactCache
logFile
logLevel
logMaxSize
logMaxFiles
duplicateDetectionListSize
unixShellPattern
serverConnectTimeout
serverReadTimeout
outboundRequestInitialRetryInterval
outboundRequestMaxRetryInterval
outboundRequestTimeout
loadProfile
idleWorkerTimeout
idlePostRunnerTimeout
idleServerRequestWorkerTimeout
idleOutboundConnectionTimeout
keyFile
certFile
verifyPeer
caFile
caPath

Properties in `<data_dir>/conf/agent/wrapper.conf`

xwrapper.java.classpath.<n>
wrapper.java.library.path.<n>
#wrapper.java.additional.701
#wrapper.java.additional.702
#wrapper.java.additional.703
#wrapper.java.initmemory.percent

wrapper.java.initmemory
#wrapper.java.maxmemory.percent
wrapper.java.maxmemory
wrapper.console.format
wrapper.logfile
wrapper.logfile.format
wrapper.logfile.loglevel
wrapper.logfile.maxsize
wrapper.logfile.maxfiles
wrapper.syslog.loglevel
wrapper.ntservice.dependency.<n>
wrapper.ntservice.starttype
wrapper.ntservice.interactive
wrapper.request_thread_dump_on_failed_jvm_exit
wrapper.ping.interval
wrapper.ping.timeout
wrapper.startup.timeout
wrapper.shutdown.timeout
wrapper.successful_invocation_time
set.ECWRAPPER_WRITE_MAX_ATTEMPTS
set.ECWRAPPER_WRITE_RETRY_INTERVAL

Properties in <data_dir>/conf/agent/agent.properties

AGENT_PORT
AGENT_LOCAL_PORT
AGENT_PROTOCOL
AGENT_ACCEPT_QUEUE_SIZE
AGENT_DOMAIN_NAME
AGENT_MAX_HTTP_THREADS
AGENT_SERVER_SESSIONS_FILE
MAX_LOGGED_MESSAGE_LENGTH
AGENT_KEYSTORE
AGENT_KEYSTORE_PASSWORD
AGENT_CRL_FILE
OUTBOUND_CONNECT_TIMEOUT
IDLE_CONNECTION_TIMEOUT
MAX_CONNECTIONS

MAX_CONNECTIONS_PER_ROUTE

Use Cases

The actual steps that you perform to upgrade from ElectricFlow 5.x or 6.x to the latest ElectricFlow release are based on your ElectricFlow environment.

Review the information in the following table, and select the use case that best matches your ElectricFlow environment to go to the detailed upgrade process steps.

Use Case	Clustered Environment	Link to the Upgrade Process Steps
7	Yes	Use Case 7: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version in a Clustered Environment on page 8-1
8	No	Use Case 8: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version on page 9-1

Upgrading Clusters

In a cluster configuration, make sure to review the tasks in [Preparing for Your Upgrade](#) on page 8-5. You must make backup copies of these files:

- `httpd.conf` in `<data directory>/apache/conf` subdirectory
- `wrapper.conf` in `<data directory>/conf` subdirectory

Before starting the upgrade process, you must shut down the ElectricFlow server service before installing ElectricFlow 7.x.

During the upgrade process, ElectricFlow is not available after the web server is stopped or after the server service on the last ElectricFlow server node is stopped.

1. Perform either of these actions:

- Stop the ElectricFlow server service on all nodes.
- Stop the ElectricFlow server service on all nodes except on the primary ElectricFlow server node.

You can keep the server service on your primary ElectricFlow server node up at this point, because the installer automatically stops the server service on the primary node when it upgrades that node.

2. On the ElectricFlow server, do the following:
 - a. Upgrade the ElectricFlow server. This will also do the following:
 - Connect the server to the database.
 - Upgrade the plugins.
 - Start the server.
 - b. Check and restore the `wrapper.conf` settings. For example, the settings for the line `"wrapper.java.additional.600="`.
 - c. Restart the ElectricFlow servers service.
3. On the ElectricFlow web server, do the following:
 - a. Upgrade the node on the web server.
 - b. Check and restore the `httpd.conf` settings.
4. Upgrade any nodes on the repository servers.

Your ElectricFlow system is now available.

5. Upgrade the remaining ElectricFlow server nodes.

During the upgrade process, some settings may be lost. Verify the following settings before connecting to the ElectricFlow system:

- `httpd.conf` settings for redirecting—These lines should be commented out:


```
# Redirect http to https
# RewriteCond %{HTTPS} !=on
# RewriteRule ^/commander/(.*) https://%{SERVER_NAME}:443%{REQUEST_URI} [NC,R,L]
```
- `httpd.conf` setting for `COMMANDER_SERVER`—This should point to the load balancer:


```
SetEnv COMMANDER_SERVER "<FQDN of your load balancer>"
```
- `wrapper.conf` contains the line pointing at your Zookeeper instances

For example:

```
wrapper.java.additional.600=-DCOMMANDER_ZK_CONNECTION=192.168.7.20:2181
```

Preparing for Your Upgrade

Review the following information before you upgrade the ElectricFlow software.

Upgrade Testing

In most implementations, ElectricFlow is used in an environment that affects many users. You should test your upgrade on a separate test server to understand all aspects of the upgrade process. This minimizes the potential impact on downstream users.

Backing Up Your Existing ElectricFlow Data

IMPORTANT: Before upgrading an ElectricFlow server, you must back up your existing ElectricFlow data. See [ElectricFlow Server Backups](#) on page 11-1 for more information about backups.

Backing Up Commander Cluster Configuration Files

The configuration files for the Commander cluster are in `<data_dir>\conf`. The default location is:

- (Linux) `/opt/electriccloud/electriccommander/conf/`
- (Windows) `C:\ProgramData\Electric Cloud\ElectricCommander\conf`

Note: Although the Commander cluster configuration files such as `commander.properties`, `database.properties`, `keystore`, and `passkey` are present in one of the directories above, they are not actually used by the cluster during runtime.

These files were uploaded to Apache ZooKeeper from the first node that was clustered as described in the [Uploading Configuration Files to ZooKeeper](#) on page 4-18 section in the "Clustering" chapter. You can download these files from ZooKeeper to a temporary folder and then compare them with those in the `\conf` folder. You can do so by using the ElectricFlow ZKConfigTool, which is discussed in the [Uploading Configuration Files to ZooKeeper](#) on page 4-18 section.

For example, complete the following steps to download these files to `C:\temp` on Windows, where `<install_dir>` is `C:\Program Files\Electric Cloud\ElectricCommander`.

1. Download the files from ZooKeeper by entering the following commands:

```
cd C:\temp

COMMANDER_ZK_CONNECTION=<ZooKeeper_Server_IP>:2181

"C:\Program Files\Electric Cloud\ElectricCommander\jre\bin\java" -jar "C:\Program Files\Electric Cloud\ElectricCommander\server\bin\zk-config-tool-jar-with-dependencies.jar" com.electriccloud.commander.cluster.ZKConfigTool -readFile /commander/conf/database.properties database.properties

"C:\Program Files\Electric Cloud\ElectricCommander\jre\bin\java" -jar "C:\Program Files\Electric Cloud\ElectricCommander\server\bin\zk-config-tool-jar-with-dependencies.jar" com.electriccloud.commander.cluster.ZKConfigTool -readFile /commander/conf/keystore keystore

"C:\Program Files\Electric Cloud\ElectricCommander\jre\bin\java" -jar "C:\Program Files\Electric Cloud\ElectricCommander\server\bin\zk-config-tool-jar-with-dependencies.jar" com.electriccloud.commander.cluster.ZKConfigTool -readFile /commander/conf/passkey passkey

"C:\Program Files\Electric Cloud\ElectricCommander\jre\bin\java" -jar "C:\Program Files\Electric Cloud\ElectricCommander\server\bin\zk-config-tool-jar-with-dependencies.jar" com.electriccloud.commander.cluster.ZKConfigTool -readFile /commander/conf/commander.properties
```

2. Make sure that the four files in `C:\temp` are the same as the ones in `<data_dir>\conf`. (You can use a file diff tool to make this easier.) If any file in `<data_dir>\conf` is different, then back up that file and replace it with the one that you downloaded from ZooKeeper.

Backing Up Other Files

The ElectricFlow files that might have been modified are too numerous to list, so you should back up the entire ElectricFlow data directory and other miscellaneous files that might have changed. But at a minimum, you must back up the following files:

- The plugins directory. The default location is the `plugins` subdirectory within `<data_dir>`.
- Files that contain your configuration and custom settings. To ensure that all important settings are saved, back up the following subdirectories in `<data_dir>`:
 - The entire `conf` subdirectory, which contains the ElectricFlow server and agent configuration files.
 - Apache web server configuration files in the `apache/conf` subdirectory.
 - (If applicable) The local MySQL database configuration file, `my.ini`, in the `mysql` subdirectory.
- Any other files where you created custom configurations, specified other custom information, or created any type of modification.
- (If you use an artifact repository) The ElectricFlow repository configuration files in the `conf/repository` subdirectory.
- (If modified) The custom editor or preflight driver script properties (installed by default).

These properties are stored in the server-level property sheet, which you can view in the web UI in the **Administration > Server** subtab.

Custom editors are stored in the nested sheet named `ec_customEditors`. Preflight driver scripts are stored in the nested sheet named `ec_preflight`. The upgrade process overwrites default custom editor and preflight driver scripts with current versions. You should back up any custom properties that you created by renaming those properties. For example, change `ec_preflight/clientDrivers/perforce` to `ec_preflight/clientDrivers/perforce_modified`.

Upgrade Installer Preservation

After you back up your ElectricFlow server, create a folder where you can download the `ElectricFlow-<version>` installation file.

MySQL Upgrades

ElectricFlow upgrades involving a MySQL database can take several hours to complete if you have a significant data set. *Do not interrupt the upgrade process.* You can corrupt your database if the upgrade process is interrupted. A restore from a previous database backup will be required.

You can use `ectool` to view the upgrade progress. On a command line, enter

```
ectool getServerStatus
```

An install/upgrade log file named `installer.log` is created in the `logs` subdirectory in `<data_dir>`.

Choosing the Correct Upgrade Method

This section describes the various upgrade methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms](#) on page 2-1 and [Supported Agent Platforms](#) on page 2-10.

User Interface Upgrade

This [method](#) provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed. See [User Interface Upgrade Method](#) on page 9-7 for more details.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Upgrade

This [method](#) provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is only available for Linux platforms. See [Interactive Command-Line Upgrade Method](#) on page 9-10 for more details.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Upgrade

This [method](#) provides a non-interactive command-line upgrade for supported server platforms. You may find this installation method preferable for upgrading multiple remote agents and servers. See [Silent \(Unattended\) Upgrade Method](#) on page 9-11 for more details.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server With an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall ElectricFlow. You can install the repository server on a different machine to avoid uninstalling and reinstalling ElectricFlow.

Non-Server Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent (that is, an agent on a machine that is not a supported ElectricFlow server platform). You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

Stand-Alone Repository Server or Web Server Upgrade

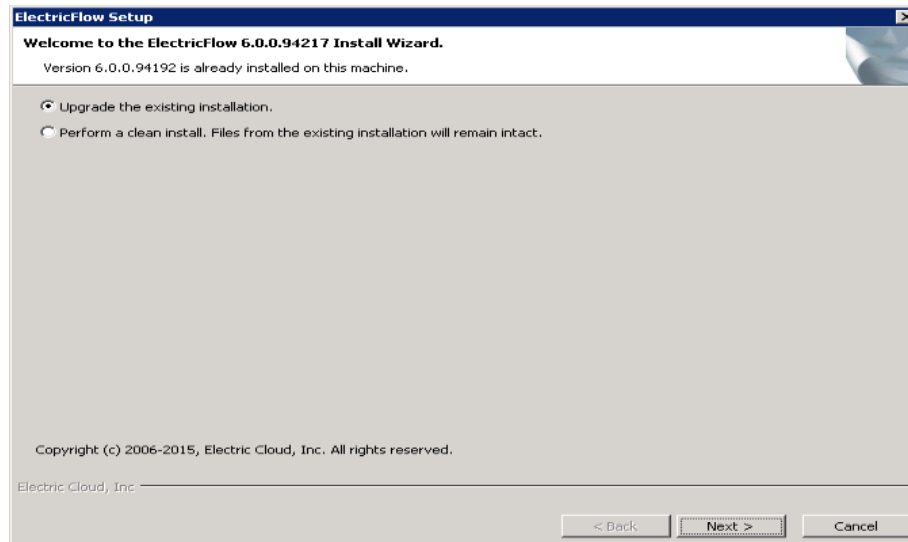
You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting in ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server. For more information, see [Uninstalling ElectricFlow](#) on page 10-1, [Installing ElectricFlow](#) on page 3-1, and [Copying Repository Contents](#) on page 9-11.

User Interface Upgrade Method

Use this procedure to upgrade the ElectricFlow software. Review [Preparing for Your Upgrade](#) on page 9-4 before performing this procedure.

IMPORTANT: This document refers to the installer using the new ElectricFlow product name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, be aware that we intend to stop providing the installer with "ElectricCommander" in the filename after this version.

1. Double-click the `ElectricFlow-<version>` file to begin installation. The Welcome to the ElectricFlow Installer screen appears.



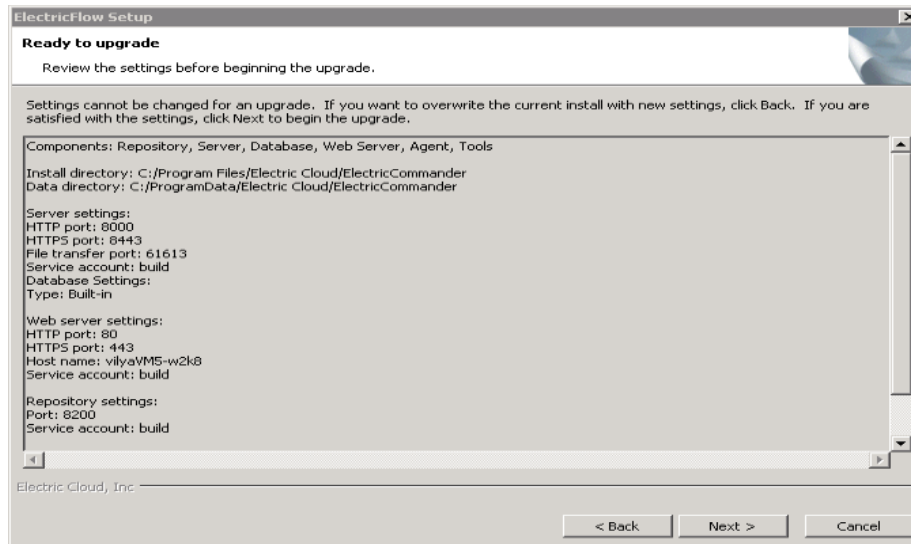
2. Choose one of the following options:

- Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
- Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next** to upgrade the existing installation.

The Ready to Install screen appears.

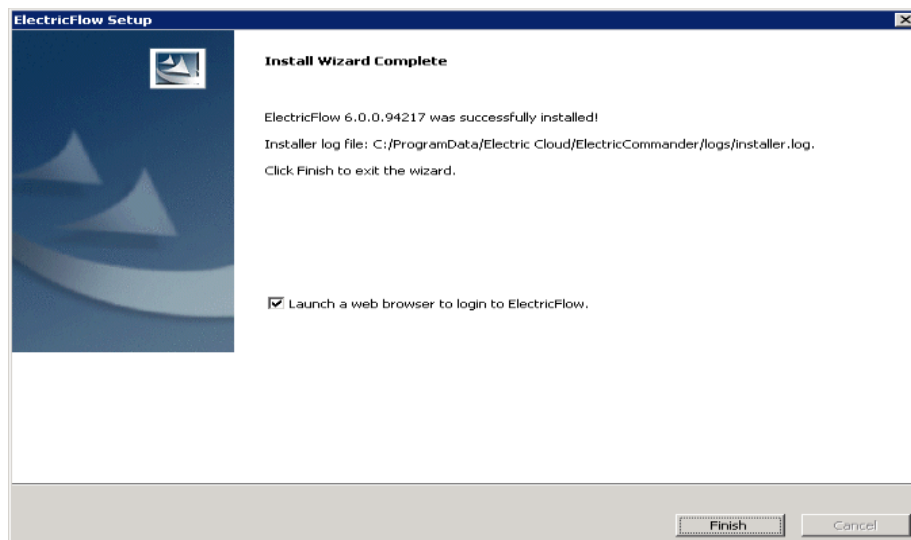


4. Review the upgrade settings.

Use the Back button to change your selections if necessary.

5. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. You can also view the `installer.log` file to see progress. The time that it takes to complete this process depends on the size of the database. It may take fifteen minutes or longer to complete. Once this process is complete, the new ElectricFlow version is installed.



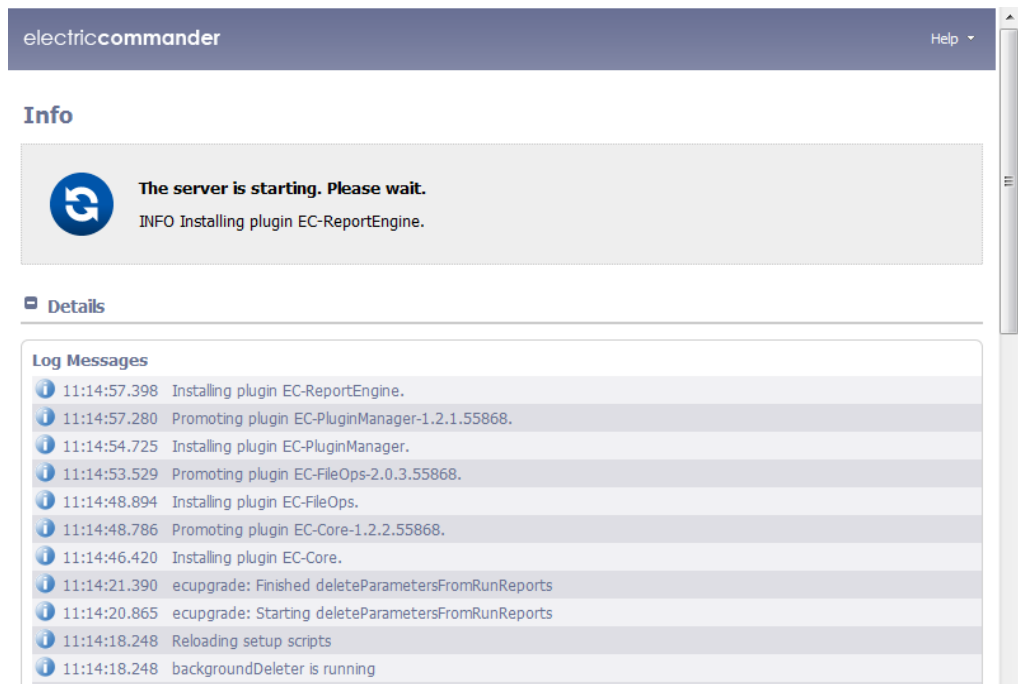
6. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.

- Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line:

```
ectool getServerStatus.
```

After clicking **Finish**, you may see a web page similar to the following screen if the upgrade is still in progress.



Interactive Command-Line Upgrade Method

Use the following procedure to complete a command-line upgrade of a Linux platform. Review [Preparing for Your Upgrade](#) on page 9-4 before performing this procedure.

IMPORTANT: This document refers to the installer using the new ElectricFlow product name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, be aware that we intend to stop providing the installer with "ElectricCommander" in the filename after this version.

- Choose one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface.

To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow on your computer. Continue? [n/Y]
```

2. Enter: `y`

The System displays the following message:

```
Upgrade the existing <version> installation to version <version>? [n/Y]
```

3. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter `y`.
- Enter `n` to exit the installer.

The system displays the following message:

```
Installing agent...
```

```
Installing server...
```

```
Copied log file to "/opt/electriccloud/electriccommander/logs"
```

```
ElectricFlow <version> was successfully installed!
```

```
Installer log file: /opt/electriccloud/electriccommander/logs/installer.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in unattended (silent) mode with no user interface for either Windows or Linux.

IMPORTANT: This document refers to the installer using the new ElectricFlow product name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, be aware that we intend to stop providing the installer with "ElectricCommander" in the filename after this version.

Enter the following command from a command line to begin a silent upgrade:

```
./ElectricFlow-<version> --mode silent
```

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (`<datadir>/repository-data`) is:

- **UNIX** - `/opt/electriccloud/electriccommander/repository-data`
- **Windows** - `C:\ProgramData\Electric Cloud\ElectricCommander\repository-data`

Uploading Configuration Files to ZooKeeper if Needed

After you upgrade the ElectricFlow server node, you must again compare the `<data_dir>\conf\commander.properties` file with the file that you downloaded from ZooKeeper (which you saved to `c:\temp`). To do so, complete the following steps.

1. Open the `<data_dir>\conf\commander.properties` file.
2. Make sure that the `COMMANDER_SERVER_NAME` property is set to `<load_balancer_FQDN>`.
3. If the following line exists, remove it:

```
COMMANDER_MQ_DISK_SPACE_CHECK_FREQUENCY=60
```

4. Check whether the following lines exist. If they do not exist, add them:

```
COMMANDER_CRITICAL_SERVICES_MONITORING_FREQUENCY=60
COMMANDER_CRITICAL_SERVICES_MONITORING_ENABLED=true
COMMANDER_CRITICAL_SERVICES_MAX_ATTEMPTS_TO_BE_IN_PRIMARY_CLUSTER=5
```

Note: Ensure that these properties are not duplicated in the file.

5. Upload the new file to ZooKeeper as described in the [Uploading Configuration Files to ZooKeeper](#) on page 4-18 section in the "Clustering" chapter.

Chapter 9: Use Case 8: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version

This section describes how to upgrade the software from ElectricFlow 5.x or 6.x to the latest ElectricFlow version. The procedure is the same as when you upgrade from ElectricFlow 5.x or 6.x to the latest version and upgrade cluster configurations at the same time, except that you do not need to perform additional tasks to upgrade the cluster.

In this section, the term "*ElectricFlow 6.x*" refers to the latest ElectricFlow version. For example, "from ElectricFlow 5.x to ElectricFlow 6.x" means from ElectricFlow 5.x or 6.x (the current version installed in your system) to the latest version (a newer version). In addition, ElectricCommander 5.x is referred to as ElectricFlow 5.x. Anything that refers to ElectricCommander 5.x also refers to ElectricFlow 5.x.

IMPORTANT: If your system is running a version earlier than ElectricCommander 4.2, you first need to upgrade to ElectricCommander 4.2.x. You should be able to safely upgrade from ElectricCommander 3.10.x, 4.0.x, or 4.1.x to ElectricCommander 4.2.x, and then upgrade to ElectricFlow 7.x.

IMPORTANT: When upgrading from ElectricCommander 4.2.x to ElectricFlow 6.x, use the migration tool, background upgrader, and installer in the upgrade bundle.

When upgrading from ElectricFlow 5.x to ElectricFlow 6.x, use only the ElectricFlow installer (*ElectricFlow-`<version>`*), which works the same way as the installers in previous releases. Do not use the migration tool or the background upgrader.

To upgrade from ElectricFlow 5.x to ElectricFlow 6.x, use the *ElectricFlow-`<version>`* installer, which collects the ElectricFlow service account credentials, uninstalls the current release, installs the latest ElectricFlow release, configures the system with all property values mined, and restores custom files and data.

After preparing for the upgrade, make sure to shut down the ElectricFlow server service before installing ElectricFlow 7.x.

Agent Configuration Settings Preserved After an Upgrade

The following agent configuration settings are saved during a software upgrade. These settings are reloaded into ElectricFlow after the upgrade.

IMPORTANT: Because custom configuration settings will be lost during the upgrade, Electric Cloud recommends that you save them before upgrading.

Properties in <data_dir>/conf/agent.conf

port
proto
pluginsPath
artifactCache
logFile
logLevel
logMaxSize
logMaxFiles
duplicateDetectionListSize
unixShellPattern
serverConnectTimeout
serverReadTimeout
outboundRequestInitialRetryInterval
outboundRequestMaxRetryInterval
outboundRequestTimeout
loadProfile
idleWorkerTimeout
idlePostRunnerTimeout
idleServerRequestWorkerTimeout
idleOutboundConnectionTimeout
keyFile
certFile
verifyPeer
caFile
caPath

Properties in <data_dir>/conf/agent/wrapper.conf

xwrapper.java.classpath.<n>
wrapper.java.library.path.<n>
#wrapper.java.additional.701
#wrapper.java.additional.702
#wrapper.java.additional.703
#wrapper.java.initmemory.percent
wrapper.java.initmemory
#wrapper.java.maxmemory.percent

wrapper.java.maxmemory
wrapper.console.format
wrapper.logfile
wrapper.logfile.format
wrapper.logfile.loglevel
wrapper.logfile.maxsize
wrapper.logfile.maxfiles
wrapper.syslog.loglevel
wrapper.ntservice.dependency.<n>
wrapper.ntservice.starttype
wrapper.ntservice.interactive
wrapper.request_thread_dump_on_failed_jvm_exit
wrapper.ping.interval
wrapper.ping.timeout
wrapper.startup.timeout
wrapper.shutdown.timeout
wrapper.successful_invocation_time
set.ECWRAPPER_WRITE_MAX_ATTEMPTS
set.ECWRAPPER_WRITE_RETRY_INTERVAL

Properties in <data_dir>/conf/agent/agent.properties

AGENT_PORT
AGENT_LOCAL_PORT
AGENT_PROTOCOL
AGENT_ACCEPT_QUEUE_SIZE
AGENT_DOMAIN_NAME
AGENT_MAX_HTTP_THREADS
AGENT_SERVER_SESSIONS_FILE
MAX_LOGGED_MESSAGE_LENGTH
AGENT_KEYSTORE
AGENT_KEYSTORE_PASSWORD
AGENT_CRL_FILE
OUTBOUND_CONNECT_TIMEOUT
IDLE_CONNECTION_TIMEOUT
MAX_CONNECTIONS
MAX_CONNECTIONS_PER_ROUTE

Use Cases

The actual steps that you perform to upgrade from ElectricFlow 5.x or 6.x to the latest ElectricFlow release are based on your ElectricFlow environment.

Review the information in the following table, and select the use case that best matches your ElectricFlow environment to go to the detailed upgrade process steps.

Use Case	Clustered Environment	Link to the Upgrade Process Steps
7	Yes	Use Case 7: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version in a Clustered Environment on page 8-1
8	No	Use Case 8: Upgrading from ElectricFlow 5.x or 6.x to the Latest Version on page 9-1

Preparing for Your Upgrade

Review the following information before you upgrade the ElectricFlow software.

Upgrade Testing

In most implementations, ElectricFlow is being used in an environment that affects many users. We recommend that you test your upgrade on a separate test server to understand all aspects of the upgrade process. This minimizes the potential impacts to downstream users.

Backing Up Your Existing ElectricFlow Data

If you are upgrading an ElectricFlow server, it is *extremely* important that you back up your existing ElectricFlow data before upgrading. See [ElectricFlow Server Backups](#) on page 11-1 for more information about backups.

- Always back up the Plugins Directory. The default location is the `plugins` subdirectory within the data directory.
- Always back up the files that contain your configuration and custom settings. To ensure that all important settings are saved, back up the following subdirectories in `<data_dir>`:
 - The entire `conf` subdirectory (which contains the ElectricFlow server and agent configuration files)
 - Apache web server configuration files in the `apache/conf` subdirectory
 - The local MySQL database configuration file, `my.ini`, in the `mysql` subdirectory (if applicable)
- Always back up any other files where you have created custom configurations, specified other custom information, or created any type of modification.

Note: The number of ElectricFlow files you may have modified is too numerous to list, so you might want to back up the entire ElectricFlow data directory and other miscellaneous files that might have changed.

- If you use an artifact repository, back up your ElectricFlow repository configuration files in the `conf/repository` subdirectory.

- Determine if any changes were made to the custom editor or preflight driver script properties (installed by default). Back up those files if changes were made.

These properties are stored in the server-level property sheet, which can be viewed in the web UI by accessing the Administration tab/ Server subtab.

Custom editors are stored in the nested sheet named `ec_customEditors`. Preflight driver scripts are stored in the nested sheet named `ec_preflight`. The upgrade process overwrites default custom editor and preflight driver scripts with current versions. We recommend backing up any custom properties you created by renaming the property. For example, change `ec_preflight/clientDrivers/perforce` to `ec_preflight/clientDrivers/perforce_modified`.

Upgrade Installer Preservation

After you back up your ElectricFlow server, create a folder where you can download the `ElectricFlow-<version>` installation file.

IMPORTANT: This document refers to the installer using the new ElectricFlow product name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, be aware that we intend to stop providing the installer with "ElectricCommander" in the filename after this version.

MySQL Upgrades

ElectricFlow upgrades involving a MySQL database can take several hours to complete if you have a significant data set. *Do not interrupt the upgrade process.* You can corrupt your database if the upgrade process is interrupted. A restore from a previous database backup will be required.

- Use the `ectool` to view the upgrade progress. On a command line, enter

```
ectool getServerStatus
```

Note: An install/upgrade log file named `installer.log` is created in the `logs` subdirectory in the data directory.

Choosing the Correct Upgrade Method

This section describes the various upgrade methods and options for specific platform configurations. For information about supported server platforms and supported non-server platforms, see [Supported Server Platforms](#) on page 2-1 and [Supported Agent Platforms](#) on page 2-10.

User Interface Upgrade

This [method](#) provides a wizard for upgrading the ElectricFlow software on a supported server platform. This upgrade method is generally preferred by Windows users, but is supported on Linux platforms with the X Window System installed. See [User Interface Upgrade Method](#) on page 9-7 for more details.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Interactive Command-Line Upgrade

This [method](#) provides an interactive command-line for upgrading the ElectricFlow software on a supported server platform. This upgrade method is only available for Linux platforms. See [Interactive Command-Line Upgrade Method](#) on page 9-10 for more details.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

- **Clean Install**

This option allows you to specify a different installation directory for the new version. The files from your previous ElectricFlow version will not be removed or modified and will remain in their original directories.

IMPORTANT: On Linux, when ElectricFlow is already installed and you want to use the clean install upgrade method, you must do an advanced installation.

IMPORTANT: On Windows, a clean installation replaces the registry entries of the current installation. On Linux, a clean installation replaces the files in the `/etc/init.d` directory. The result is that only one instance of ElectricFlow (the new version) is running.

Silent Unattended Upgrade

This [method](#) provides a non-interactive command-line upgrade for supported server platforms. You may find this installation method preferable for upgrading multiple remote agents and servers. See [Silent \(Unattended\) Upgrade Method](#) on page 9-11 for more details.

Upgrade options:

- **Upgrade Existing Installation**

This option uninstalls the current release, installs the latest ElectricFlow release, collects the ElectricFlow service account credentials, configures the system with all property values, and restores custom files and data.

IMPORTANT: You cannot add a new repository server with this upgrade method.

Repository Server With an ElectricFlow Upgrade

After ElectricCommander 4.x, the only way to install a repository server on the same machine as other services is to uninstall and reinstall ElectricFlow. You can install the repository server on a different machine to avoid uninstalling and reinstalling ElectricFlow.

Non-Server Platform Agent Upgrade

You cannot directly upgrade a non-server platform agent (that is, an agent on a machine that is not a supported ElectricFlow server platform). You must uninstall and then reinstall these machines using the ElectricFlow installer. For more information, see [Uninstalling ElectricFlow](#) on page 10-1 and [Non-Server Platform Agent Installation Method](#) on page 13-24.

Stand-Alone Repository Server or Web Server Upgrade

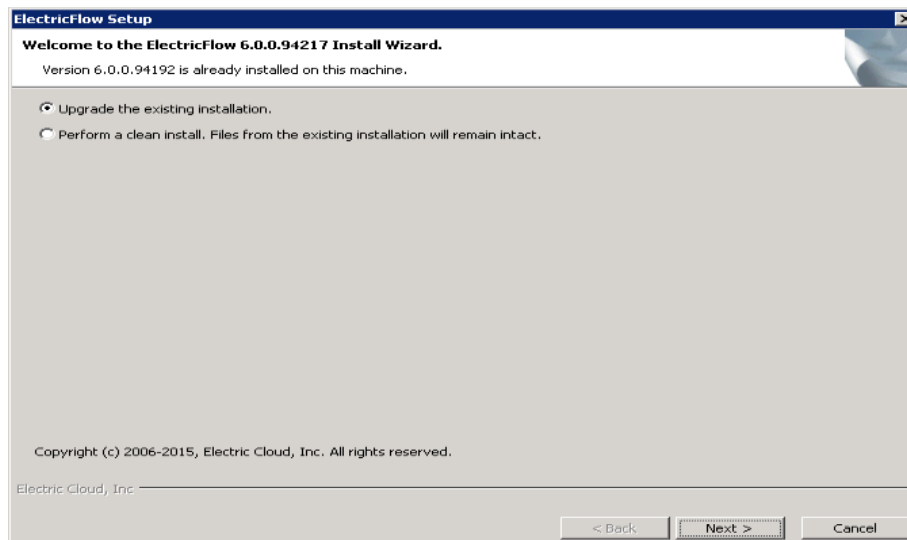
You cannot directly upgrade a stand-alone repository server or stand-alone web server. You must uninstall and then reinstall these servers using the ElectricFlow installer. The uninstall and reinstall process is required to install an agent on the server machine. Starting in ElectricCommander 4.2, an agent is required on the machine with the stand-alone repository server or web server. For more information, see [Uninstalling ElectricFlow](#) on page 10-1, [Installing ElectricFlow](#) on page 3-1, and [Copying Repository Contents](#) on page 9-11.

User Interface Upgrade Method

Use this procedure to upgrade the ElectricFlow software. Review [Preparing for Your Upgrade](#) on page 9-4 before performing this procedure.

IMPORTANT: This document refers to the installer using the new ElectricFlow product name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, be aware that we intend to stop providing the installer with "ElectricCommander" in the filename after this version.

1. Double-click the `ElectricFlow-<version>` file to begin installation. The Welcome to the ElectricFlow Installer screen appears.



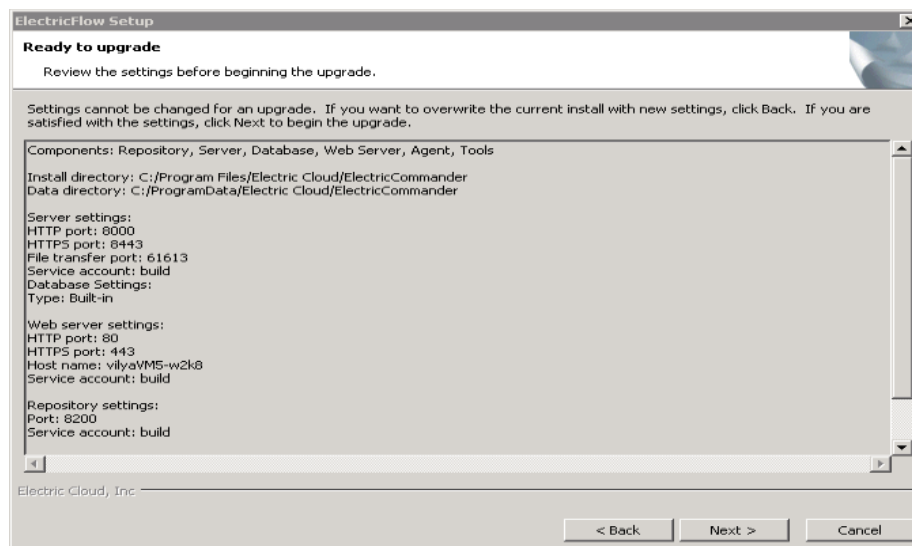
2. Choose one of the following options:

- Select **Upgrade the existing installation** if you want to upgrade your current ElectricFlow installation directory.
- Select **Perform a clean install** if you want to specify a different installation directory for the new version.

IMPORTANT: During a clean installation, current services remain running until you click **Next** on the Ready to Install screen. This means the new installation cannot use the same ports and directories as the existing installation. To use the same ports and directories, you must manually stop the existing services. This will free the existing ports and directories.

3. Click **Next** to upgrade the existing installation.

The Ready to Install screen appears.



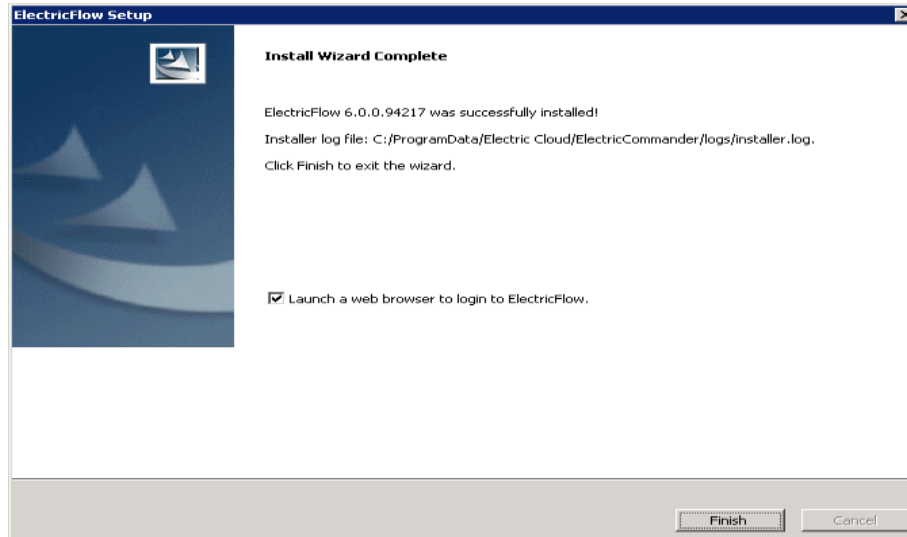
4. Review the upgrade settings.

Use the Back button to change your selections if necessary.

5. Click **Next** to continue.

The installer displays a status bar to show the progress of the upgrade process. You can also view the `installer.log` file to see progress. The time that it takes to complete this process depends on the size of the database. It may take fifteen minutes or longer to complete. Once this process is complete, the

new ElectricFlow version is installed.



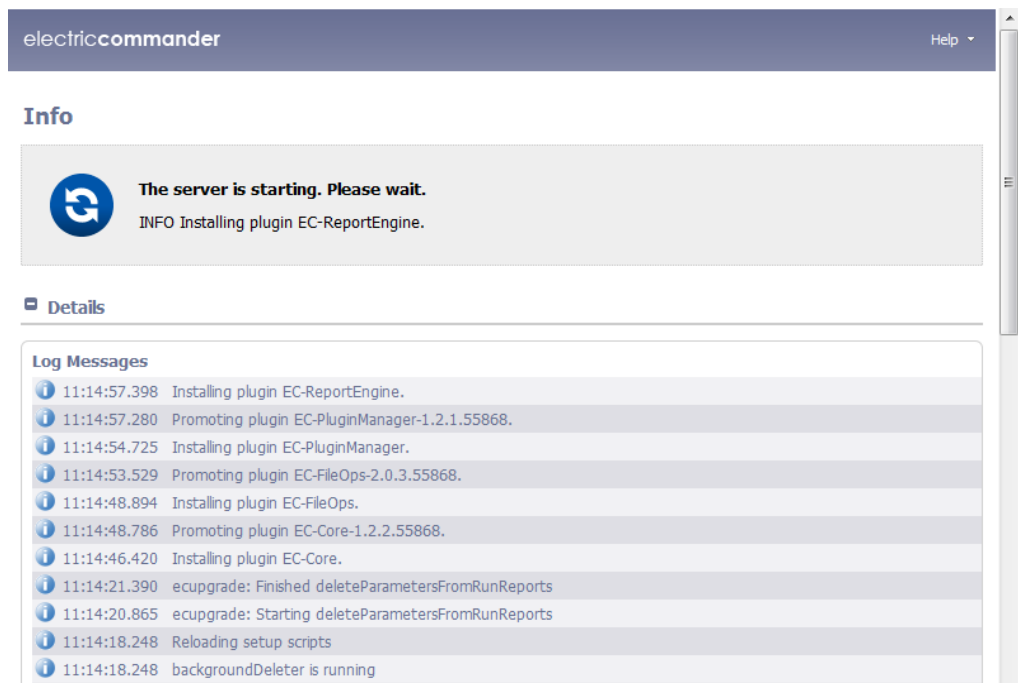
6. Select the **Launch a web browser to login to ElectricFlow** check box if you want ElectricFlow to open the login screen now.
7. Click **Finish** to complete the upgrade.

If an ElectricFlow server is being upgraded, when the installation is complete the server will continue to upgrade the database (if applicable). You will not be able to log in to the ElectricFlow server until the database upgrade is complete. You can view the upgrade status by using `ectool` from a command line:

```
ectool getServerStatus.
```

After clicking **Finish**, you may see a web page similar to the following screen if the upgrade is still in

progress.



Interactive Command-Line Upgrade Method

Use the following procedure to complete a command-line upgrade of a Linux platform. Review [Preparing for Your Upgrade](#) on page 9-4 before performing this procedure.

IMPORTANT: This document refers to the installer using the new ElectricFlow product name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, be aware that we intend to stop providing the installer with "ElectricCommander" in the filename after this version.

1. Choose one of the following commands to begin the upgrade:
 - If you have a Linux platform, enter `./ElectricFlow-<version> .`
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface.

To override this behavior, enter `./ElectricFlow-<version> --mode console.`

The system displays the following message:

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
This will install ElectricFlow on your computer. Continue? [n/Y]
```

2. Enter: `y`

The System displays the following message:

```
Upgrade the existing <version> installation to version <version>? [n/Y]
```

3. Choose one of the following options:

- If you want to upgrade your current ElectricFlow installation directory, enter `y`.
- Enter `n` to exit the installer.

The system displays the following message:

```
Installing agent...
Installing server...

Copied log file to "/opt/electriccloud/electriccommander/logs"

ElectricFlow <version> was successfully installed!

Installer log file: /opt/electriccloud/electriccommander/logs/installer.log.
```

Silent (Unattended) Upgrade Method

You can run the ElectricFlow upgrade in unattended (silent) mode with no user interface for either Windows or Linux.

IMPORTANT: This document refers to the installer using the new ElectricFlow product name. Anywhere the installer is referenced, you can also use the ElectricCommander named installer. However, be aware that we intend to stop providing the installer with "ElectricCommander" in the filename after this version.

Enter the following command from a command line to begin a silent upgrade:

```
./ElectricFlow-<version> --mode silent
```

Copying Repository Contents

Perform the following steps to copy the contents of an existing repository server into a newly installed repository server:

1. Install the new repository server software.
2. Stop the existing and new repository servers.
3. Copy the entire contents of the repository backingstore directory from the existing repository server to the corresponding location on the newly installed repository server.

The default location for the backingstore directory (`<datadir>/repository-data`) is:

- **UNIX** - `/opt/electriccloud/electriccommander/repository-data`
- **Windows** - `C:\ProgramData\Electric Cloud\ElectricCommander\repository-data`

Chapter 10: Uninstalling ElectricFlow

This section contains instructions for uninstalling ElectricFlow from various types of platforms.

Uninstalling ElectricFlow on Windows

Choose one of the following procedures to completely uninstall ElectricFlow from a Windows server, web server, agent, or developer machine.

Uninstalling on Windows 2008 or Windows 7

Use this procedure to completely uninstall ElectricFlow from a Windows 2008 or Windows 7 machine.

1. Go to **Control Panel > Uninstall a program**.
2. Select **ElectricCommander**.
3. Click **Uninstall**.

The system displays an "uninstall complete" message when the ElectricFlow software is removed.

4. Check the `<install_location>\electriccloud\electriccommander` directory and `C:\ProgramData\electriccloud\electriccommander` directory for any files that might remain. The uninstaller does not remove files that have been created or modified by ElectricFlow after the initial installation is complete.
5. Remove the files if you will reinstall the ElectricFlow software. You might want to move the files to a new location if you need to retain the files for troubleshooting.

Uninstalling ElectricFlow on UNIX

Use this procedure to completely uninstall ElectricFlow from a UNIX server, web server, agent, or developer machine.

1. Log in as **root**.
2. Choose one of the following commands to run:

- For Linux, enter:

```
/opt/electriccloud/electriccommander/uninstall
```

- For any other UNIX platform, enter:

```
/opt/electriccloud/electriccommander/uninstaller/uninstall
```

The system displays the following message.

```
This will completely remove ElectricFlow from your system. Are you sure you want to do this? [y/N]
```

3. Enter `y` to uninstall the software.

The system displays the following messages.

```
Uninstalling ElectricFlow...
```

```
Uninstall complete.
```

4. Check the `/opt/electriccloud/electriccommander` directory for any files that might remain. The uninstaller does not remove files that have been created or modified by ElectricFlow after the initial installation is complete.
5. Remove the files if you will reinstall the ElectricFlow software. You might want to move the files to a new location if you need to retain the files for troubleshooting.

Disaster Recovery

This topic has the steps to configure a Disaster Recovery (DR) setup. It explains what to expect after a successful failover.

Disaster Recovery Environment Setup

As part of a DR environment setup, you need to set up a secondary ElectricFlow site. This site must include a complete setup of all ElectricFlow components including:

- Database (Oracle, SQL Server, or MySQL)
- ElectricFlow server
- Web server
- Repository server
- Zookeeper
- Gateway agent
- Agent

ElectricFlow components that can be load balanced include the ElectricFlow server, web server, repository server, and gateway agent. Go to [Installing and Configuring a Load Balancer](#) on page 4-6 for details.

Along with replicating the component setup, data stores also need to be replicated and made available for ElectricFlow to operate the following:

- Database (using the vendor-recommended replication)
- Repository Server Data Store (replication of shared file locations where artifacts are stored)
- Plugins (typically copied to a shared file location)
- Certificates signed by an ElectricFlow CA (Certificate Authority)

Configurations and Settings

Follow these guidelines on the primary and secondary sites:

- Both the primary and secondary sites should be running the same version of ElectricFlow.
- Under normal operation, it should not be possible to do active transactions directly on the secondary (or replicated) database. That is, the secondary database should be in replication mode.
- Servers in the secondary site should be configured and should point to the secondary (or replicated) database. Therefore, all ElectricFlow services in the secondary site (the ElectricFlow server, web server, and repository server) should be set to start in **Manual** mode. These services should not automatically restart after a reboot. This prevents inadvertent write operations into the replicated database. Details on recommended steps for setting up a secondary site appear below.
- We recommend using DNS Failover to minimize the downtime when moving from the primary site to the secondary site. This allows end users accessing the web servers to continue using the same URL. Agents that are running jobs can send their finish job notification to the secondary server, allowing the jobs to succeed.

To configure Disaster Recovery, follow these steps:

1. Add following lines to the `wrapper.conf` file for the server nodes:

```
wrapper.java.additional.1600=DCOMMANDER_IGNORE_SERVER_MISMATCH=1
wrapper.java.additional.1601=DCOMMANDER_PRESERVE_SESSIONS=1
```

This avoids the following errors during the failover to the secondary site:

```
20160815T22:10:52.406 | 10.0.2.206 | DEBUG | bootstrap
|                                     | schemaMaintenance
| OperationInvoker | Exception: InvalidServer: The ZooKeeper/Exhibitor
setting of the last cluster
('ip100179155:8080,ip100145251:8080,ip100133239:8080') to connect to the
database is different from the ZooKeeper/Exhibitor setting of the current cluste
r
('ip1008088:8080,ip100196103:8080,ip100515:8080'). Check that the cluster
is configured for the correct database. To allow this server cluster to become t
he new
owner for this data, update the database configuration with the ignoreServerMism
atch
flag set.
```

```
20160815T22:10:52.406 | 10.0.2.206 | WARN | bootstrap |
|                                     |
| ServerStatus | InvalidServer: The ZooKeeper/Exhibitor setting
of
the last cluster ('ip100179155:8080,ip100145251:8080,ip100133239:8080') to
connect to the database is different from the ZooKeeper/Exhibitor setting of the
current cluster ('ip1008088:8080,ip100196103:8080,ip100515:8080'). Check
that the cluster is configured for the correct database. To allow this server cl
uster
to become the new owner for this data, update the database configuration with th
e
ignoreServerMismatch flag set.
```

2. Make sure that the following recommended standard setup steps are performed:

- a. `commander.properties` in ZooKeeper should have the `COMMANDER_SERVER_NAME` set to the FQDN (Fully Qualified Domain Name) of the load balancer of the ElectricFlow server cluster. This can be set using the following command:

```
ecconfigure --serverName <FLOW_SERVER_LOAD_BALANCER_FQDN>
```

- b. Both the primary and secondary sites should have the same files (including content):

- `keystoreFile`
- `passkeyFile`
- `commander.properties`

In the cluster setup, these files are stored in their respective ZooKeeper instances (the primary and secondary instances).

Note: The ZooKeeper connection is configured using:

```
ecconfigure --serverZooKeeperConnection  
<ZooKeeper_servers_comma_seperated_list>
```

For example:

```
ecconfigure --serverZooKeeperConnection  
ip1008088:2181,ip100196103:2181,ip100515:2181
```

- c. Run the following command on the web servers to ensure that the `COMMANDER_SERVER` property in the `httpd.conf` file is set to ElectricFlow server's load balancer FQDN:

```
ecconfigure --webTargetHostName <FLOW_SERVER_LOAD_BALANCER_FQDN>
```

Note: The `httpd.conf` file is usually in `apache/conf` on a Linux machine and `ProgramData\Electric Cloud\ElectricCommander\apache\conf` on a Windows machine.

- d. Similarly, on each repository server, run the following command to set `COMMANDER_HOST` in the `server.properties` file:

```
ecconfigure --repositoryTargetHostName <FLOW_SERVER_LOAD_BALANCER_FQDN>
```

Note: These are the default locations for the `server.properties` file:

- In Linux,

```
/opt/electriccloud/electriccommander/conf/repository/server.properties
```

- In Windows, `C:\ProgramData\Electric`

```
Cloud\ElectricCommander\conf\repository\server.properties
```

- e. Set the "Server IP address" in the ElectricFlow server property to the Flow Server Load Balancer FQDN (the same value as `FLOW_SERVER_LOAD_BALANCER_FQDN`).

- f. Set the “Stomp Client URI” server property to `stomp+ssl://<FLOW_SERVER_LOAD_BALANCER_FQDN>:61613`.

Note: If the load balancer does SSL termination, uncheck the **Use SSL for Stomp** option. All the ElectricFlow server nodes must be restarted to affect this change, because by default, the check box is selected.

- g. In the **Cloud > Resource** page, register all the ElectricFlow local agents (the ones that run on same machine as ElectricFlow server) from both primary and secondary sites. Make sure that for the cluster setup, you do not have *local* resources.
- h. Create resource pools named “*local*” and “*default*”. For both, add ElectricFlow local agents.
- Both the local and default pools are used by the ElectricFlow standard job processing. For example, sentry jobs run on local pool resources.
- i. (Optional) If trusted agents are used, in addition to copying the keystore file, along with keystore file, the `conf/security` folder should be copied to the secondary site’s ZooKeeper. This folder stores the ElectricFlow Certificate Authority information along with the certificates that are signed by ElectricFlow.

Perform the following steps to copy this folder from the primary to secondary site:

- i. Log into the primary site’s ElectricFlow server, and run the following command to get the `conf/security` folder from the primary ZooKeeper into the local `/tmp/electriccommander/conf/security` folder:

```
COMMANDER_ZK_CONNECTION=<ZooKeeper_Primary_Server_IP>:2181
<INSTALL_DIRECTORY>/jre/bin/java -cp
<INSTALL_DIRECTORY>/server/bin/zk-config-tool-jar-with-dependencies.jar
com.electriccloud.commander.zkconfig.ZKConfigTool -readFolder
/commander/conf/security /tmp/electriccommander/conf/security
```

- ii. Log into the secondary site’s ElectricFlow server, and run the following command to upload the `conf/security` folder from the local folder to ZooKeeper:

```
COMMANDER_ZK_CONNECTION=<ZooKeeper_Secondary_Server_IP>:2181
<INSTALL_DIRECTORY>/jre/bin/java -cp
<INSTALL_DIRECTORY>/server/bin/zk-config-tool-jar-with-dependencies.jar
com.electriccloud.commander.zkconfig.ZKConfigTool -writeFolder
/commander/conf/security /tmp/electriccommander/conf/security
```

3. Set the repository data store on the primary and secondary sites.

Repository servers set up on primary and secondary sites can share the same repository data store.

The repository data can be replicated, and the repository servers can point to the respective data store locations. For each repository server, set the `REPOSITORY_BACKING_STORE` in the `server.properties` file to a UNC path on a network share on the file server.

For example:

```
REPOSITORY_BACKING_STORE=//10.0.109.72/repo_data/repositorydata
```

4. Register the repository server in the ElectricFlow UI. If the repository server cluster is set up, use the load balancer URL (for example, `https://<REPOSITORY_SERVER_LOAD_BALANCER_FQDN>:8200`).

During the failover to the secondary site, FQDN should point to the repository servers in the secondary site.

5. It is recommended that the plugins folder in the network share must be accessible from the remote web servers as mentioned in [Universal Access to the Plugins Directory](#) on page 5-10.
6. For the initial installation and setup of the secondary site, perform the following recommended steps:
 - a. Set up the secondary database in normal or nonreplicated mode.
 - b. Follow instructions as described in this installation guide to set up all the servers, including the ElectricFlow, web, repository, and Zookeeper servers.
 - c. Make sure that the `database.properties` file is set up to point to the correct secondary site's database server. This file will be stored in ZooKeeper when the cluster setup is used. For the primary site, it should point to the primary database. For the secondary site, it should point to the secondary database.
 - d. Ensure that all the servers are running properly, including the connection to the database. At this time, the secondary site is not set up for replication, and operates as a separate installation of ElectricFlow.
 - e. Before setting up the secondary site's database in replication mode, shut down all secondary ElectricFlow servers, web servers, repository servers, and others. After these steps are performed, there should not be any write transactions to the secondary database.
 - f. The first time that the secondary database is set up, a schema for database tables is created. Before proceeding to set this database with replication, this schema may have to be deleted. This avoids a schema name conflict when replication is enabled.
 - g. Based on the disaster recovery option chosen for the database, set up the secondary database in replication mode.

Steps to Perform During a Disaster Recovery Failover

When a disaster event happens that interrupts the operations on the primary site, follow these steps to move the operations to the secondary site.

1. Shut down any services that might still be running on the primary site. All components including the database, ElectricFlow server, and repository server can be shut down. Doing this ensures that no more transactions happen on the primary site.
2. Begin switching operations to the secondary site by restoring and updating the secondary site's database with the latest data. The steps to do this may vary based on the disaster recovery method used for database.
3. Delete the `brokerdata` folder on the ElectricFlow server nodes. For example, in Windows, delete the `C:\ProgramData\Electric Cloud\ElectricCommander\brokerdata` folder.
4. Follow the DNS failover procedure, and update the DNS entries to point to servers in the secondary site. This includes updating the entries for servers including web server, ElectricFlow server, and repository server. There may also be other servers based on configuration.
5. Bring up all the servers and infrastructure in the secondary site. Start the ElectricFlow services on different machines, including the ElectricFlow server, web server, repository server, and gateway agents.

6. Based on the nature of the disaster event, certain active operations running on the primary site may be interrupted, and need to be restarted. For example, a build or deploy application process may fail and error out. Use the ElectricFlow UI to review such failures, and take the appropriate corrective actions, usually by executing those failed processes again.
7. The secondary's site database now acts as the master. We recommend that you set up a database that will act as a slave.

Server Maintenance

See [Maintenance](#) on page 11-1 for details on ElectricFlow server maintenance.

Chapter 11: Maintenance

This section contains common maintenance procedures.

ElectricFlow Server Backups

You should back up your existing ElectricFlow data frequently. We recommend full regular (nightly) database backups and database backups before an upgrade.

Data Backup Methods

There are two ways to back up your data. You can use a database-specific backup tool to create a database dump, or you can use the ElectricFlow (`ectool export`) tool to create a complete XML database backup. This section describes the differences between the two types of backups.

IMPORTANT: In ElectricCommander 4.0 and later, we recommend that you do not use the `ectool export` tool with jobs on an active system to create a trusted database backup.

Database Dumps

You must use a database-specific backup tool to create a database dump. Database dumps have the following characteristics:

- The backup process takes much less time to complete than full XML exports
- Database dumps (for example MySQL) must be performed while the database is live, up, and running.
- You can quickly restore a database from a database dump.

IMPORTANT: A database dump can only be restored to the same type of database. If you are planning to switch your database type when you restore from the backup, you must create an XML backup.

Complete XML Database Backup

You must use the `ectool export` tool to create a complete XML database backup. Complete XML database backups have the following characteristics:

- The tool must be used while the ElectricFlow server is running.
- The database backup process can take considerably longer than simply creating a database dump, but this method is necessary in the following situations:

- Backing up the database is not an available option.
- You need to migrate from one type of database to another. For example, MySQL to Oracle.
- You want a full export in a text form you can search with an editor.

Note: While it might not be feasible to run a full XML export regularly (such as nightly), it is much faster to export all data except jobs by using the `--excludeJobs` option. We recommend doing jobless exports if the system is active (running jobs). For more information about the `ectool export` command, see the “Using ectool and the ElectricFlow API” help topic in the ElectricFlow online help system.

Preparing for a Backup

Before you backup your ElectricFlow server:

- Make sure you have plenty of free space available because full database dumps and XML export files can be extremely large.
 - Compress database dumps if they are not compressed by default.
- Regularly perform maintenance such as, but not limited to:
 - Pruning job workspaces
 - Deleting or compressing ElectricFlow log files
 - Deleting unused projects and/or procedures

Note: You can use ElectricFlow to perform backups by creating a procedure that runs the database dump or export command.

Backing Up an ElectricFlow Server

Use the following procedure to back up your ElectricFlow server data. Review [Preparing for a Backup](#) on page 11-2 before performing this procedure.

1. Choose one of the following methods to back up your data:
 - Use a database-specific backup tool to create a database dump.
 - Use `ectool export` to create a complete XML database backup.

Note: For more information about database backup methods, see [Data Backup Methods](#) on page 11-1.

2. Save the `passkey` file located in one of these directories:
`/opt/electriccloud/electriccommander/conf/passkey` in Linux or
`C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey` in Windows . When you restore your server, this passkey must be in place so ElectricFlow can decrypt passwords for user impersonation, LDAP, and the database connection.
3. Back up the plugins directory.
 - The plugins directory is stored in a server setting property (`/server/settings/pluginsDirectory`).
 - If the property does not exist, the server uses the default location, which is the `plugins` subdirectory in the data directory.
4. Back up the files containing your custom configurations and settings to ensure all important settings are saved.

- The default location for ElectricFlow server and agent configuration files is the `conf` subdirectory in the data directory.
 - The default location for the Apache web server configuration files is the `apache/conf` subdirectory in the data directory.
 - The default location for the local MySQL database configuration file, `my.ini`, is the `mysql` subdirectory in the data directory.
5. Verify that your backup contains the following items:
- Database dump and/or XML export
 - The `passkey` file
 - The contents of the plugins directory
 - Configuration files
 - Keystores

ElectricFlow Server Restores

This section describes common restore related procedures for recovering ElectricFlow data.

Preparing for a Restore

Before you attempt to restore an ElectricFlow server:

- You *must* have a backup of your source ElectricFlow server.
 - If you are restoring your data to the exact same database or the same database type (for example, from one MySQL database to another MySQL database on a different system), a database backup is sufficient.
 - If you are switching to a different database type, you will need an XML export.

Note: Any activity on the source server *after the backup was created* will not exist on the destination server.

- The destination system must have an ElectricFlow server already installed and running, and this server must be running the same version or newer version than the source server.

Restoring Your ElectricFlow Server

The following section contains various procedures for restoring ElectricFlow data. Review [Preparing for a Restore](#) on page 11-3 section before performing any of the following procedures.

Note: All ectool commands used in the following scenarios are fully documented in the ElectricFlow online help system. See the “Using ectool and the ElectricFlow API” help topic.

Restore the Same ElectricFlow Server and Database

Use the following procedure to restore your ElectricFlow server due to a catastrophic failure or unsuccessful upgrade.

1. Obtain a backup of the source system.
2. Stop the destination ElectricFlow server. For more information, see [Starting and Stopping Servers and Agents Manually](#) on page 11-12 for platform-specific commands.

3. If you are using a database dump (where the source and destination systems must both be using the same type of database), load the backup into the destination database.

This will be done with a command specific to the database you are using.

4. Start the destination ElectricFlow server.
5. If you are using an XML export file, use the `ectool import` command to import the data into the destination ElectricFlow server.
6. Use the `ectool shutdownServer --restart 1` command to restart the destination server.

Keep the Same ElectricFlow Server but Switch the Database

Use the following procedure to restore your ElectricFlow server if you are doing one of these tasks:

- Switching from the built-in database installation to an external database.
 - Upgrading to a higher performance system for the database.
1. Obtain a backup of the source system.
 2. Stop the destination ElectricFlow server. For more information, see [Starting and Stopping Servers and Agents Manually](#) on page 11-12 for platform-specific commands.
 3. Stop and disable the original database.
 4. If you are using a database backup (where the source and destination systems must both be using the same type of database), load the database dump into the destination database.

This will be done with a command specific to the database you are using.

5. Start the destination ElectricFlow server.
6. Set the server database configuration to point to the new database. Point to the new database one of these ways:

See the “Database Configuration” help topic in the ElectricFlow web interface.

Use the `ectool setDatabaseConfiguration` command.

7. If you are using an XML export file, use the `ectool import` command to import the data into the destination ElectricFlow server.
8. Use the `ectool shutdownServer --restart 1` command to restart the destination server.

Switch the ElectricFlow Server but Keep the Same Database

Before switching the server, be aware of the following:

- All files and directories copied to the Destination ElectricFlow Server should be owned by the user configured to run the ElectricFlow server daemon.
- Make sure that the host name of local agent is set to `127.0.0.1` using **Cloud > Resources > Local > Resource Details**.
- When you install ElectricFlow without a built-in database, you can configure the database only by using `ectool`.

Use the following procedure to restore ElectricFlow if you are upgrading to a higher performance ElectricFlow server system.

1. Make sure you have a backup of the source system.
2. Check the IP Address System property by selecting **Administration > Server > Settings** on the old (source) ElectricFlow system.

This field is empty by default to enable dynamic connections between the ElectricFlow server and agents.

If the field is not empty, you must enter the IP address for the Destination ElectricFlow Server.

3. Stop the destination ElectricFlow server.

For more information, see [Starting and Stopping Servers and Agents Manually](#) on page 11-12 for platform-specific commands.

4. Stop and disable the source ElectricFlow server.
5. Copy the `passkey` and `keystore` files from the source ElectricFlow backup to the destination system in one of these directories: `/opt/electriccloud/electriccommander/conf/passkey` in Linux or `C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey` in Windows.
6. Copy the backed-up plugins to the destination system.

You may encounter one of these scenarios:

- If the `/server/settings/pluginsDirectory` property does not exist, the server uses the default location (the `plugins` subdirectory in the data directory).

Copy the backed-up plugins to that directory on the destination system.

- The plugins are stored in a local directory valid on both systems.

Copy the backed-up plugins to the same directory on the destination system.

- The plugins are stored in a shared directory valid on both systems.

You do not need to do anything.

- The plugins are stored in a directory not accessible on the destination system.

This can happen

- If the source and destination systems have different operating systems (such as Windows to Linux).
- If the plugins directory on the source system is on a drive that does not exist on the destination system.

Copy the backed-up plugins to a new directory accessible to the destination system. When the server starts, set the `/server/settings/pluginsDirectory` property to the new directory and restart the ElectricFlow server.

7. If you use MySQL database, do these steps on destination system:
 1. Install MySql JDBC driver: http://docs.electric-cloud.com/commander_doc/4_2_6/HTML5/Install/Content/ec_install_guide/installing_ec/10MySQL%20Driver.htm.
 2. Configure access to the ElectricFlow database user from the IP address or FQDN on the destination system.
8. Start the destination ElectricFlow server.

9. Because the ElectricFlow host changed, connect the ElectricFlow database to the new host:

On the command-line:

1. Use `ectool setDatabaseConfiguration` to specify a database configuration and set the `--ignoreServerMismatch` option.
2. Use the following command to restart the destination server: `ectool shutdownServer --restart 1`.

In the web interface, you should automatically be redirected to the **Database Configuration** page.

1. Enter the appropriate database configuration.
 2. Select the **Ignore server hostname mismatch** check box.
 3. Select **Same instance on a new host**.
 4. Click **Save and Restart**.
10. If you copied the plugins directory to a directory that does not match the plugins directory on the source system:
 1. Set the `/server/settings/pluginsDirectory` property to this new directory.

You can use the `ectool setProperty` command to set this value.
 2. Restart the ElectricFlow server.

Switch Both the ElectricFlow Server and Database

Use the following procedure to restore ElectricFlow if you are upgrading to higher performance systems for both the ElectricFlow server and the database.

1. Make sure you have a backup of the source system.
2. Check the IP Address System property by selecting **Administration > Server > Settings** on the old ElectricFlow system.

This field is empty by default to enable dynamic connections between the ElectricFlow server and agents.

If the field is not empty, you must enter the IP address for the new ElectricFlow server.

3. Stop the destination ElectricFlow server.

For more information, see [Starting and Stopping Servers and Agents Manually](#) on page 11-12 for platform-specific commands.

4. Stop and disable the source ElectricFlow server.
5. Stop and disable the original database.
6. Copy the `passkey` file from the backup to the destination system (in one of these directories: `/opt/electriccloud/electriccommander/conf/passkey` in Linux or `C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey` in Windows).

7. Copy the backed-up plugins to the destination system.

You may encounter one of these scenarios:

- If the `/server/settings/pluginsDirectory` property does not exist, the server uses the default location (the `plugins` subdirectory in the data directory).

Copy the backed-up plugins to that directory on the destination system.

- The plugins are stored in a local directory valid on both systems.
Copy the backed-up plugins to the same directory on the destination system.

- The plugins are stored in a shared directory valid on both systems.

You do not need to do anything.

- The plugins are stored in a directory not accessible on the destination system.

This can happen:

- If the source and destination systems have different operating systems (such as Windows to Linux).
- If the plugins directory on the source system is on a drive that does not exist on the destination system.

Copy the backed-up plugins to a new directory accessible to the destination system. When the server starts, set the `/server/settings/pluginsDirectory` property to the new directory and restart the ElectricFlow server.

8. If you are using a database backup (where the source and destination systems must both be using the same type of database), load the database dump into the destination database.

This operation is completed with a command specific to the database you are using.

9. Start the destination ElectricFlow server.

10. Because we have replaced the `passkey`, the database password is no longer valid. You need to reset the database password (default: `commander`) and ignore the `passkey` mismatch either from the command-line or the web interface.

- On the command-line, use `ectool setDatabaseConfiguration` to specify the password and set the `--ignoreServerMismatch` and `--ignorePasskeyMismatch` options.
- In the web interface, you should automatically be redirected to the **Database Configuration** page. Enter the database password and select the **ignore invalid passkey** check box.

11. If you are using an XML export file, use the `ectool import` command to import the data into the destination ElectricFlow server.

12. Use the `ectool shutdownServer --restart 1` to restart the destination server.

13. If you copied the plugins directory to a directory that does not match the plugins directory on the source system, set the `/server/settings/pluginsDirectory` property to this new directory and restart the ElectricFlow server.

You can use the `ectool setProperty` command to set this value.

Create a Clone of the ElectricFlow Server and the Database

Use the following procedure to restore your ElectricFlow server if you are setting up a production-like environment for testing.

1. Make sure you have a backup of the source system.
2. Check the IP Address System property by selecting **Administration > Server > Settings** on the old ElectricFlow system.

This field is empty by default to enable dynamic connections between the ElectricFlow server and agents.

If the field is not empty, you must enter the IP address for the new ElectricFlow server.

3. Stop the destination ElectricFlow server.

For more information, see [Starting and Stopping Servers and Agents Manually](#) on page 11-12 for platform-specific commands.

4. Copy the `passkey` file from the backup to the destination system)(in one of these directories: `/opt/electriccloud/electriccommander/conf/passkey` in Linux or `C:\ProgramData\Electric Cloud\ElectricCommander\conf\passkey` in Windows).
5. Copy the backed-up plugins to the destination system.

You may encounter one of these scenarios:

- If the `/server/settings/pluginsDirectory` property does not exist, the server uses the default location (the `plugins` subdirectory in the data directory).

Copy the backed-up plugins to that directory on the destination system.

- The plugins are stored in a local directory valid on both systems.
Copy the backed-up plugins to the same directory on the destination system.
- The plugins are stored in a shared directory valid on both systems.

You do not need to do anything.

- The plugins are stored in a directory not accessible on the destination system.

This can happen:

- If the source and destination systems have different operating systems (such as Windows to Linux).
- If the plugins directory on the source system is on a drive that does not exist on the destination system.

Copy the backed-up plugins to a new directory accessible to the destination system. When the server starts, set the `/server/settings/pluginsDirectory` property to the new directory and restart the ElectricFlow server.

6. If you are using a database backup (the source and destination systems must both be using the same type of database), load the database dump into the destination database.

This operation is completed with a command specific to the database you are using.

7. If you are using a database backup, disable schedules, resources, or both on both servers.
 - Two servers should never *talk* to the same agent. The two servers share the same identity because they share exact copies of the database.
 - Disabling schedules prevents jobs from launching unexpectedly.
 - Disabling resources prevents scheduled or manually launched jobs from running on production agents. This operation is completed with a command specific to the database you are using.
8. Start the destination ElectricFlow server.
9. Because we have replaced the `passkey`, the database password is no longer valid. You need to reset the database password (default: `commander`) and ignore the `passkey` mismatch either from the command-line or the web interface.
 - On the command-line, use `ectool setDatabaseConfiguration` to specify the password and set the `--ignoreServerMismatch` and `--ignorePasskeyMismatch` options.
 - In the web interface, you should automatically be redirected to the **Database Configuration** page. Enter the database password and select the **ignore invalid passkey** check box.
11. If you are using an XML export file, disable schedules, resources, or both on both servers.
 - Two servers should never “talk” to the same agent. The two servers share the same identity because they share exact copies of the database.
 - Disabling schedules prevents jobs from being launched unexpectedly.
 - Disabling resources prevents scheduled or manually launched from running on production agents.

Disable the schedules and resources one of these ways:

- Modify the import file by replacing `<resourceDisabled>0</resourceDisabled>` with `<resourceDisabled>1</resourceDisabled>`.
 - Use the `ectool import` command with the `--disableSchedules` flag turned on to disable schedules.
12. Use the `ectool shutdownServer --restart 1` command to restart the destination server.
 13. If you copied the `plugins` directory to a directory that does not match the `plugins` directory from the source system, set the `/server/settings/pluginsDirectory` property to the new directory and restart the ElectricFlow server.

You can use the `ectool setProperty` command to set this value.

Switching to an Alternate Database

If you did not deselect the “database” check box during installation, you can switch to another database at any time. You can use this procedure to switch from the built-in database or to switch from the current alternate database to a different alternate database.

IMPORTANT: If you are using two different ElectricFlow servers, they cannot point to the same database.

The export operation is run by the server process, not through ectool. The command is not run by the agent, but by the server itself. Therefore, it has some impact if the server agent service user and the server service user are different. For example, the following commands in the same step are executed by two different users:

```
mkdir ("/path/foo");
$ec->export ("/path/foo/project.xml",
{path=>"/projects/MYPROJ"}
);
```

The /path/foo directory creation is executed by the agent service, which means that the agent user needs permission to create the directory. The export is executed by the ElectricFlow service user.

Use these procedures to configure a new database and migrate the existing data.

Preventing Database Changes During the Export

Before you perform an export, ensure that the ElectricFlow server is inactive (meaning that it cannot accept jobs) by completing the following steps on the server:

1. Disable ElectricSentries.
2. Disable project schedules.
3. Check that no jobs are running on any resources.
4. Disable all resources so that no new job steps can run.

This ensures a complete XML file by preventing changes to the ElectricFlow database during the export.

Exporting and Importing Your Data

1. Export your data by entering the following command:

```
ectool export <filename> --compress 1
```

2. Set the database configuration using the web interface or ectool. For more information, see [Configuring ElectricFlow to Use an Alternate Database](#) on page 5-3.
3. Restart the ElectricFlow server by entering the following command:

```
ectool shutdownServer --restart 1
```

4. Import your data by entering the following command:

```
ectool import <filename> --force 1
```

Apache Web Server or Agent Certificates

By default, ElectricFlow generates a temporary self-signed certificate during web server installation. This certificate is used whenever a browser makes an HTTPS connection to the Apache server. Because the certificate is self-signed, browsers will generate untrusted certificate messages. To prevent these types of warnings, you must generate a new Apache web server or agent certificate signed by a recognized certificate authority (CA).

IMPORTANT: Before performing any of the following procedures, back up the `$DATA_DIRECTORY/conf` and `$DATA_DIRECTORY/apache/conf` directories.

Generating a CA Request

Use the following procedure to generate a CA request.

1. Locate the `DATA_DIRECTORY` directory for your platform. The default directory locations are:
 - Linux – `/opt/electriccloud/electriccommander`
 - Windows 2008 or Windows 7 – `C:\ProgramData\Electric Cloud\ElectricCommander`
2. Locate the appropriate certificate signing request file generated during installation:
 - Agent – `$DATA_DIRECTORY/conf/agent.csr`
 - Web Server – `$DATA_DIRECTORY/apache/conf/server.csr`
3. (Optional) Update `server.csr` with custom SSL configuration data.
 - Edit the file `$DATA_DIRECTORY/apache/conf/serverssl.cnf` to add your custom configuration data.
 - From the `$DATA_DIRECTORY/apache/conf`, run this command:


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

Sending the CA Request

Send the `server.csr` (or `agent.csr`) file to a certificate authority to sign the certificate. The CA verifies the information inside and sends you a signed certificate in response. The signed certificate includes the original certificate and the CA signature.

Installing the Signed Certificate

Use the following procedure to install a signed certificate:

1. Replace the existing certificate in the `DATA_DIRECTORY` directory with the new signed certificate you received from the CA. The signed certificate file should be placed in one of the following locations:
 - Agent – `$DATA_DIRECTORY/conf/agent.crt`
 - Web Server – `$DATA_DIRECTORY/apache/conf/server.crt`
2. Restart the agent and/or Apache services.

Using chkconfig

`chkconfig` is a simple command-line tool for maintaining the `/etc/rc[0-6].d` directory hierarchy. This tool relieves system administrators from the task of directly manipulating numerous symbolic links in those directories. The Linux `chkconfig` command can be used to manipulate ElectricFlow services running on UNIX platforms.

`chkconfig` - updates and queries runlevel information for system services

```
chkconfig --list [name]
chkconfig --add name
chkconfig --del name
chkconfig [--level levels] name <on|off|reset>
chkconfig [--level levels] name
```

Examples

```
(list current settings for the local ElectricFlow repository service)
/sbin/chkconfig commanderRepository --list
commanderRepository 0:off 1:off 2:off 3:on 4:off 5:on 6:off
```

```
(disable autostart on reboot)
/sbin/chkconfig commanderRepository off
/sbin/chkconfig commanderRepository --list
commanderRepository 0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

Note: For every service, each runlevel has either a “start” script or a “stop” script. When switching runlevels, `init` will not restart an already-started service, and will not re-stop a service that is not running.

Starting and Stopping Servers and Agents Manually

ElectricFlow servers and agents need to be manually stopped and started during administrative maintenance, upgrades, third party software installs, or system maintenance.

Stop All ElectricFlow Agent Services

To stop all ElectricFlow agent services, choose one of the following options:

- For Windows:
 - a. Go to **Control Panel > Administrative Tools > Services**.
 - b. Stop “ElectricFlow Agent”.
- From a command line, enter `sc stop CommanderAgent`
- For UNIX / Mac: using a shell, logged in as `root`, enter one of the following commands:
 - Linux - `/etc/init.d/commanderAgent stop`
 - Solaris - `/etc/init.d/ecmdrAgent stop`
 - HP-UX - `/sbin/init.d/ecmdrAgent stop`
 - Mac - `launchctl unload /Library/LaunchDaemons/ecmdrAgent.plist`

Stop All ElectricFlow Server Services

To stop all ElectricFlow server services, choose one of the following options:

- For Windows:
 - a. Go to **Control Panel > Administrative Tools > Services**.
 - b. Stop “ElectricFlow Server”.
 - c. Stop “ElectricFlow Web Server”.
 - d. Stop “ElectricFlow Database” (if it exists).
 - e. Stop “ElectricFlow Repository Server”.
- From a command line, enter:
 - a. `sc stop CommanderServer`
 - b. `sc stop CommanderApache`

- c. `sc stop CommanderMySQL (if it exists)`
- d. `sc stop CommanderRepository`
- For Linux: using a shell, logged in as `root`, enter:
 - a. `/etc/init.d/commanderServer stop`
 - b. `/etc/init.d/commanderApache stop`
 - c. `/etc/init.d/commanderMySQL stop (if it exists)`
 - d. `/etc/init.d/commanderRepository stop`

Start All ElectricFlow Agent Services

To start all ElectricFlow agent services, choose one of the following options:

- For Windows:
 - a. Go to **Control Panel > Administrative Tools > Services**.
 - b. Start "ElectricFlow Agent".
- From a command line, enter `sc start CommanderAgent`
- For UNIX / Mac: using a shell, logged in as `root`, enter one of the following commands:
 - Linux - `/etc/init.d/commanderAgent start`
 - Solaris - `/etc/init.d/ecmdrAgent start`
 - HP-UX - `/sbin/init.d/ecmdrAgent start"`
 - Mac - `launchctl load /Library/LaunchDaemons/ecmdrAgent.plist`

Start All ElectricFlow Server Services

To start all ElectricFlow server services, choose one of the following options:

- For Windows:
 - a. Go to **Control Panel > Administrative Tools > Services**.
 - b. Start "ElectricFlow Database" (if it exists).
 - c. Start "ElectricFlow Server".
 - d. Start "ElectricFlow Web Server".
 - e. Start "ElectricFlow Repository Server".
- From a command line, enter:
 - a. `sc start CommanderMySQL (if it exists)`
 - b. `sc start CommanderServer`
 - c. `sc start CommanderApache`
 - d. `sc start CommanderRepository`
- For Linux: using a shell, logged on as `root`, enter:
 - a. `/etc/init.d/commanderMySQL start (if it exists)`
 - b. `/etc/init.d/commanderServer start`

- c. `/etc/init.d/commanderApache start`
- d. `/etc/init.d/commanderRepository start`

Log File Locations

The following information is for default “run time” log locations.

Agent Logs

Platform	Default Path
Windows 2008	C:\ProgramData\Electric Cloud\ElectricCommander\logs\agent
Windows 7	
UNIX	/opt/electriccloud/electriccommander/logs

Note: Agent logs “roll over” periodically so individual logs do not grow too large and older logs are deleted. Roll-over parameters are configurable in `conf/logback.xml` and `conf/agent.conf`.

Server Logs

Platform	Default Path
Windows 2008	C:\ProgramData\Electric Cloud\ElectricCommander\logs
Windows 7	
UNIX	/opt/electriccloud/electriccommander/logs

Note: Server logs “roll over” periodically so individual logs do not grow too large and older logs are deleted. Roll-over parameters are configurable in `conf/logback.xml` and `conf/agent.conf`.

Web Server Logs

Platform	Default Path
Windows 2008	C:\ProgramData\Electric Cloud\ElectricCommander\apache\logs
Windows 7	
UNIX	/opt/electriccloud/electriccommander/apache/logs

Repository Server Logs

Platform	Default Path
Windows 2008	C:\ProgramData\Electric Cloud\ElectricCommander\logs\repository
Windows 7	
UNIX	/opt/electriccloud/electriccommander/logs/repository

Installer Logs

Platform	Default Path
Windows 2008	C:\ProgramData\Electric Cloud\ElectricCommander\logs
Windows 7	
UNIX	/opt/electriccloud/electriccommander/logs

Web Interface Online Help System

Open the ElectricFlow online help system for more information. Click the **Help** link in the top-right corner of any product web page to see a help topic for that page.

When the help system opens, We recommend reviewing the Help table of contents. All Help folders above the Web Interface Help folder are user-guide style help topics that provide more detailed information on each of their subjects.

If you generally prefer to use a command-line tool rather than the ElectricFlow web interface, you will find complete ectool (the ElectricFlow command-line tool) and API (perl script) commands and options within the online help system too.

Chapter 12: Troubleshooting

This chapter contains troubleshooting procedures for some of the more common issues you might experience during the ElectricFlow installation process. More troubleshooting information can be found in ElectricFlow Knowledge Base articles located at <https://helpcenter.electric-cloud.com/hc/en-us/sections/200516863-Commander-KB>.

ElectricFlow Server is Unresponsive and Displays an OutOfMemory Error

Description:

The ElectricFlow server becomes unresponsive and displays an `OutOfMemory` error message that states that the server is out of `PermGen` space. This message occurs when the server is not configured properly for a 64-bit JVM.

Workaround:

Change a setting in `wrapper.conf`.

1. Locate the `wrapper.conf` file.

- The file to edit on Linux is:

```
/opt/electriccloud/electriccommander/conf/wrapper.conf
```

- The file to edit on Windows is:

```
<ElectricCommander data dir>/conf/wrapper.conf
```

- On Windows 7 or 2008, the `DATA_DIRECTORY` is typically:

```
C:\ProgramData\Electric Cloud\ElectricCommander
```

2. Add the following line to the end of the existing “`wrapper.java.additional`” section, being careful to use the next consecutive number.

In the following example, 7 was the last pre-existing sequence number in this section.

```
wrapper.java.additional.8=-XX:MaxPermSize=128m
```

Note: We recommend setting the Java `MaxPermSize` to a number greater than the default (which is 84m) on an 64-bit ElectricFlow server (java) installation.

3. After the file has been edited, restart the server to utilize the new value:

- On Linux: `/etc/init.d/commanderServer restart`
- On Windows: use “Services”

Windows PHP Does not Handle Time Zones Correctly

Description

PHP does not handle certain operating system time zones correctly on a Windows system. If the web server is running on a machine set for one of these time zones, users connected to that web server will see all times displayed as UTC times, instead of the web server time zone.

Workaround:

In the `config.php` file, you must explicitly set the PHP “`timezone_identifier`”.

To set the timezone:

1. Edit the following file.

```
C:\Program Files\Electric  
Cloud\ElectricCommander\apache\htdocs\commander\config.php
```

2. Add the following line anywhere between the opening and closing PHP tags:

```
date_default_timezone_set("<timezone_identifier>");
```

For example:

To set the timezone for Taipei, you would add: `date_default_timezone_set("Asia/Taipei");`

Note: For a complete "List of Supported Timezones", see <http://us2.php.net/manual/en/timezones.php>

ElectricFlow Server Certificate Fails Security Scan

Description:

You might need to replace the ElectricFlow server certificate if it fails the security scan. There are three relevant configuration entries in the `server/conf/commander.properties` file:

```
COMMANDER_HTTPS_PORT=8443  
COMMANDER_KEYSTORE=file:conf/keystore  
COMMANDER_KEYSTORE_PASSWORD=abcdef
```

Where:

- `COMMANDER_HTTPS_PORT` configures the SSL port
- `COMMANDER_KEYSTORE` is the location of the java keystore where the ElectricFlow HTTP server finds its host certificate
- `COMMANDER_KEYSTORE_PASSWORD` is the password to the keystore

Workaround:

Follow these steps to generate and inject a self-signed certificate for 1 year.

1. Back up the keystore file.
2. Delete the original key.

```
user@USER /cygdrive/c/ProgramData/Electric Cloud/ElectricCommander/conf
```

```
$ "c:/Program Files/Electric Cloud/ElectricCommander/jre/bin/keytool" -delete -a
lias jetty -keystore keystore -keypass passkey
Enter keystore password: abcdef
```

3. Generate and inject a new certificate.

```
user@USER /cygdrive/c/ProgramData/Electric Cloud/ElectricCommander/conf
$ "c:/Program Files/Electric Cloud/ElectricCommander/jre/bin/keytool" -keystore
keystore -alias jetty -genkey -keyalg RSA -sigalg MD5withRSA -validity 365
Enter keystore password: abcdef
What is your first and last name?
[Unknown]: localhost
What is the name of your organizational unit?
[Unknown]: <Enter>
What is the name of your organization?
[Unknown]: <Enter>
What is the name of your City or Locality?
[Unknown]: <Enter>
What is the name of your State or Province?
[Unknown]: <Enter>
What is the two-letter country code for this unit?
[Unknown]: <Enter>
Is CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correc
t?
[no]: yes
Enter key password for <jetty>
(RETURN if same as keystore password): <Enter>
```

4. Restart the server.

Your new certificate will look similar to this:

```
user@USER /cygdrive/c/ProgramData/Electric Cloud/ElectricCommander/conf
$ "c:/Program Files/Electric Cloud/ElectricCommander/jre/bin/keytool" -list -v -
keystore keystore_orig -keypass passkey
Enter keystore password: abcdef
Keystore type: JKS
Keystore provider: SUN
Your keystore contains 1 entry
Alias name: jetty
Creation date: Jan 31, 2012
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
Issuer: CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown
Serial number: 4f28603f
Valid from: Tue Jan 31 13:42:23 PST 2012 until: Wed Jan 30 13:42:23 PST 2013
Certificate fingerprints:
```

```
MD5: 38:50:CD:29:8C:16:3A:78:29:0F:45:56:E0:CA:42:D9
SHA1: 9B:A3:E4:EA:A7:C0:3A:ED:BF:63:24:18:F0:08:78:22:59:85:BC:8A
Signature algorithm name: MD5withRSA
Version: 3
*****
*****
```

Linux Upgrade Breaks Symbolic Links

Description:

When using the Linux installer to perform an upgrade, you might encounter problems moving broken symbolic links. You might see errors that begin with a line similar to, "could not read "/opt/electriccloud/electriccommander/workspace/FileOperationsLinux-LocalMove-7689/fileSymLink": no such file or directory".

Workaround:

Manually remove the file and rerun the installer if you encounter these types of errors.

Chapter 13: Installing ElectricFlow Agents

This document describes how to install 32-bit and 64-bit ElectricFlow agents on Windows and Linux. An agent is an ElectricFlow component that runs on a machine resource. It executes ElectricFlow job steps, monitors step progress, and records job completion information.

Graphical User Interface Installation Methods

The graphical user interface installation methods are supported by Windows platforms and Linux platforms running the X Window System.

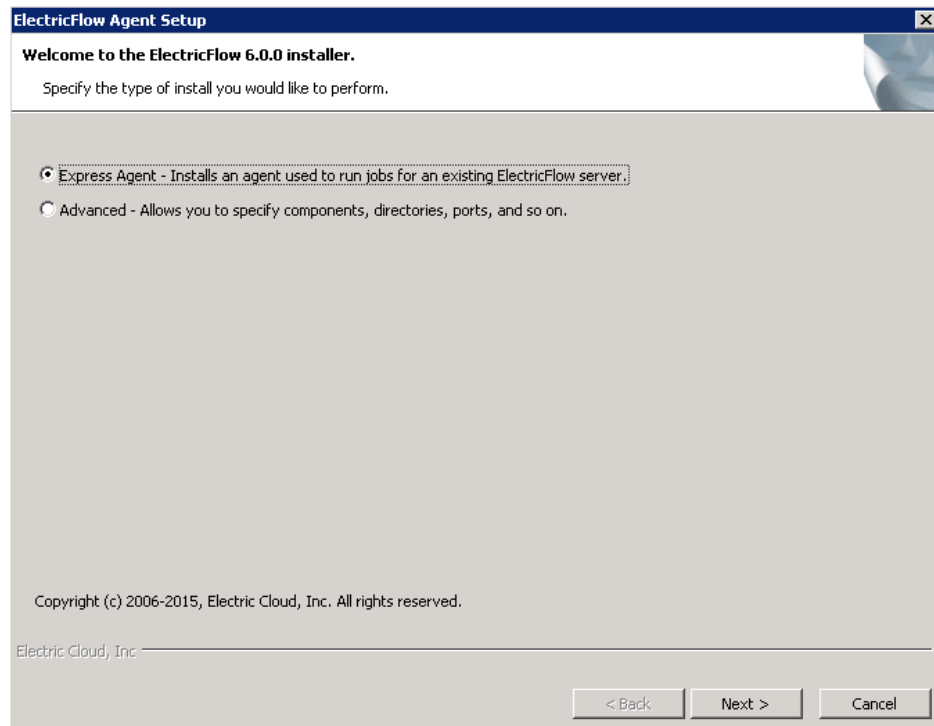
Running an Express Agent Graphical User Interface Installation (Agent-Only Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. You can install a 32-bit agent or a 64-bit agent. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

Note: You might install ElectricFlow agent software on Windows or Linux with this installation method. For Solaris, HP-UX, Mac, or other supported UNIX agent machines, see [Non-Server Platform Agent Installation Method](#) on page 13-24.

1. Double-click the `ElectricFlowAgent-x86-<version>` file (for a 32-bit agent) or the `ElectricFlowAgent-x64-<version>` file (for a 64-bit agent).

The Welcome to the ElectricFlow Installer screen appears.



Note: The screen examples in this procedure are from a Windows system. Different options might appear on a Linux system.

2. Select the **Express Agent** installation option, and then click **Next** to continue.

The Remote ElectricFlow Server screen appears.

ElectricFlow Setup

Remote ElectricFlow server

Specify an existing ElectricFlow server and what you would like to retrieve from it.

Server Host Name

ElectricFlow User Name

Password

☒ Discover the plugins directory

☒ Create a resource

☐ Trusted (restrict to one server)

Resource Name

☐ Create a repository

Repository Name

☒ Create in default zone

Agent Gateway URL

Zone Name

Electric Cloud, Inc.

< Back Next > Cancel

3. Complete the following information on the Remote ElectricFlow Server screen.

- **Server Host Name** - Use this field to enter the name of the ElectricFlow server that will communicate with this agent. If the remote server is using a non-default HTTP port, you must specify the Server Host Name as `host:port`.
- **ElectricFlow User Name** - Use this field to enter the name of an ElectricFlow user on the ElectricFlow server who has sufficient privileges to create a resource. This field defaults to the ElectricFlow-supplied `admin` user.
- **Password** - Use this field to enter the password for the ElectricFlow user. The default password for the `admin` user is `changeme`.
- **Discover the plugins directory** - Select this check box if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

Note: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10

- **Create a resource** - Select this check box if you want to create a resource on the remote ElectricFlow server for the agent you are installing.
- **Trusted** - Select this check box to restrict this agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.
- **Resource Name** - Use this field to enter the name of the resource you would like to use for the agent. This field is available for use when the Create a resource check box is selected.
- **Create in default zone** - Select this check box if you want to create the agent in the default zone.

- **Agent Gateway URL** - Use this field to enter the URL of the gateway used to communicate with the ElectricFlow server. This field is available for use when the Create in default zone check box is cleared.
 - **Zone Name** - Use this field to enter the name of the zone used during remote agent and/or remote repository creation. This field is available for use when the Create in default zone check box is cleared.
4. Click **Next** to continue.

The Agent Service Account screen appears.

The screenshot shows a Windows-style dialog box titled "ElectricFlow Setup". Inside, the section "Agent service account" is active, with the instruction "Specify the account you would like the agent service to run as." Below this are three input fields: "User Name" (containing "build"), "Password" (empty), and "Domain - Leave blank for local users" (empty). There is a checkbox labeled "Use the local system account" which is currently unchecked. At the bottom left is the text "Electric Cloud, Inc.". At the bottom right are three buttons: "< Back", "Next >", and "Cancel".

5. Select the appropriate steps for your platform and complete the following information on the screen.
- If you have a Windows system:
 - **User Name** - Use this field to enter the name of the user who will run the ElectricFlow agent service.

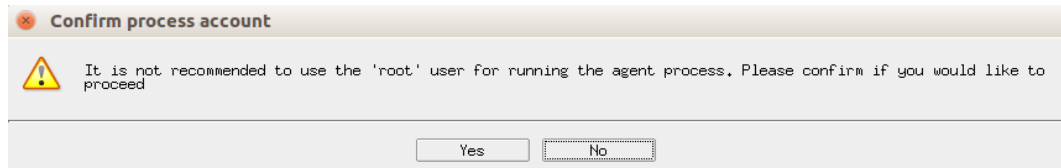
The user that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory.
 - **Password** - Use this field to enter the password of the user who will run the ElectricFlow agent service.
 - **Domain** - Use this field to enter the domain name information for the user. For example, `electric-cloud.com`. Leave this field blank if this is a local user.
 - **Use the local system account** - Select this check box if you want the ElectricFlow agent service to run as the local Windows system account.

Note: The local system account cannot access network resources such as shared filesystems used for plugins or workspaces.

- If you have a Linux system:

- **User Name** - Use this field to enter the name of the user who owns the ElectricFlow agent process.

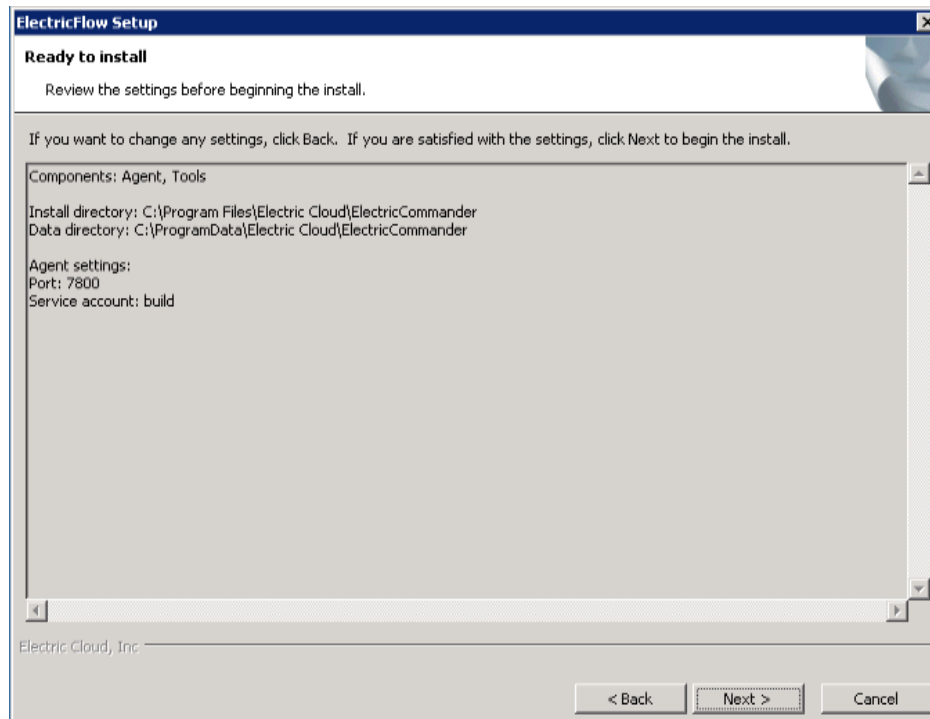
The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, click **Yes** when the following confirmation appears:



- **Group Name** - Use this field to enter the name of the group who owns the ElectricFlow agent process.

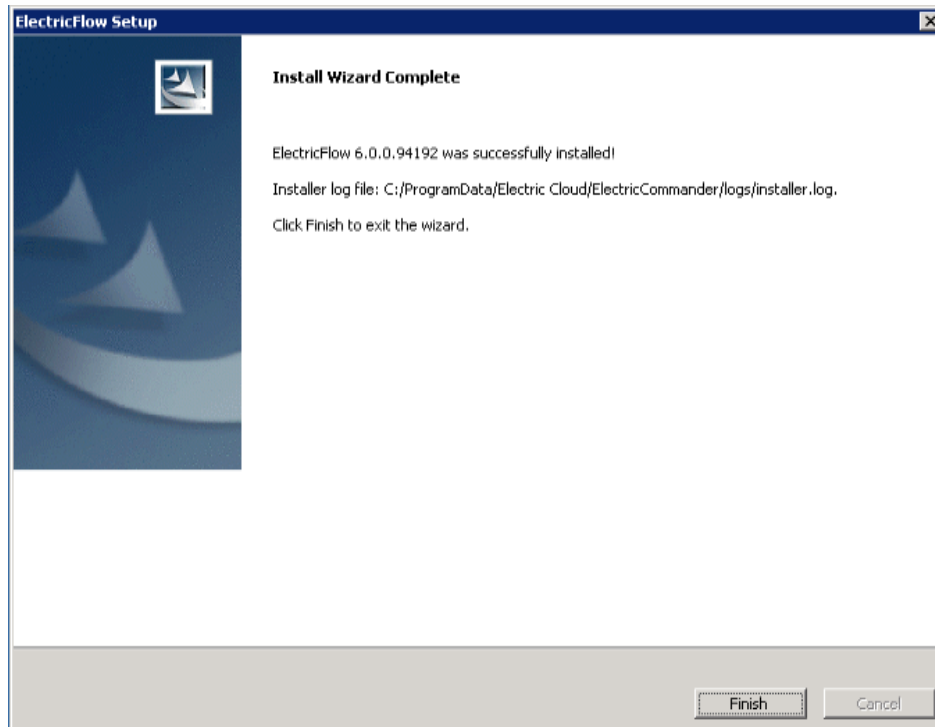
6. Click **Next** to continue.

The Ready to Install Screen appears.



7. Review this screen to verify your selections. Use the **Back** button to change any of your settings if necessary.
8. Click **Next** to continue.

ElectricFlow installs the agent and tools components. This process can take a few minutes.



9. Click **Finish** to complete the installation.

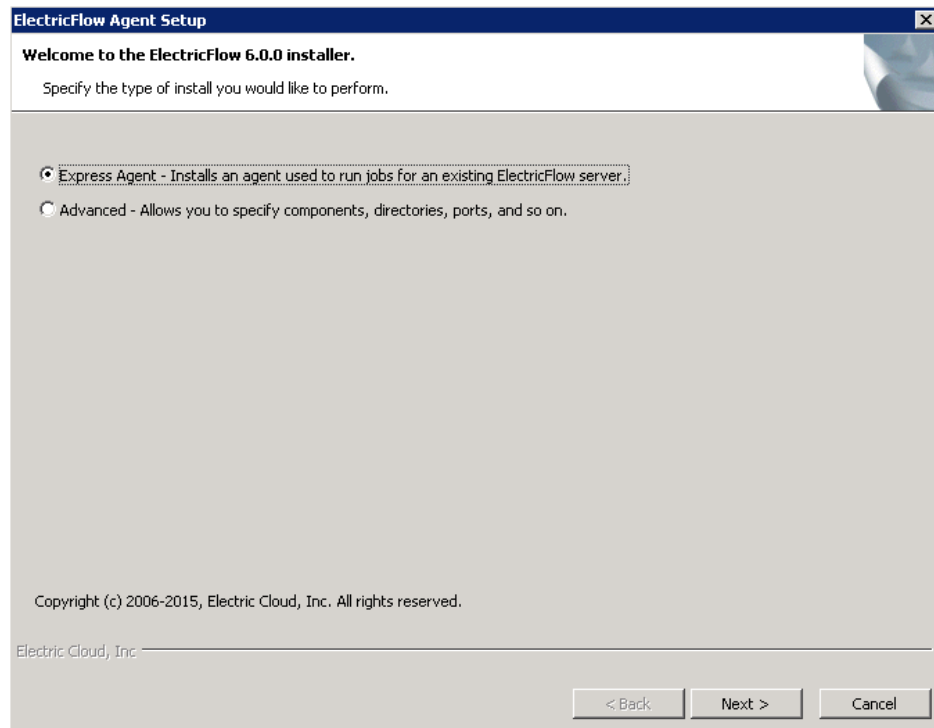
Running an Advanced Agent Graphical User Interface Installation (Agent-Only Installer)

The ElectricFlow agent software must be installed on each agent machine you intend to use with ElectricFlow. You can install a 32-bit agent or a 64-bit agent. Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

Note: You might install ElectricFlow agent software on Windows or Linux with this installation method. For Solaris, HP-UX, Mac, or other supported UNIX agent machines, see [Non-Server Platform Agent Installation Method](#) on page 13-24.

1. Double-click the `ElectricFlowAgent-x86-<version>` file (for a 32-bit agent) or the `ElectricFlowAgent-x64-<version>` file (for a 64-bit agent).

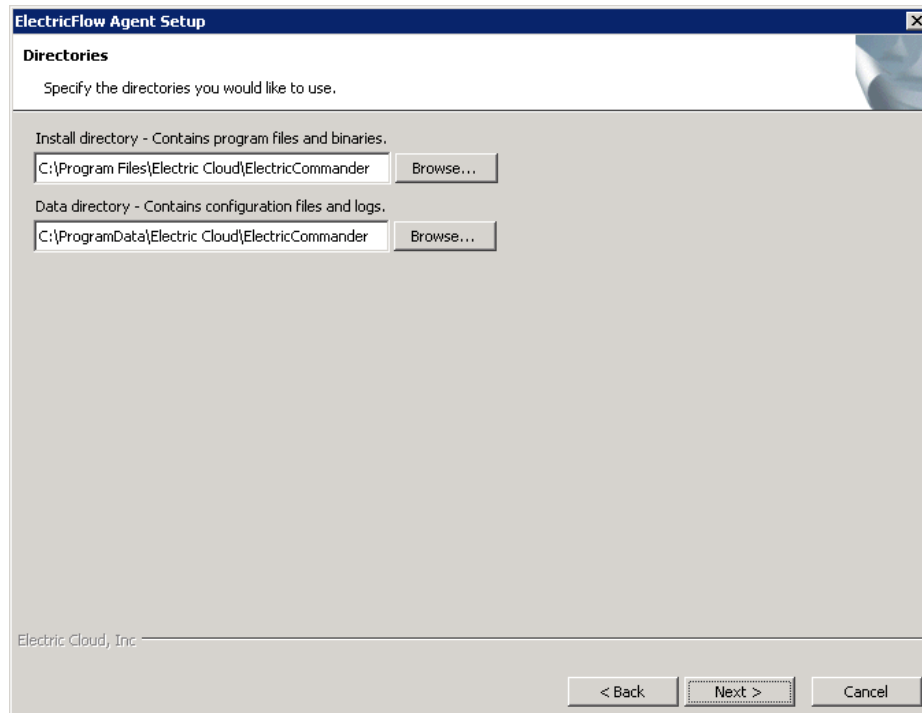
The Welcome to the ElectricFlow Installer screen appears.



Note: The screen examples in this procedure are from a Windows system. Different options might appear on a Linux system.

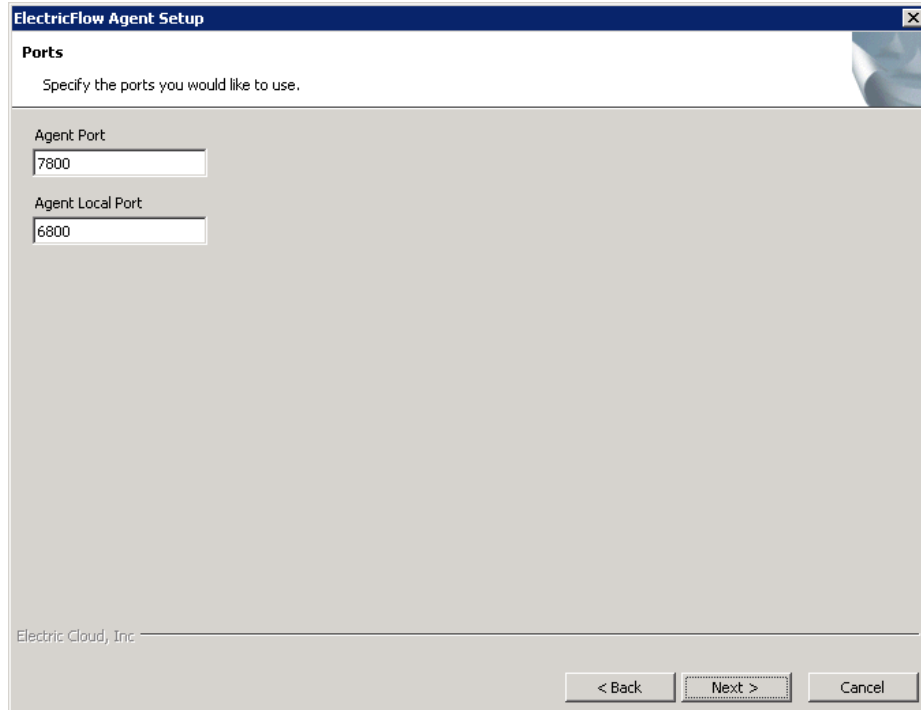
2. Select the **Advanced Agent** installation option, and then click **Next** to continue.

The Directories screen appears.



3. Complete the following information on the Directories screen.
 - **Install directory** - Use this field to enter a new installation directory path for program files and binaries.
 - **Data directory** - Use this field to enter a new installation directory path for configuration files and logs.
4. Click **Next** to continue.

The Ports screen appears.

The screenshot shows a window titled "ElectricFlow Agent Setup" with a "Ports" tab. The instruction "Specify the ports you would like to use." is at the top. Below it are two text input fields: "Agent Port" with the value "7800" and "Agent Local Port" with the value "6800". At the bottom left is the text "Electric Cloud, Inc.". At the bottom right are three buttons: "< Back", "Next >", and "Cancel".

ElectricFlow Agent Setup

Ports

Specify the ports you would like to use.

Agent Port
7800

Agent Local Port
6800

Electric Cloud, Inc.

< Back Next > Cancel

5. Complete the following information on the Ports screen.
 - **Agent port** - Use this field to specify a different port to eliminate any conflicts with your existing system configuration.
 - **Agent local port** - Use this field to specify a different port to be used by the agent for HTTP communication on the localhost network interface.
6. Click **Next** to continue.

The Remote ElectricFlow Server screen appears.

ElectricFlow Setup

Remote ElectricFlow server

Specify an existing ElectricFlow server and what you would like to retrieve from it.

Server Host Name:

ElectricFlow User Name:

Password:

☒ Discover the plugins directory

☒ Create a resource

☐ Trusted (restrict to one server)

Resource Name:

☐ Create a repository

Repository Name:

☒ Create in default zone

Agent Gateway URL:

Zone Name:

Electric Cloud, Inc.

< Back Next > Cancel

7. Complete the following information on the Remote ElectricFlow Server screen.

- **Server Host Name** - Use this field to enter the name of the ElectricFlow server that will communicate with this agent. If the remote server is using a non-default HTTP port, you must specify the Server Host Name as `host:port`.
- **ElectricFlow User Name** - Use this field to enter the name of an ElectricFlow user on the ElectricFlow server who has sufficient privileges to create a resource. This field defaults to the ElectricFlow-supplied `admin` user.
- **Password** - Use this field to enter the password for the ElectricFlow user. The default password for the `admin` user is `changeme`.
- **Discover the plugins directory** - Select this check box if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

Note: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10

- **Create a resource** - Select this check box if you want to create a resource on the remote ElectricFlow server for the agent you are installing.
- **Trusted** - Select this check box to restrict this agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.
- **Resource Name** - Use this field to enter the name of the resource you would like to use for the agent. This field is available for use when the Create a resource check box is selected.
- **Create in default zone** - Select this check box if you want to create the agent in the default zone.

- **Agent Gateway URL** - Use this field to enter the URL of the gateway used to communicate with the ElectricFlow server. This field is available for use when the Create in default zone check box is cleared.
 - **Zone Name** - Use this field to enter the name of the zone used during remote agent and/or remote repository creation. This field is available for use when the Create in default zone check box is cleared.
8. Click **Next** to continue.

The Agent Service Account screen appears.

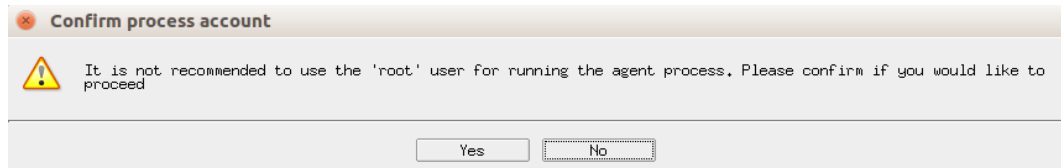
9. Select the appropriate steps for your platform and complete the following information on the screen.
- If you have a Windows system:
 - **User Name** - Use this field to enter the name of the user who will run the ElectricFlow agent service.
The user that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory.
 - **Password** - Use this field to enter the password of the user who will run the ElectricFlow agent service.
 - **Domain** - Use this field to enter the domain name information for the user. For example, `electric-cloud.com`. Leave this field blank if this is a local user.
 - **Use the local system account** - Select this check box if you want the ElectricFlow agent service to run as the local Windows system account.

Note: The local system account cannot access network resources such as shared filesystems used for plugins or workspaces.

- If you have a Linux system:

- **User Name** - Use this field to enter the name of the user who owns the ElectricFlow agent process.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, click **Yes** when the following confirmation appears:

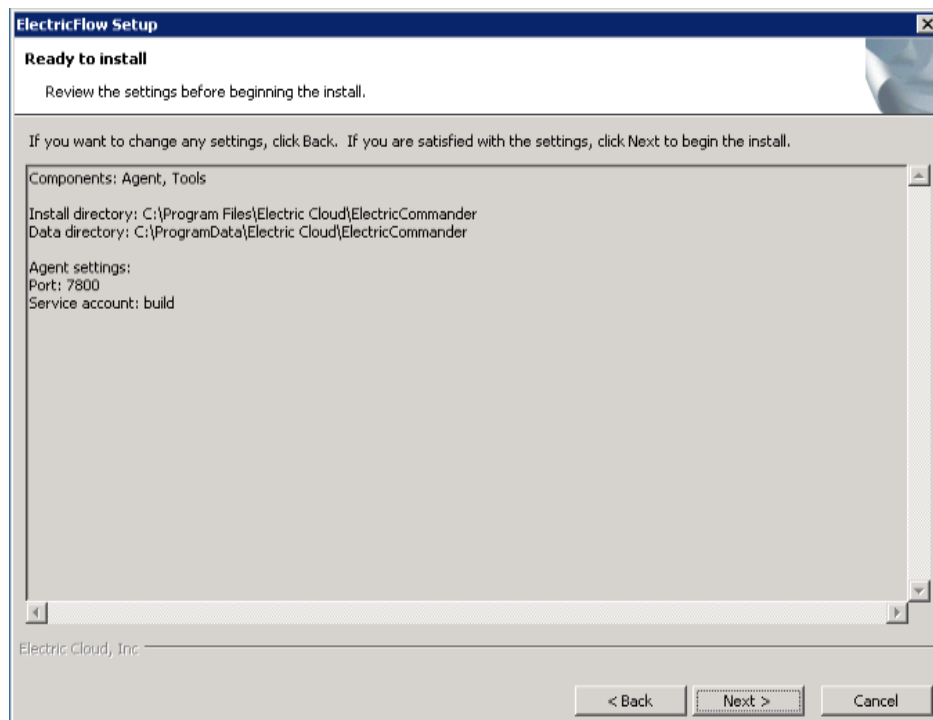


- **Group Name** - Use this field to enter the name of the group who owns the ElectricFlow agent process.

10. Select the appropriate steps for your platform and complete the following information on the screen.

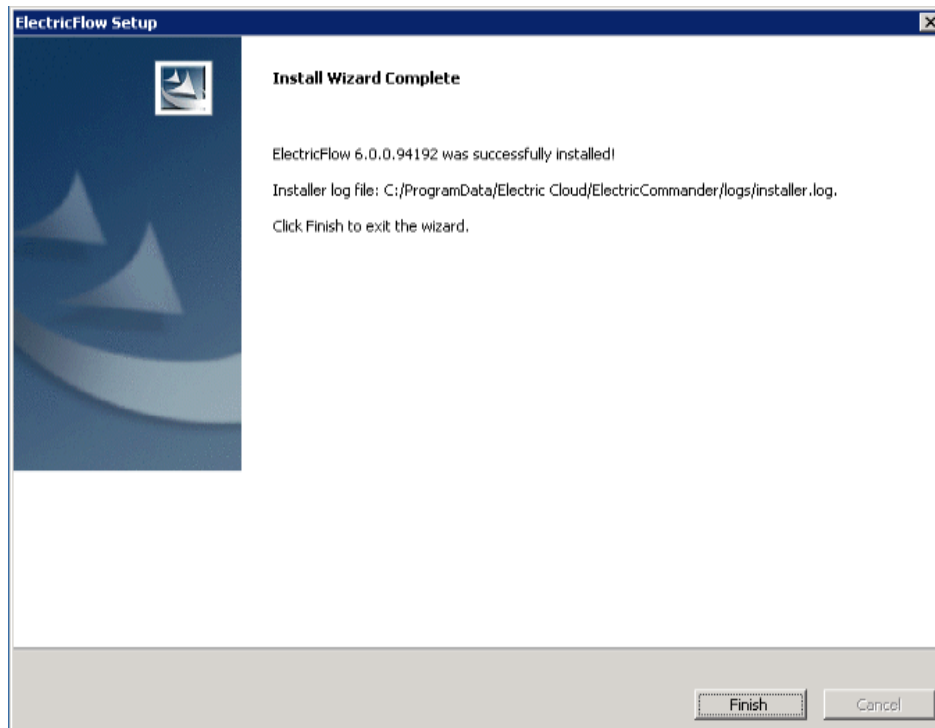
11. Click **Next** to continue.

The Ready to Install Screen appears.



12. Review this screen to verify your selections. Use the **Back** button to change any of your settings if necessary.
13. Click **Next** to continue.

ElectricFlow installs the agent and tools components. This process can take a few minutes.



14. Click **Finish** to complete the installation.

Interactive Command-Line Installation Methods

The interactive command-line installation methods are only supported for Linux-only installs on a local Linux volume. Electric Cloud does not support installing the ElectricFlow server on a network volume.

Note: You might install ElectricFlow agent software on Linux with this installation method. For Solaris, HP-UX, Mac, or other supported UNIX agent machines, see [Non-Server Platform Agent Installation Method](#) on page 13-24.

Running an Express Agent Command-Line Installation

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:

- If you have a Linux platform, enter: `./ElectricFlowAgent-<version>`
- If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface. To override this behavior, enter: `./ElectricFlow--<version> --mode console`

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow on your computer. Continue? [n/Y]
```

2. Continue the installation by entering `y`.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressServer,  
expressAgent, or advanced. [expressServer]
```

3. Enter: `expressAgent`.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686  
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.  
Install them? [n/Y]
```

4. Enter `y`.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

5. Enter `y` if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

```
Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]
```

6. Enter `y` to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for **pre-existing** ElectricFlow servers.

The software displays the following message.

```
Register as trusted agent (required for gateway)? [y/N]
```

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

7. Choose one of the following options:

- If a gateway used to communicate with the ElectricFlow server, you must select `y`. This option allows you to create a trusted network connection between the agent and server under the same certificate authority. This will allow the agent and the ElectricFlow server to communicate across the network.
- If there is no gateway between the agent and ElectricFlow server, enter `n`.

IMPORTANT: If you deviated from the recommended agent options, you will see variations in the installation options that appear on your system.

The software displays the following message.

```
Create repository and/or agent in the default zone? [n/Y]
```

7. Enter `y` to create the agent in the default zone.

The software displays the following message.

```
Specify the host:port of a remote ElectricFlow server the agent, repository
server and/or web server being installed can link to. The port is only required
if it is not the default. [] <hostName:port>
```

8. Enter the Server Host Name of the ElectricFlow server that will communicate with this agent. You must specify the Server Host Name as `hostName:port` if the remote server is using a non-default HTTP port.

The software displays the following message.

```
Specify the user name with which to login to <hostName:port>. [admin]
```

9. Enter the user name of a user on the ElectricFlow server who has sufficient privileges to create a resource. The default is the ElectricFlow-supplied `admin` user.

The software displays the following message.

```
Specify the password for "<electricflow_user>" on <hostName:port>. []
```

10. Enter the password for the ElectricFlow user. The default password for the `admin` user is `changeme`.

The software displays the following message.

```
Specify the name of the resource to create on <hostName:port>. [<resource_name>]
```

11. Enter the following information if the agent must be registered as a trusted agent. These options only appear if you entered `y` for `Register as trusted agent (required for gateway)? [y/N]`.

- a. Enter a resource name to use on the ElectricFlow server.

The software displays the following message.

```
Specify the agent gateway URL in the form of 'ipOrHostname:port' []
```

- b. Enter an agent gateway URL. This is the URL of the gateway used to communicate with the ElectricFlow server.

The software displays the following message.

```
Specify the zone name for the agent and/or repository []
```

- c. Enter the Zone Name. This is the zone used during remote agent and/or remote repository creation.

The software displays the following message.

```
Specify the user the agent will run as. []
```

12. Enter a user name. This is the user who owns the ElectricFlow agent process. For example, you might enter `build`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.  
Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

13. Enter a Group Name. This is the group that owns the ElectricFlow agent process. For example, you might enter `build`.

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line `"ElectricFlow <version> was successfully installed!"` appears.

Running an Express Agent Command-Line Installation (Agent-Only Installer)

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:
 - If you have a Linux platform, enter `./ElectricFlowAgent-x86-<version>` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version>` (for a 64-bit agent).
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface.

To override this behavior, enter `./ElectricFlow-x86-<version> --mode console` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version> --mode console` (for a 64-bit agent).

The software displays the following message.

Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.

This will install ElectricFlow Agent on your computer. Continue? [n/Y]

2. Continue the installation by entering `y`.

The software displays the following message.

Specify the type of setup you would like to perform: `expressAgent` or `advanced`.
[`expressAgent`]

3. Enter: `expressAgent`.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

Discover the plugins directory from a remote ElectricFlow server? [n/Y]

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

The following libraries are missing on your system: `libstdc++.i686`
`nss-pam-ldapd.*i686` . They are required for correct work of ElectricFlow.
Install them? [n/Y]

4. Enter `y`.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

Discover the plugins directory from a remote ElectricFlow server? [n/Y]

5. Enter `y` if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]

6. Enter `y` to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for **pre-existing** ElectricFlow servers.

The software displays the following message.

Register as trusted agent? [y/N]

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development systems.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected

by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

7. Choose one of the following options:

- If a gateway is used to communicate with the ElectricFlow server, you must select `y`. This option allows you to create a trusted network connection between the agent and server under the same certificate authority. This will allow the agent and the ElectricFlow server to communicate across the network.
- If there is no gateway between the agent and ElectricFlow server, enter `n`.

IMPORTANT: If you deviated from the recommended agent options, you will see variations in the installation options that appear on your system.

The software displays the following message.

```
Specify the user the agent will run as. []
```

7. Enter a user name. This is the user who owns the ElectricFlow agent process. For example, you might enter `build`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.  
Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

8. Enter a Group Name. This is the group that owns the ElectricFlow agent process. For example, you might enter `build`.

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line "ElectricFlow <version> was successfully installed!" appears.

Running an Advanced Agent Command-Line Installation (Agent-Only Installer)

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:

- If you have a Linux platform, enter `./ElectricFlowAgent-x86-<version>` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version>` (for a 64-bit agent).
- If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface.

To override this behavior, enter `./ElectricFlow-x86-<version> --mode console` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version> --mode console` (for a 64-bit agent).

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow on your computer. Continue? [n/Y]
```

2. Continue the installation by entering `y`.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressAgent or advanced.  
[expressAgent]
```

3. Enter: `advanced`.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Specify the install directory (for program files and binaries).  
[/opt/electriccloud/electriccommander]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686  
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.  
Install them? [n/Y]
```

4. Enter `y`.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Specify the install directory (for program files and binaries).  
[/opt/electriccloud/electriccommander]
```

5. Enter a new installation directory path for program files and binaries.

The software displays the following message.

```
Specify the data directory (for configuration files and logs).  
[/opt/electriccloud/electriccommander]
```

6. Enter a new installation directory path for configuration files and logs.

The software displays the following message.

```
Specify the agent port. [7800]
```

7. Enter a different port to eliminate any conflicts with your existing system configuration.

The software displays the following message.

Specify the agent local port. [6800]

8. Enter a different port to be used by the agent for HTTP communication on the localhost network interface.

The software displays the following message.

Discover the plugins directory from a remote ElectricFlow server? [n/Y]

9. Enter *y* if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]

10. Enter *y* to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for **pre-existing** ElectricFlow servers.

The software displays the following message.

Register as trusted agent? [y/N]

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development environments.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

11. Choose one of the following options:

- If a gateway is used to communicate with the ElectricFlow server, you must select *y*. This option allows you to create a trusted network connection between the agent and server under the same certificate authority. This will allow the agent and the ElectricFlow server to communicate across the network.
- If there is no gateway between the agent and ElectricFlow server, enter *n*.

IMPORTANT: If you deviated from the recommended agent options, you will see variations in the installation options that appear on your system.

The software displays the following message.

Specify the user the agent will run as. []

11. Enter a user name. This is the user who owns the ElectricFlow agent process. For example, you might enter `build`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.
Please confirm if you would like to proceed [y/N]
```

The software displays the following message.

```
Specify the group the agent will run as. []
```

12. Enter a Group Name. This is the group that owns the ElectricFlow agent process. For example, you might enter `build`.

The ElectricFlow software is installed on the machine. When the installation completes successfully, a message that contains the line "ElectricFlow <version> was successfully installed!" appears.

Running an Express Agent Command-Line Installation (Agent-Only Installer) When the Server Uses Registered and Concurrent Licenses

Use this procedure when the ElectricFlow server uses a mix of registered and concurrent licenses.

The agent software must be installed on each machine you intend to use with ElectricFlow. An agent is an ElectricFlow component that runs on a machine resource. The agent executes ElectricFlow job steps, monitors step progress, and records job completion information.

Review [Before You Install ElectricFlow](#) on page 3-5 before performing this procedure.

1. Choose one of the following commands to start the installer:
 - If you have a Linux platform, enter `./ElectricFlowAgent-x86-<version>` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version>` (for a 64-bit agent).
 - If you have a Linux platform with the X Window System, the installer will automatically bring up the graphical user interface. To override this behavior, enter `./ElectricFlow-x86-<version> --mode console` (for a 32-bit agent) or `./ElectricFlowAgent-x64-<version> --mode console` (for a 64-bit agent).

The software displays the following message.

```
Copyright (c) 2006-2015, Electric Cloud, Inc. All rights reserved.
```

```
This will install ElectricFlow Agent on your computer. Continue? [n/Y]
```

2. Enter `y` to continue the installation.

The software displays the following message.

```
Specify the type of setup you would like to perform: expressAgent or advanced.
[expressAgent]
```

3. Enter: `expressAgent`.

- If you are installing ElectricFlow on a 64-bit machine and the 32-bit libraries are already installed, go to Step 5 when the software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

- If you are installing ElectricFlow on a 64-bit machine but the 32-bit libraries required for ElectricFlow to run correctly are not installed, the software displays the following message:

```
The following libraries are missing on your system: libstdc++.i686
nss-pam-ldapd.*i686 . They are required for correct work of ElectricFlow.
Install them? [n/Y]
```

4. Enter `y`.

The missing 32-bit libraries are installed.

Note: If you enter `n`, the installation process stops and the installer exits.

The software displays the following message.

```
Discover the plugins directory from a remote ElectricFlow server? [n/Y]
```

5. Enter `y` if you want the agent machine to have access to the plugins directory. We recommend that you allow access to the plugins directory so agents have access to collections of features, third-party integrations, or third-party tools.

IMPORTANT: The plugins directory on the ElectricFlow server must be “shared” before the agent machine can use “discover” to find the directory. For more information, see [Universal Access to the Plugins Directory](#) on page 5-10.

The software displays the following message.

```
Create a resource for the installed agent on a remote ElectricFlow server? [n/Y]
```

6. Enter `y` to automatically create a resource object for the agent on a remote ElectricFlow server. This option is recommended to save time configuring new ElectricFlow resources for *pre-existing* ElectricFlow servers.

The software displays the following message.

```
Register as trusted agent? [y/N]
```

Note: Making an agent trusted restricts the agent to one ElectricFlow server. The agent will not respond to incoming communication from any other ElectricFlow server. This is useful when you want to create a secure production environment, but generally not needed for test or development environments.

IMPORTANT: Gateways *may be* run without trusted agents. However, we strongly recommend that you use gateways with trusted agents to prevent security issues in the firewall between zones connected by a gateway.

There are exceptions to using gateways without trusted agents:

- The firewall between two zones is not required in your environment or is needed only to protect the ElectricFlow server.
- There is a specific reason to use gateways without trusted agents, such as a requirement to prevent unauthorized users from accessing your network. All incoming traffic from the internet is routed to a data center through a load balancer, and the load balancer routes the traffic to the appropriate machine in your network.

7. Enter `n` if you are installing the ElectricFlow Community Edition.

The software displays the following message.

```
Create repository and/or agent in the default zone? [y/n]
```

8. Enter `y`.

The software displays the following message.

Specify the host:port of a remote ElectricFlow server that the agent, repository server and/or web server being installed can link to. The port is only required if it is not the default. []

9. Enter the `<host:port>`.

The software displays the following message.

Specify the user name with which to login to "`<host:port>`". [admin]

10. Enter `admin`.

The software displays the following message.

Specify the password for "admin" on "`<host:port>`". []

11. Enter a password.

The software displays the following message.

Specify the name of the resource to create on "`<host:port>`". []

12. Enter a resource name.

The software displays the following message.

Specify resource type for remote server: Registered or Concurrent. []

13. Enter `Registered`.

The software displays the following message.

Specify the user the agent will run as. []

14. Enter a user name.

This is the user who owns the ElectricFlow agent process. For example, you can enter `deploy`.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

It is not recommended to use the 'root' user for running the agent process.
Please confirm if you would like to proceed [y/N]

The software displays the following message.

Specify the group the agent will run as. []

15. Enter a group name.

This is the group that owns the ElectricFlow agent process. For example, you can enter `deploy`.

The ElectricFlow software is installed on the machine.

When the installation completes successfully, a message that contains the line "ElectricFlow `<version>` was successfully installed!" appears.

Non-Server Platform Agent Installation Method

To install agents and tools on machines that are not supported ElectricFlow server platforms, you must use the appropriate UNIX installer instead of the `./ElectricFlow-<version>` installer file. For more information about supported agent platforms, see [Supported Agent Platforms](#) on page 2-10.

You must use the `commander_<OSType>.bin` file to install the ElectricFlow agent software on a Solaris, HP-UX, or Mac OS agent machine. You can find copies of the various agent UNIX installers on the Electric Cloud FTP site.

Installing Agents with the UNIX Installer

Use the `commander_<OSType>.bin` file to install the ElectricFlow agent software on a Solaris, HP-UX, or Mac OS agent machine.

1. Obtain a copy of `commander_<OSType>.bin`
2. Log in as **root**.
3. Run `chmod +x ./commander_<OSType>.bin` to ensure the installer is executable.
4. Run `./commander_<OSType>.bin` to begin the ElectricFlow agent installation on a UNIX machine.

Where `<OSType>` is the agent platform.

The following information is displayed.

```
Checking installer integrity, please wait...
ElectricCommander 6.0.0.94439 for Linux Installer
Copyright 2006-2015 Electric Cloud, Inc. All rights reserved.
```

```
Press CTRL-C to exit at any time.
```

```
Press Enter to accept default settings.
```

```
log file: /tmp/commander_install_20150713_151337.log
This suite installer can install several different product options.
Note: The default is to install everything.
Which products would you like to install (agent, tools): agent
```

5. Enter: agent

Note: When you enter the “agent” option, an agent and tools will be installed.

6. Press **Enter** to begin the ElectricFlow UNIX agent installation.

The following information is displayed.

```
Installing agent and tools.
```

```
Where would you like the software to be installed?
```

```
NOTE: The destination should NOT be an nfs filesystem.
```

```
If no directory is specified, by default an “electricCloud” directory is created
in /opt.
```

```
Enter destination directory (default is /opt): <directory path>
```

7. Enter the destination directory path.

The following information is displayed.

```
Enter a pre-existing user to own installed agent files and run agent processes:
```

8. Enter the name of the user who owns the ElectricFlow agent process.

The user/group that the agent runs as must have permission to write to the `$INSTALL_DIRECTORY/log` directory. If you specify `root`, enter `y` when the following confirmation appears:

```
It is not recommended to use the 'root' user for running the agent process.
Please confirm if you would like to proceed [y/N]
```

The following information is displayed.

```
Enter a pre-existing group to own installed agent files and run agent processes:
```

9. Enter the name of the group who owns the ElectricFlow agent process.

The following information is displayed.

```
Enter the agent port (default is 7800):
```

10. Accept the default port or specify a different port to eliminate any conflicts with your existing system configuration.
11. Press **Enter** to start the agent software installation.

When the installation is complete, the following information is displayed.

```
OK: Installation successful!
```

UNIX Agent Silent Installation

The UNIX installers also have a command-line interface for unattended (silent) installations. You can specify configuration options on the command line or in a configuration file.

UNIX Silent Installation Command Arguments

The following table lists the arguments for running a silent install with the UNIX installer.

<code>-q</code>	This option causes the installer to operate in silent mode. The default installation options will be used unless overridden in a configuration file or on the command line.
<code>-f</code>	This option forces the installer to remove and replace any existing files in the destination directory. This option does not uninstall the previous version, it completely removes the directory and writes a new one. For upgrades, see Roadmap to Upgrade ElectricFlow on page 6-1.
<code>--config</code>	This option can be used to specify a file containing configuration variables and values.

Running a Silent UNIX Installation

Use the following procedure to run a silent UNIX installation.

- Enter the following command to begin a silent install:

```
commander_<OSType>.bin -q <arguments>
```

Where:

- <OSType> is the agent platform.
- <arguments> represents any additional silent install arguments you add to the command.

The following example command-line string uses a configuration file to override default values on a silent install:

```
commander_sun4u_SunOS.bin -q -f --config myconfig
```

Example Configuration File Parameters

The following examples are parameter values to use for an unattended (silent) installation configuration file.

Agent Install Parameters

```
EC_INSTALL_TYPE=agent
DESTINATION_DIR="/opt"
AGENT_USER_TO_RUN_AS=(user for agent files and processes)
AGENT_GROUP_TO_RUN_AS=(group for agent)
EC_AGENT_PORT=(agent port number)
```

Tools Install Parameters

```
EC_INSTALL_TYPE=tools
DESTINATION_DIR="/opt"
USER_TO_RUN_AS="user"
GROUP_TO_RUN_AS="group"
```

Index

A

- access to plugins directory 5-10
- adjust
 - Linux swappiness 5-16
 - server memory use 2-15
- agent
 - certificates 11-10
 - platforms 2-10
- agent services, starting and stopping 11-12
- agents on non-server platforms 3-64, 13-24
- alternate database 11-10
- apply enterprise license 5-1
- architecture 1-4

B

- backup
 - complete backup 11-1
 - database dump 11-1
 - ElectricFlow server 11-2
 - methods 11-1
 - preparation for 11-2
- browser requirements 2-11

C

- certificates 11-10
- challenges solved 1-4
- change to enterprise license 5-1
- changing ElectricFlow database 11-10
- checksum utility 2-16
- chkconfig 11-11

- choosing the correct
 - install process 3-3
 - upgrade process 8-7, 9-5
- clustering 4-1
 - access ElectricFlow 4-27
 - additional availability 4-27
 - architecture 4-1
 - benefits 4-1
 - configuration 4-5
 - considerations 4-2
 - dependencies 4-4
 - failure points 4-27
 - load balancers 4-6
 - single-server mode 4-17
 - software 4-4
 - strategies 4-2
- command-line
 - database configuration 5-4
 - upgrade 8-12, 9-10
- configure
 - clustering 4-5
- configuring
 - alternate database 5-3
 - proxy agents 5-14
 - proxy servers 5-13
 - proxy settings 5-14

D

data backup methods 11-1

database

 dumps 11-1

 interactions 5-2

 requirements 2-13

 sizing 2-13

 user 5-2

database configuration

 command line 5-4

 web 5-3

database.properties file 5-2, 5-9

default

 installation directories 3-8, 8-1, 9-1

 ports 2-12

disk space use 2-13

E

ecconfigure

 set agents 5-14

 set servers 5-14

ectool, set database 5-4

ElectricFlow

 agent disk space 2-13

 backing up 11-2

 change database 11-10

 database user 5-2

 login 3-104

 memory use 2-15

 ports 2-12

 restoring 11-3

 server backups 11-1

 server disk space 2-13

 server restores 11-3

enterprise license 5-1

environment proxy servers 5-13

external database configuration 5-2

F

file descriptors 5-15

G

graphical user interface

 advanced agent install (standalone) 3-31, 13-6

 advanced install 3-13

 express agent install 3-21

 express agent install (standalone) 3-26, 13-1

 express server install 3-8

 installation methods 3-8, 13-1

H

hardware requirements 2-11

horizontal scalability

 agent separation 4-5, 4-26

 ClusterInfoTool 4-20

 getting server cluster information 4-20

 ZooKeeper import 4-18

I

installation

 directories 3-8, 8-1, 9-1

 non-server platform agents 3-64, 13-24

 options 3-3

 prerequisites 3-5

 processes 3-3

 silent unattended 3-53

 with graphical user interface 3-8, 13-1

 with interactive command-line 3-38, 13-13

 ZooKeeper 4-7

interactive command-line

 advanced install 3-40

 express agent install 3-43, 13-13

 express agent install (standalone) 3-46, 3-48,
 3-51, 13-16, 13-18, 13-21

 express server install 3-38

 installation methods 3-38, 13-13

L

licensing 2-16
Linux swappiness 5-16
log files 11-14
log into ElectricFlow 3-104

M

memory use 2-15
MySQL
 characteristics 7-80, 7-94
 JDBC driver 3-103

N

non-server platform
 agent installation methods 3-64, 13-24
 silent install 3-66, 13-25
 UNIX install 3-65, 13-24

O

online help 11-15
options for installation 3-3
overview
 upgrade process 8-4

P

platforms for agents 2-10
plugins directory 5-10
port use 2-12
prepare for database backup 11-2
prerequisites
 installation 3-5
 upgrade 8-5, 9-4
proxy agents 5-14
proxy server 5-13
 configuration 5-14
 test settings 5-14

R

Recommended 5-3
recommended disk space 2-13
replicating plugins directory 5-13

requirements

 browser 2-11
 database 2-13
 hardware 2-11

restoring ElectricFlow 11-3

run time log files 11-14

running silent install 3-53

S**server**

 backups 11-1
 local configuration 1-4
 platforms 2-1
 remote configuration 1-5
 restore methods 11-3

server services, starting and stopping 11-12

set database

 with ectool 5-4

setDatabaseConfiguration

 command options 5-8
 procedure 5-4

settings for memory use 2-15

silent install

 command arguments 3-54
 example Repository Server Linux 3-63
 example Repository Server Windows 3-64
 examples Linux 3-58
 examples Windows 3-61
 methods 3-53
 running 3-53

silent unattended upgrade 8-13, 9-11

sizing ElectricFlow database 2-13

software

 compatibility 2-10
 functionality 1-2
 licenses 2-16

SQL server authentication 5-4

start and stop

agent services 11-12

server services 11-12

supported

agent platforms 2-10

databases 2-13

server platforms 2-1

swappiness kernel parameter 5-16

switch to alternate database 11-10

T

test proxy server 5-14

troubleshooting 12-1

types of installations 3-3

U

uninstall ElectricFlow

UNIX 10-1

Windows 10-1

universal access to plugins 5-10

UNIX installer for agents 3-64, 13-24

untrusted certificates 11-10

upgrade

methods 8-4, 8-7, 9-5

prerequisites 8-5, 9-4

silent unattended 8-13, 9-11

with command-line 8-12, 9-10

with user interface 8-9, 9-7

user interface upgrade 8-9, 9-7

W

web interface

database configuration 5-3

Z

ZooKeeper 4-11

install 4-7