



INSTALLATION GUIDE

Axway Application Studio

Version 1.4



Copyright © 2016 Axway. All rights reserved.

This documentation describes the following Axway software:

Axway Application Studio 1.4

No part of this publication may be reproduced, transmitted, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of the copyright owner, Axway.

This document, provided for informational purposes only, may be subject to significant modification. The descriptions and information in this document may not necessarily accurately represent or reflect the current or planned functions of this product. Axway may change this publication, the product described herein, or both. These changes will be incorporated in new versions of this document. Axway does not warrant that this document is error free.

Axway recognizes the rights of the holders of all trademarks used in its publications.

The documentation may provide hyperlinks to third-party web sites or access to third-party content. Links and access to these sites are provided for your convenience only. Axway does not control, endorse or guarantee content found in such sites. Axway is not responsible for any content, associated links, resources or services associated with a third-party site.

Axway shall not be liable for any loss or damage of any sort associated with your use of third-party content.

Contents

Preface	5
About Application Studio	5
Who should use this guide	5
Axway 5 Suite reference solutions	5
Other documentation	6
Axway online	6
Accessibility	7
Product accessibility	7
Documentation accessibility	7
1 System requirements	8
Operating systems	8
Linux user	8
Databases	8
RAC note	8
Browsers	9
2 Prerequisites	10
Database prerequisites	10
MySQL	10
Oracle	11
Database case-sensitivity rules	11
License	12
3 Plan and install	13
Installation directory	13
Database	13
MySQL	13
Oracle	13
Shared secret	14
Ports	14
SSL certificate	14
Ready to install	15

4 Silent mode installation	16
5 Start and log on the first time	17
6 Install a cluster	19
Cluster architecture	19
Why cluster	20
Sticky sessions	20
Clustering guidelines	20
Licenses for cluster nodes	21
Other resources	21
7 Upgrade	22
Restore auto-start after upgrade	22
Prerequisites	22
What changes after upgrading	23
Upgrade steps	23
Upgrade nodes in cluster	24
8 Stop Application Studio	25
9 Uninstall	26
Glossary	27

Preface

This guide describes the requirements and tasks for installing and starting Axway Application Studio. It also provides information about upgrading and installing Application Studio in a cluster of multiple computers.

About Application Studio

Application Studio combines development and operations in one product. It is a rapid application-development environment for building web applications that implement business processes. It is also an application operations environment for business users.

Application Studio is rapid in the sense it uses model-driven, visual development approaches rather than code for much, if not all, of the application development process. It does involve software-development engineering, but lets developers work at a higher level of abstraction than when coding. Business processes are designed using sequences of activities that involve participants, represented as blocks laid out in swim lanes customary to **BPM**¹. Forms are created by dragging visual elements from a palette onto the form canvas.

Application Studio is built on Joget Workflow Enterprise Edition, a platform for building enterprise web applications.

Who should use this guide

This guide is for people who oversee planning and installation of Application Studio. This guide presumes you have knowledge of:

- Your company's business processes and practices
- Your company's hardware, software and IT policies
- The Internet, including use of a browser

Axway 5 Suite reference solutions

Application Studio, a **Unified Flow Management** product, is integral to all Axway 5 Suite reference solutions. These meld selected Axway products into seamlessly integrated systems:

¹Business process management (BPM) is the discipline of managing processes for improving business performance outcomes and operational agility. Processes span organizational boundaries, linking people, information flows, systems and other assets to create and deliver value to customers and constituents.

- Manged File Transfer to securely transfer data in one-to-one, one-to-many and many-to-many scenarios.
- B2B Integration to exchange, transform and process standardized business documents withing an enterprise's B2B community
- Data Flow Integration to provide services for standardizing the exchange of business data with internal and external partners.
- Financial Integration to support data transfers in finance channels such as SWIFT and EBICS and transformations of data in financial protocols.

Your organization might use this product in context of a reference solution. Find details about the product's role in documentation on the Axway Sphere support website at support.axway.com.

Other documentation

Refer to the Application Studio Help Center and the Joget knowledge base for information about installing, getting started and using Application Studio. The URLs are:

Application Studio Help Center

<http://app-studio-help-center.squarespace.com/>

You also can click Help Center in the user interface to access it.

Joget knowledge base

<http://dev.joget.org/community/display/KBv4/Joget+Workflow+v4+Knowledge+Base>

Application Studio uses Joget Workflow 4, and this URL opens version 4 of the knowledge base.

Axway online

Go to Axway Sphere at support.axway.com to contact a representative, learn about training programs, or download software, documentation and knowledge-base articles. Check Sphere before installing and at intervals thereafter for the latest product documentation. Sphere is refreshed periodically with updated user documentation.



Sphere is for customers with active Axway support contracts. You need a user name and password to log on.

Post comments and questions to the Axway user forum at forums.axway.com/index.php

Accessibility

Axway strives to create accessible products and documentation for users. The following describes the accessibility features of Application Studio and its documentation.

Product accessibility

Application Studio has the following accessibility features.

The Application Studio user interface supports browsers with accessibility features, including keyboard shortcuts. Check the browser user documentation for information.

Documentation accessibility

The product documentation provides the following accessibility features.

- Alternative text is provided for images whenever necessary.
- The PDF documents are tagged to provide a logical reading order.
- The documentation can be used in high-contrast mode.
- There is sufficient contrast between the text and the background color.
- The graphics have the right level of contrast and take into account the way color-blind people perceive colors.

System requirements

1

The following are the system requirements for Application Studio.

If you use multiple Axway products, you can compare this product's system requirements with the supported operating systems, databases, browsers and more for other products. See the *Axway 5 Suite Supported Platforms* guide on Sphere at support.axway.com.

Operating systems

The following are the supported 64-bit operating systems.

- SUSE Linux Enterprise Server 11.2
- Oracle Linux 5.10
- Red Hat Enterprise Linux 6.6

Linux user

You can install and run Application Studio as the root user or non-root user on Linux.

Databases

The following databases are supported.

- MySQL 5.6.12
- Oracle 11g Release 2 (11.2)
- Oracle **RAC**¹ 11g Release 2 (11.2)

RAC note

Application Studio has been tested with Oracle RAC 11g R2 using the Single Client Access Name (SCAN) feature. Oracle JDBC driver version ojdbc6-11.2.0.4 is included in Application Studio to support RAC. Other RAC configurations were not tested.

¹Oracle Real Application Clusters (RAC) provides software for clustering and high availability in Oracle database environments.

Browsers

The following browsers are supported for navigating the user interface.

- Google Chrome 24 and later
- Mozilla Firefox 18 and later
- Microsoft Internet Explorer 9, 10, 11

The following topics include tasks to perform or information to know before installing Application Studio.

Database prerequisites

The following are the prerequisites for using a supported external database. See [Databases on page 8](#) for the list of supported databases.

Using a database requires having a **DBA**¹ or other technically qualified person with knowledge of the database software. This person must perform setup and maintenance tasks, such as database backups.

MySQL

The following are the prerequisites for using a MySQL database.

- The database must be installed and running before installing Application Studio.
- Set the `max_allowed_packet` property to 16 MB.
- You must create a database in the database system before installing Application Studio. The database can have any name. For example, **appstudio**.
- The database user must have full privileges for the database you create.
- If Application Studio and the database are on different computers, Application Studio must have rights to access the remote database. See [Remote access example on page 11](#).

Create database example

The following is an example of creating an empty database in a MySQL session.

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
```

¹database administrator

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> create database appstudio;
```

```
mysql> quit;
```

Remote access example

The following is an example of granting Application Studio access to a remote database.

```
mysql> grant ALL on *.* to 'root'@'aas-ip-address' identified by  
'password';
```

Where:

- `aas-ip-address` is the IP address of the AAS server
- `password` is the MySQL root user password

Oracle

The following script specifies the minimal set of permissions that can be granted while creating the schema or user using Oracle. Grant the user and schema unlimited access to the following permissions: table created, create session, create table, create sequence.

```
DROP TABLESPACE APP_STUDIO_TABLESPACE  
  
INCLUDING CONTENTS AND DATAFILES CASCADE CONSTRAINTS;  
  
CREATE TABLESPACE "APP_STUDIO_TABLESPACE" LOGGING DATAFILE 'C:\DBFILES\APP_  
STUDIO_TABLESPACE.dbf' SIZE 128M REUSE AUTOEXTEND ON NEXT 1M MAXSIZE  
UNLIMITED EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO ;  
  
DROP USER APPSTUDIOUSER cascade;  
  
create user APPSTUDIOUSER identified by APPSTUDIOUSER default tablespace  
APP_STUDIO_TABLESPACE temporary tablespace TEMP;  
  
grant CREATE SESSION to APPSTUDIOUSER;  
  
grant CREATE TABLE to APPSTUDIOUSER;  
  
grant CREATE SEQUENCE to APPSTUDIOUSER;  
  
Alter user APPSTUDIOUSER QUOTA UNLIMITED ON APP_STUDIO_TABLESPACE;
```

Database case-sensitivity rules

You must follow guidelines to make sure the applications you develop in Application Studio are database agnostic. This ensures portability of applications among instances of Application Studio regardless of the database type. For example, an application developed in an instance of Application

Studio with an Oracle database can be exported and imported to an instance of Application Studio with a MySQL database, and vice versa.

Regardless of the database type, apply the following rules when creating a database, tables and column names and in custom queries:

- For database names and table names, mixed case is acceptable, but the names must be case sensitive.
- For database column names, use all upper-case.

The following are the case requirements for database objects by database type and operating system:

For database names and table names

MySQL - case sensitive

Oracle - the Application Studio hibernate layer allows case insensitive

For database column names

MySQL - case insensitive

Oracle - upper-case

License

You do not need a license to install Application Studio; the installer does not prompt for one. After installing and logging on to the user interface, Application Studio is limited to having only three users, including the default admin user. You are allowed to create more than three users, but only the first three alphabetically are allowed to be involved in workflows.

Shortly after installing on your intended production computer, you should request a license key from Axway to fully enable Application Studio. See [Start and log on the first time on page 17](#) for more information.

The following describes the information you are prompted to enter when installing Application Studio. You might want to gather this information before starting the installer.

Note If you already are running Application Studio and want to install an upgrade, skip this topic and see [Upgrade on page 22](#).

Installation directory

You can use or replace the default value for the installation directory.

Database

The installer prompts for values to connect to your external database. It also verifies the database connection before allowing you to proceed. Complete the tasks in [Database prerequisites on page 10](#) before installing.

MySQL

When the database is MySQL:

- Host name is the fully qualified domain name or IP address of the computer running the database Application Studio will use.
- Port is the listening port of the database server.
- User name is the user connecting to the database.
- Password is the password of the database user.
- Database name is name of the database you have created for Application Studio.

Oracle

When the database is Oracle:

- Host name is the fully qualified domain name or IP address of the computer running the database Application Studio will use.
- Port is the listening port of the database server.
- User name is the name of the user and schema.

- Password is the password of the schema user.
- Service name is the alias used when connecting to the database.

Shared secret

The installer prompts for a shared secret. This is used for all users and the database and nodes in a cluster environment. It is for generating a common **salt**¹. It is used to encrypt user passwords and the database password.

You should record and save your shared secret in a safe place, such as in a secure password manager. The secret is shared between the database and one or more instances of Application Studio.

You will need the shared secret to reattach the database to a fresh installation of Application Studio should the original installation become corrupted.

Ports

Default ports for connections to the user interface are 80 for HTTP and 443 for HTTPS. You can use or replace the default values. The installer verifies whether the ports are already in use.

SSL certificate

The installation process generates a self-signed **SSL 2.0**² web server certificate to secure the HTTPS connections from web browser users to Application Studio and its applications. Best practice is to replace this certificate with your own certificate after installing.

You can specify:

- A certificate key size of 2048 (default), 3072 or 4096 bits.
- A validity period of between 1 and 365 days for the certificate. The default is 60 days.

You also are prompted to enter certificate details:

- Two-letter country code
- State or province
- Locality name (for example, your city)
- Organization name (for example, your company)

¹In cryptography, a salt is random data that are used as an additional input to a one-way function that hashes a password or passphrase. The primary function of salts is to defend against dictionary attacks and pre-computed rainbow table attacks. A new salt is randomly generated for each password. In a typical setting, the salt and the password are concatenated and processed with a cryptographic hash function, and the resulting output (but not the original password) is stored with the salt in a database. Hashing allows for later authentication while defending against compromise of the plaintext password in the event that the database is somehow compromised.

²SSL 2.0 is Secure Socket Layer version 2.0.

- Organizational unit name (for example, your department or division)
- Common name (for example, name of computer running Application Studio)

Ready to install

Once your planning is completed, you are ready to install Application Studio.

Extract the compressed file you downloaded from Sphere. To install interactively, run `install.sh` to start the installer in text mode. Follow the prompts to install Application Studio. To install non-interactively in silent mode see [Silent mode installation on page 16](#).

You can install and run Application Studio as the root user or non-root user on Linux.

Silent mode installation

4

Optionally, you can install Application Studio in non-interactive silent mode. Using this requires you first to edit a properties file the installer reads during installation. The file has default values for the same settings you would encounter if installing interactively. You can use or change the default values as you want.

The properties file is in the installer directory and is named `build.properties`.

If you plan to install Application Studio with a non-root user, set the `port.http` and `port.https` properties to values greater than 1024 in `build.properties` unless the Linux administrator allows these ports to be opened for the non-root user. This does not apply for installation with a root user.

Once you have edited the file, start the installer with the following command:

```
install.sh -silent
```

Once the installation has completed successfully, you regain control of the command line. If installation fails, the screen output describes the possible cause. You can do additional failure analysis by reviewing the `install.log` file in the Application Studio installation directory.

Start and log on the first time

5

Use this procedure to start Application Studio and log on the first time after installing.

After logging on, click **Help Center** at the top of the page for information about getting started with Application Studio.

1. Start Application Studio. Run `start-appStudio.sh` in the installation directory.
2. Use a URL in the following format to open the log-on page in a browser:

```
https://<host>:<port>/appstudio
```

Where:

- `host` is the fully qualified domain name or IP address of the computer running Application Studio.
- `port` is the HTTPS port specified during installation. If the HTTPS port is 443, you do not need to specify it in the URL.

If a message displays that a certificate is not trusted, click the appropriate button to proceed to the log-on page.

Note If you instead attempt to connect to the log-on page via the HTTP port specified during installation, you are redirected to HTTPS. If the HTTP port is 80, you do not need to specify it in the URL.

3. On the log-on page, type **admin** as the user name and **axway** as the password. After you submit, you are prompted to set a new password for the admin user. Rules are provided for the new password format.
4. Once logged on, scroll down to the bottom of the page. Notice the message that says **Licensed Users: 3**. You need a license to remove this restriction.

To expand the number of licensed users, you must request from Axway a license that is bound to a specific computer. If you intend to run Application Studio on another computer, you may want to wait before requesting the license. Meantime, you can experiment with Application Studio on other computers with the three-user limit in place.

5. To request a license:
 - a. Click **Axway Application Studio** at the bottom of the page to open the License page.
 - b. Copy the system key on the License page and send it to Axway with a request for a license. Follow the instructions in your Axway welcome letter for requesting the license. This was the email message that informed you how to download the Application Studio software from Axway Sphere.
 - c. Once you have received the license, open the License page and enter the admin user name and password and license and click **Submit**.

After a valid license is entered, the three-user limitation is removed. You might have to refresh the browser page after submitting the license to display the new license summary.

Install a cluster

6

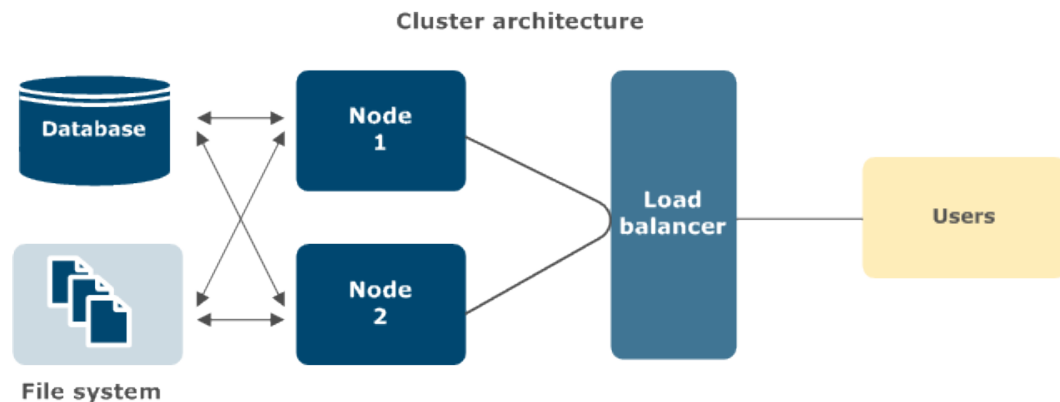
The following topics describe installing Application Studio in an **active-active**¹ cluster environment on multiple computers.

Cluster architecture

A typical architecture of a cluster environment for Application Studio can consist of:

- Two or more application server nodes running Application Studio
- One shared database used by all nodes
- A shared file system (for example, **NAS**² running the **NFS**³ protocol). You can have more than one. For example, one to share the `wflow` directory and another to share the `appstudio-keystore` and `appstudio-truststore` files.
- A load balancer to distribute processing among the application server nodes

In this architecture, if one application server node goes down, users can access Application Studio through other nodes. However, this does not protect from failures in a shared database or shared file system. To further improve availability, you can use a database cluster and **RAID**⁴ in the shared file system.



¹Active-active is a type of high-availability cluster environment. In an active-active cluster, traffic intended for the failed node is forwarded to an existing node or load-balanced across the remaining nodes. This is usually possible only when the nodes have a homogeneous software configuration.

²Network-attached storage (NAS) is file-level computer data storage connected to a computer network providing data access to a heterogeneous group of clients. NAS operates as a file server. It is specialized for this task by its hardware, software or configuration of those elements.

³Network File System (NFS) is a distributed file system protocol originally developed by Sun Microsystems in 1984. It allows a user on a client computer to access files over a network in a manner similar to how local storage is accessed.

⁴Redundant array of independent disks (RAID) is a storage technology that combines multiple disk drive components into a logical unit for the purposes of data redundancy and performance improvement. Data are distributed across the drives in one of several ways, referred to as RAID levels, depending on the specific level of redundancy and performance required.

Why cluster

The following are reasons for operating Application Studio in a cluster environment:

- Scalability. The processing load can be distributed among two or more nodes.
- High availability. If a node goes down, a load balancer can direct user requests to nodes that are still running.

Sticky sessions

Application Studio supports active-active clustering without session synchronization. Nodes do not share sessions. You must use **sticky sessions**¹ to have the load balancer send you to the same node you logged on to originally. If that node fails, the load balancer sends you to the other node, where you have to log on again.

Clustering guidelines

There are various ways to set up clustering and load balancing. Refer to the third-party user documentation for the methods you want to implement. However, the following are points to keep in mind when setting up Application Studio in an active-active cluster.

- Make sure Application Studio servers (all nodes) have full access to the shared database.
- Configure Application Studio servers to use a shared file system. Depending on how the shared file system is set up, you might have to edit the `setenv.sh` script in the installation directory to point to the correct location. The shared file system normally shares the `wflow` directory and the `appstudio-keystore` and `appstudio-truststore` files found in each node's installation directory.
- Make sure all Application Studio servers have read-write access to the shared file system. Edit `setenv.sh` and configure `Dsecure.config.file` variable to the absolute path of the shared secret key as follows:

```
-Dsecure.config.file=<user-home dir>/appStudio/.appstudio/
```

For example: `-Dsecure.config.file=/home/axway/.appStudio/.appstudio/`; where `axway` is the user installed Application Studio.
- Do not use URLs in the format `https://<node IP_address>/appstudio/<other content>`. Such URLs cannot be relied upon to redirect to the load balancer.
- Edit the `server.xml` file at `<install directory>\apache-tomcat-<version>\conf` to point to the load balancer or proxy or firewall so traffic is routed correctly.

¹A sticky session is a feature of many commercial load balancing solutions for web-farms to route requests for a session to the same physical machine that serviced the first request for that session. This is mainly used to ensure an in-proc session is not lost as a result of requests for a session being routed to different servers. Since requests for a user are always routed to the same machine that first served the request for that session, sticky sessions can cause uneven load distribution across servers.

- Edit the `server.xml` file to have identical values on all cluster nodes for the following properties:
 - `keystoreFile` is the absolute path of the keystore
 - `keystorePass` is the keystore password
- Edit the `repair.sh` and `recover.sh` scripts in `<install directory>/cli-tools/` to replace the string:

```
-DappStudioData=$AppStudioDataDir
```

With:

```
-DappStudioData=<user-home dir>/appStudio
```

Where `<user-home dir>` is the path of the user's home directory in Linux.

Licenses for cluster nodes

You must request and apply a license for each Application Studio node before joining the nodes together in a cluster. Moreover, before replacing or updating the license for a node, you must remove the node from the cluster and rejoin the node only after the license has been replaced or updated. See the steps for requesting and applying a license for a single Application Studio in [Start and log on the first time on page 17](#).

If nodes already are clustered, which means the `server.xml` files have been updated to the cluster settings, you must restore the `server.xml` files to the default, non-clustered state and then license the nodes. A good approach is to save the default `server.xml` as `server.xml.original` on each node. However, if the default `server.xml` is not available, save a copy of the current `server.xml` as `server.xml.cluster` and remove the cluster settings from `server.xml`. Once restored to the default state, you can use it in the licensing procedure and save the file as `server.xml.original`. Then you can resume using `server.xml.cluster`.

Stop Application Studio before working on a `server.xml` file. The file is at `<install directory>/apache-tomcat-<version>/conf`.

Other resources

Application Studio is built on Joget Workflow Enterprise Edition, a platform for building enterprise web applications.

Joget does not have a special mechanism for clustering. It uses clustering support provided by the Apache Tomcat application server. This is described at:

<http://tomcat.apache.org/tomcat-7.0-doc/cluster-howto.html>

Use this procedure to upgrade Application Studio from version 1.2 to version 1.4. If you have never installed Application Studio or want to install a new instance of the product, skip this topic and see [Plan and install on page 13](#).

When upgrading a single instance or a cluster, if there are custom attributes or any other customizations in the `server.xml` file, you must add them to the file manually in the upgraded Application Studio. After upgrading you can find the old `server.xml` file in the backup directory specified during the upgrade.

Although Application Studio supports multiple types of external databases, the upgrading process only enables installing a new product version that uses the same database as the previous version. You cannot change databases or database types. For example, you cannot change from an existing MySQL database to a new MySQL database and preserve data from the old database. You also cannot change from MySQL to Oracle. Changing the database requires performing a complete installation with a new database.

Note If you have a version earlier than 1.2, you must upgrade to 1.2 before upgrading to 1.4. See upgrading in the 1.2 installation guide. You can get the guide on the Application Studio Help Center at <http://app-studio-help-center.squarespace.com/installation/>.

Restore auto-start after upgrade

In version 1.2, the script `/etc/init.d/appstudio` does not work; it is supposed to start Application Studio automatically when a Linux computer starts. You must start 1.2 manually by running `start-appStudio.sh` in the installation directory. After upgrading to 1.4, however, there is a new `appstudio` script in `/extras/scripts`. Copy the file to `/etc/init.d`, overwriting the existing file. This restores the auto-start feature.

Prerequisites

- Verify the existing installation of Application Studio by locating the installation directory. There also should be a security data directory. The security data directory, if installed as root, is at `/root/.appStudio`.
- Make sure you have the upgrade package. This is a tar.gz file you must unpack.
- Review the release notes for the upgrade to determine whether any issues might affect installation or operations.
- Best practice is installing the upgrade as root. However, if 1.2 was installed under non-root user, you can upgrade as non-root.
- Stop the server if running.

- Back up the database and any database configurations.
- Back up any custom attributes in `server.xml`.
- Back up trust store and certificate.

What changes after upgrading

Upgrading results in the following changes.

- The following objects are upgraded:
 - The version of Joget that Application Studio runs on
 - **Tomcat**¹
 - **JDK**²
 - **Ant**³
 - The Application Studio **WAR**⁴ file
- The upgraded installation points to the existing database
- No changes to truststore and certificate

Upgrade steps

1. Stop Application Studio. See [Stop Application Studio on page 25](#).
2. Back up the database and the existing installation of Application Studio, including the security data directory. In the event of problems, you can revert to the previous version of Application Studio only if you perform backups before upgrading. See [Uninstall on page 26](#).
3. Run `upgrade.sh` to start the installer in text mode. Follow the prompts to continue.
4. Specify the path for the currently installed version to proceed with the upgrade.
5. Read the warning about stopping and backing up Application Studio and its data. Determine whether to proceed with the upgrade or cancel. If you want to proceed, enter when prompted a path for backing up data files. This is in case you want to revert to the previous version after upgrading.
6. After upgrading, start Application Studio and log on to the user interface. You can confirm you are running the new version by scrolling to the bottom of the page and reading the Application Studio version number.
7. Copy the `/extras/scripts/appstudio` file to `/etc/init.d`, overwriting the existing file. See [Restore auto-start after upgrade on page 22](#).

¹Apache Tomcat, often referred to as Tomcat, is an open-source web server and servlet container developed by the Apache Software Foundation (ASF).

²The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

³Apache Ant is a software tool for automating software build processes. Ant stands for "another neat tool."

⁴A Web application ARchive (WAR) is a JAR file used to distribute a collection of JavaServer Pages, Java servlets, Java classes, XML files, tag libraries, static web pages (HTML and related files) and other resources that together constitute a web application.

If the upgrade fails, you can restore backed-up data and revert to the previous version of Application Studio.

Upgrade nodes in cluster

If you are running Application Studio in a cluster of multiple nodes, upgrade each node using the procedure in [Upgrade on page 22](#). Then edit the `setenv` file in the installation directory to set the `AppStudioWflow` property to the location of the shared workflow folder. For example, if `wflow` is mounted to the path `/opt/share/wflow`, use `export AppStudioWflow="/opt/share/wflow"`.

You also must edit `setenv.sh` on each of the cluster node and configure `Dsecure.config.file` variable to the absolute path of the shared secret key as follows:

```
-Dsecure.config.file=<user-home dir>/.appStudio/.appstudio/
```

For example:

```
-Dsecure.config.file=/home/axway/.appStudio/.appstudio/;
```

Where `axway` is the user-installed Application Studio.

Also edit the `repair.sh` and `recover.sh` scripts in `<install directory>/cli-tools/` to replace the string:

```
-DappStudioData=$AppStudioDataDir
```

With:

```
-DappStudioData=<user-home dir>/.appStudio
```

Where `<user-home dir>` is the path of the user's home directory in Linux.

Stop Application Studio

8

To stop Application Studio, run `stop-appStudio.sh` in the installation directory.

Before uninstalling, you might want to save sensitive files or directories to reuse in a future installation of Application Studio. You copy the saved files to the new instance, overwriting existing files if necessary.

To re-use application configuration

Save any changes made to files in `<install directory>/apache-tomcat-<version>/conf`.

To re-use apps and plugins

Save the directory `<install directory>/wflow`.

To re-use a database after uninstalling

Save the sensitive files in `~/.appStudio`.

Before uninstalling, stop Application Studio.

To stop Application Studio, run `stop-appStudio.sh` in the installation directory.

Next, review the sensitive data files in `~/.appStudio` and determine whether to save or discard them.

After reviewing the sensitive data files, run `uninstall.sh` in the installation directory to start the uninstall process. Follow the prompts. The user performing the uninstall must have permissions to do so.

Glossary

This is a glossary of terms for Axway Application Studio.

active-active

Active-active is a type of high-availability cluster environment. In an active-active cluster, traffic intended for the failed node is forwarded to an existing node or load-balanced across the remaining nodes. This is usually possible only when the nodes have a homogeneous software configuration.

Ant

Apache Ant is a software tool for automating software build processes. Ant stands for "another neat tool."

BPM

Business process management (BPM) is the discipline of managing processes for improving business performance outcomes and operational agility. Processes span organizational boundaries, linking people, information flows, systems and other assets to create and deliver value to customers and constituents.

CA

A certificate authority or certification authority (CA) is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others to rely upon signatures or assertions made by the private key that corresponds to the public key that is certified. A CA is a third party trusted by the subject (owner) of the certificate and the party relying upon the certificate.

DBA

database administrator

JDK

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

MAC address

A media access control address (MAC address) is a unique identifier assigned to network interfaces for communications on the physical network segment. MAC addresses are used as a network address for most IEEE 802 network technologies, including Ethernet.

NAS

Network-attached storage (NAS) is file-level computer data storage connected to a computer network providing data access to a heterogeneous group of clients. NAS operates as a file server. It is specialized for this task by its hardware, software or configuration of those elements.

NFS

Network File System (NFS) is a distributed file system protocol originally developed by Sun Microsystems in 1984. It allows a user on a client computer to access files over a network in a manner similar to how local storage is accessed.

RAC

Oracle Real Application Clusters (RAC) provides software for clustering and high availability in Oracle database environments.

RAID

Redundant array of independent disks (RAID) is a storage technology that combines multiple disk drive components into a logical unit for the purposes of data redundancy and performance improvement. Data are distributed across the drives in one of several ways, referred to as RAID levels, depending on the specific level of redundancy and performance required.

salt

In cryptography, a salt is random data that are used as an additional input to a one-way function that hashes a password or passphrase. The primary function of salts is to defend against dictionary attacks and pre-computed rainbow table attacks. A new salt is randomly generated for each password. In a typical setting, the salt and the password are concatenated and processed with a cryptographic hash function, and the resulting output (but not the original password) is stored with the salt in a database. Hashing allows for later authentication while defending against compromise of the plaintext password in the event that the database is somehow compromised.

SDK

A software development kit (SDK or "devkit") is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

self-signed certificates

In cryptography and computer security, a self-signed certificate is an identity certificate that is signed by the same entity whose identity it certifies. This term has nothing to do with the identity of the person or organization that actually performed the signing procedure. In technical terms a self-signed certificate is one signed with its own private key.

SHA-1

In cryptography, SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function designed by the United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST.

SSL

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols for providing communication security over the Internet.

SSL 2.0

SSL 2.0 is Secure Socket Layer version 2.0.

sticky sessions

A sticky session is a feature of many commercial load balancing solutions for web-farms to route requests for a session to the same physical machine that serviced the first request for that session. This is mainly used to ensure an in-proc session is not lost as a result of requests for a session being routed to different servers. Since requests for a user are always routed to the same machine that first served the request for that session, sticky sessions can cause uneven load distribution across servers.

Tomcat

Apache Tomcat, often referred to as Tomcat, is an open-source web server and servlet container developed by the Apache Software Foundation (ASF).

WAR

A Web application ARchive (WAR) is a JAR file used to distribute a collection of JavaServer Pages, Java servlets, Java classes, XML files, tag libraries, static web pages (HTML and related files) and other resources that together constitute a web application.

X.509

In cryptography, X.509 is an ITU-T standard for a public key infrastructure (PKI) and Privilege Management Infrastructure (PMI). X.509 specifies standard formats for public key certificates, certificate revocation lists, attribute certificates and a certification path validation algorithm.