



QAD Adaptive Applications  
QAD Adaptive

Installation Guide  
**QAD Adaptive On-Premise**

70-3392-Adaptive

QAD Adaptive Applications

QAD Adaptive

November 2025

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

This document contains trademarks owned by QAD Inc. and other companies.

Copyright © 2025 by QAD Inc.

**QAD Inc.**

100 Innovation Place  
Santa Barbara, California 93108  
Phone (805) 566-6000  
<http://www.qad.com>

# Table of Contents

QAD Adaptive On-Premise Installation Guide .....	4
Assumptions .....	5
Prerequisites .....	6
Preparation Steps .....	7
Installation Steps - Introduction .....	11
Installation Steps - Direct Installation .....	12
Installation Steps - Prepared Context .....	18
Post-Installation Steps .....	24
QAD Reporting Framework on Linux .....	26
Initial Environment Validation .....	28
QAD Adaptive Installation Considerations .....	29
Customizing the QAD Adaptive Installation .....	30
QAD Adaptive Tuning and Configuration Options .....	32
Installing New Versions of QAD Products .....	33
Installing QAD Production Orders .....	34
Installing QAD Enterprise Asset Management .....	35
Installing Updates .....	36
Troubleshooting .....	37

# QAD Adaptive On-Premise Installation Guide

This guide describes the steps required to install the QAD Adaptive release into an existing QAD Enterprise Edition environment or to upgrade an existing installation of QAD Adaptive.

# Assumptions

1. This guide assumes that all necessary software and prerequisites have been made available on the server being used for the installation.
2. The user performing the installations should be familiar with basic Linux administration, OpenEdge environment administration, and YAB.
3. QAD Adaptive is only supported on the QAD Reference Architecture. This consists of:
  - a. Linux Operating System
  - b. UTF-8 based databases

# Prerequisites

Additional details are provided in subsequent sections, where appropriate, but the following prerequisites are necessary to complete a successful QAD Adaptive installation:

## Technical Prerequisites

1. RedHat 8
2. Progress OpenEdge, minimum version OE12.8.9.
3. If upgrading from OpenEdge 11 to OpenEdge 12, both versions are required. The OE11 version can be removed after a successful upgrade.
4. Java (note: only OpenJDK versions are supported i.e. Oracle Java is no longer supported)
  - a. Java 17 - A Java 17 install (17.0.7) is required and will be used as the default Java version (`java.default.dir`) for the application
  - b. Java 11 - A Java 11 install (11.0.5) is required and will be used as the Java version (`java.v11.dir`) for the Logi Composer component
  - c. Java 21 - A Java 21 install (21.0.7) is required and will be used as the Java version (`java.v21.dir`) for the NiFi 2.4 component
5. Python 3.12.1
6. Linux packages: `podman`, `podman-compose`

```
> sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
> sudo dnf install -y podman podman-compose
```

7. The host system or network firewall allows outbound HTTPS (TCP port 443) network access to the following:
  - a. Docker Registry: [docker.io](https://docker.io) (required for NGINX, Cassandra, Redis and Kafka Mirror Maker containers)
8. Zip64 support:  
The `channel-islands-apps-image` zip file is a zip64 formatted archive. The zip program installed must support this format.
9. Disk space:
  - a. The installation of the `channel-islands-apps-image` zip file requires the addition of approximately 20 GB of files into the environment. It is recommended that you have at least 25 GB space available in the target filesystem before performing the installation.
  - b. Some of this disk space can be reclaimed after the upgrade is completed by the removal of obsolete packages. See the Temporary Files section of the *QAD Configuration and Administration Guide YAB 1.19* for details.
  - c. This estimate does not include other QAD products also being installed. When installing additional products, it is recommended an additional 5GB of space is made available.
  - d. In YAB 1.15 and later, the use of temp space during the package installation has been greatly reduced from previous releases and, as such, is less of a concern. However, it is recommend to have approximately 5GB of temp space available for the installation.
  - e. The location used by YAB for temp space can be controlled by setting the `YAB_JAVA_OPTS` option. See the Clean command section of the *QAD Configuration and Administration Guide YAB 1.19* for details.

## QAD Software Prerequisites

1. QAD Enterprise Edition 2022.
  - a. Including a minimum of Operations Patch 6
2. QAD Financials:
  - a. QAD Financials 2025 is now distributed with QAD Adaptive and will be installed automatically
3. QAD Enterprise Asset Management 2017.1 is required for systems that have EAM installed.
4. QAD .NET UI, cumulative patch 14 (`netui-cumulative-3.8.16.3`) must be installed
5. QAD QXtend 2.0 is required.
6. If implementing a secure configuration, as described in the *QAD Security Administration Guide*:
  - a. QAD Alerts 1.4.0 is required for systems using Alerts.
7. QAD Production Orders 3.2.39 is required for systems that have Production Orders installed:
8. QAD BPM is not compatible with OE12.
9. QAD TAM 3.7 is delivered as part of the QAD Adaptive release. If an existing version of TAM is installed, it will be automatically upgraded as part of the installation.
10. If QAD Production Execution (PE) is in use, version 4.2 or higher (certification pending) is required.



This document describes the process for the installation of these prerequisites as part of a QAD Adaptive installation.

These prerequisites may also be installed before beginning the QAD Adaptive installation. If this approach is taken, refer to the relevant *Installation Guide* for these products.

# Preparation Steps



This section covers the most frequent preparation and prerequisite steps. However, it is recommended that you read the entire document before beginning the installation.

Future sections may provide details that will influence how you prepare for and proceed with your installation.

## Validate Technical Prerequisites

### Operating System Account

The QAD Adaptive installation must be executed by the same OS user account used to perform the QAD Enterprise Edition installation.

Due to restrictions in some of the technologies used and for security reasons, it is strongly recommended that this account not be the root account. Executing the installation as the root user will result in errors.

### Validate the Operating System Version

QAD Adaptive is only supported on RHEL. Before proceeding, validate that you are running on a supported version of RHEL.

#### How to check your version if the server is running RHEL

```
mfg@gvaranald01:~$ cat /etc/redhat-release
Red Hat Enterprise Linux release 8.x (Ootpa)
```

Refer to your Operating System Provider documentation for additional details.

### Ensure the Appropriate Java Versions are Installed

Validate the Java version installed and compare it to the minimum version, as outlined in the [Prerequisites](#) section. If the Java version is not correct, install an appropriate version.

Refer to your OS provider on installing Java JDK.

### Validate the Python Version

Ensure that the Python is version 3.12 or higher.

To validate the Python version installed, enter the following command:

```
> python --version
```

### Validate OpenSSL Libraries

For enhanced security, QAD Adaptive generates an access token during login. The token is signed using the OpenSSL library to verify its authenticity.



The QAD Adaptive installer validates your system's OpenSSL library during installation. If you do not have the appropriate OpenSSL library, the installation will fail.

When using RHEL8, the minimum version required is 1.1.1.n.

Refer to the *Installation Guide* of your QAD Enterprise Edition release for details on how to configure the OpenSSL libraries.

### Review the Available Semaphores

Installing QAD Adaptive introduces a number of new architectural components that require a significant number of free semaphores on the installation server. Review the available semaphores on your server and ensure that they are sufficient before continuing.

## Install ZeroMQ 4

Installing QAD Adaptive requires ZeroMQ 4 to be available as prerequisite. If it is not installed, the QAD Adaptive installation may fail. The following examples use yum to check for and install ZeroMQ. Refer to your operating system provider documentation for specific guidance for your operating system.

To verify if ZeroMQ is already installed, run:

```
> sudo yum list installed | grep zeromq
```

If ZeroMQ is installed, the output is similar to:

```
zeromq.x86_64 4.1.4-7.el7 @salt-latest
```

**Note:** The version referenced may vary, but should be 4.x.

If ZeroMQ is not installed, the output is blank. To install ZeroMQ, run:

```
> sudo yum install zeromq
```

## Logi Licenses

An OEM software license for supported Logi Analytics products (Logi Composer) will be installed automatically during installation and no preemptive action is required on your part.

The same license is installed in both production and non-production environments.

## Logi Upgrade

The QAD Adaptive release no longer includes built-in support for Logi Platform Services (LPS) migration processes. If you need to migrate LPS artifacts from the source environment, you must first install AUX2024 to perform this migration.

## Validate Existing QAD Instance Configurations

The QAD Adaptive installation requires that you run a `yab update` command. To ensure that this is successful, a number of pre-checks are recommended.

### Validate the User Configuration in YAB

Run the following commands to determine the YAB configured username and password:

```
> yab config user
> yab config password
```

Log in to the existing QAD Enterprise Edition installation using the .NET UI or CHUI client to confirm that the username and password configuration is valid.



QAD Adaptive includes Web UI Maintenance Mode, which prevents users from logging into the Web UI while the installation is running. To ensure that YAB can complete the installation, the user configured must be assigned either the `qadadmin` or `SuperUser` roles.

## Ensure that Existing Patches or Customizations Compile

Execute the appropriate YAB commands to ensure that all existing patches and customizations compile successfully.

For example, execute:

```
> yab code-mfg-customizations-update
> yab code-mfg-update
```

## Export Any Extension Apps in Development

If you are upgrading from an existing QAD Adaptive environment and have extension apps under active development, you must export these before continuing.



If you do not export these apps before proceeding with the update, then any changes that have not been exported may be lost.

Your extension apps must also be OE12 compatible before upgrading.

Refer to the *QAD Enterprise Platform Developer Guide* for additional details.

## Update to YAB 1.19

If not already installed, download and install the YAB 1.19 release.

This installation is performed by running the command:

```
> yab install <path to media download>/yab-ee-image-1.19.0.XXX.zip
```

Refer to the documentation with the YAB 1.19 release for detailed instructions and any upgrade considerations.

If you have installed the YAB client script outside of your environment, the client will not automatically update and must be updated before proceeding. Refer to the YAB 1.19 release notes for details on this step.



If you have installed the YAB client outside of the QAD Enterprise Edition environment, you must update the client before proceeding.

## Review the Appserver Startup Parameters

If you have not previously installed Adaptive UX 2019 or later, the installation of QAD Adaptive includes an update to define database connections in a pf file, rather than the previous `bootstrap.xml` file.

When databases are defined in a parameter file, they are processed as startup parameters and a shared memory connection only requires read permission on the database files. This removes the previous requirement to open file permissions on the database files.

When the value is `true`, all Progress sessions that use bootstrapping MUST reference the bootstrap's parameter file. When the value is `false`, all Progress sessions that use bootstrapping MUST NOT reference the bootstrap's parameter file.

The factory default settings for standard Enterprise Edition sessions respect this option, but if these values have been overridden (that is, `svrstartupparam`), they should be reviewed for compatibility. Review any custom sessions as well.

To use database connections defined in `bootstrap.xml`, edit `install.conf` before the installation and remove the line:

```
bootstrap._useparameterfile=true
```

Or set this value to `false` in `configuration.properties`, after executing the `yab install` command, but before running `yab update`.

## Update Overridden Settings in the Environment

Settings that are overridden in an environment will not be updated by settings in the `install.conf` file during an installation, unless the installation prompts for the new value or the setting is explicitly set up to replace an overridden value.

A setting is considered overridden in the environment if it is defined in a system configuration file (`build/config/system`) or in `configuration.properties`. These settings can be reviewed using the `-overrides` option in `yab config`.

This behaviour can be overridden using the installer option `install-conf-override`, which can be configured as a request argument (`-install-conf-override`) or in `install.conf`. When enabled, all configuration settings in the `install.conf` file are merged into the environment, even if the settings are already configured in the environment.

## Installation Steps - Introduction

There are two approaches available to install QAD Adaptive. The selected approach depends on the environment into which QAD Adaptive is being installed.

Using a Prepared Context installs QAD Adaptive into an isolated context so the installation can be prepared, without impacting the running environment. This allows changes to be reviewed and prepared, without disrupting users and increases confidence that updating the environment will be successful.

This approach can be of benefit when installing into a live environment with existing users as it allows you to prepare parts of the update before requiring downtime. For more information on Prepared Context, see the *QAD Configuration and Administration Guide YAB 1.19*.

A direct installation immediately impacts the running environment and is recommended if running an installation into an isolated environment that is not currently in use.



Certain types of changes are not supported in a Prepared Context. If you are performing an OpenEdge upgrade as part of the installation, then you must use the direct installation technique.

### Upgrading an Existing QAD Adaptive Installation

If you have previously installed Channel Islands or Adaptive UX and are upgrading to the QAD Adaptive release, the same basic installation process should be followed. Any steps that were performed as part of the previous installation can be skipped.

# Installation Steps - Direct Installation

## Prerequisite Installation

### Operations Patch

If an Operations patch is required and not installed, install it now:

```
> yab -install-update:false install <mfg-patch>.zip
```

### QAD .NET UI Cumulative patch

If a .NET UI cumulative patch is required and not installed, install it now:

```
> yab -install-update:false install netui-cumulative-3.8.x.x.zip
```

### QAD Financials

QAD Financials 2025 will automatically be installed with the QAD Adaptive installation.

### Upgrade from OE11 to OE12

To check the current OpenEdge version:

```
> yab config openedge.dir
```

If the current version is OE11 stop the environment and take a database-backup

```
> yab stop  
> yab database-backup -tag:oe11
```

Add the following configuration:

```
# Preallocate configuration can be removed after a successful upgrade  
db._base.bipreallocate=false  
  
# @exclude  
ws.default=
```

## QAD Adaptive Installation

Download the QAD Adaptive image zip file, `channel-islands-apps-image`, into a location available on the server.

Execute the QAD Adaptive installation by running a `yab install` command against the downloaded image. If you are upgrading from a previous Channel Islands or Adaptive UX release, each prompt will be populated from your previous installation. Review and adjust the values, as needed.

```
> yab -install-update:false install channel-islands-apps-image-4.12.x.x.zip
```

The QAD Adaptive installer will now execute and guide you through the installation of QAD Adaptive.

Press Enter to confirm that you want to begin the installation process:

```
This installer will guide you through the process of installing QAD Adaptive.  
  
The installation is being recorded in the file:  
/tmp/install.mfg.log  
  
At the prompt the ENTER key submits a response and moves to the next question.  
'!' will enter command mode to navigate through the questions.  
  
Hit ENTER to continue  
>
```

Enter the location of the Progress OpenEdge installation.

The current value from the environment will be populated, but can be updated, if required:

```
[Progress OpenEdge]  
  
Enter the location of Progress OpenEdge:  
> /tech/progress/dlc1289
```

Enter the location of the Java 11 installation.

The current value from the environment will be populated, but can be updated, if required:

```
[Java v11]  
  
Enter the location of Java version 11 (minimum 11.0.5)  
> /tech/java/openjdk11.0.5
```

Enter the location of the default Java 17 installation.

The current value from the environment will be populated, but can be updated, if required:

```
[Java default (v17)]  
  
Enter the location of default Java (minimum 17.0.7)  
> /tech/java/openjdk17.0.7.7
```

Enter the location of the Java 21 installation.

This value will only be used for the NiFi 2.4 installation component.

```
[Java v21]  
  
Enter the location of Java version 21 (minimum 21.0.7)  
> /tech/java/openjdk21.0.7_6
```

Enter the location of the Python installation.

The current value from the environment will be populated, but can be updated, if required.

```
[Python]
```

```
Enter the location of the Python interpreter:  
> /tech/python/3.12.1/bin/python3
```

Enter the QAD end user number for the customer associated with this environment:

```
[Customer End User Number]
```

```
Enter the QAD End User Number for the customer associated with this environment:
```

```
Ex. 00012345  
> 12345678
```

Enter the vertical market for the customer associated with this environment. This information is used to provide vertical-specific help content.

```
[Customer Vertical]
```

```
Enter the vertical market for the customer associated with this environment:
```

```
Used to provide help content specific to your vertical.
```

```
Valid values: - auto (Automotive) - food (Food & Beverage) - comp (Consumer Products) - elec  
(Electric/High Tech) - ind (Industrial) - lsm (Life Science Medical) - lsp (Life Science Pharma)  
> auto
```

Enter the type of environment being installed. The type of environment being installed is used to configure the appropriate functionality.

```
[Environment Type]
```

```
Enter the type of environment being installed:
```

```
Describes the role of the environment and is used to configure appropriate functionality.
```

```
Valid values: - development - test - production  
>
```

Enter the Namespace of the environment into which you are installing. The Namespace controls the set of apps that can be maintained in the environment.

When installing into multiple environments (for example: Dev, Test, and Production), the same Namespace should be used.

```
[Environment Namespace]
```

```
Enter the namespace to associate with this environment:
```

```
The namespace identifies the organization using "Reverse Domain Notation" and is used to control  
which Apps can be maintained using this environment.
```

```
Ex. com.[ORGANIZATION]  
>
```

Enter the default app to be created. This is the app that is used as the default by the maintenance programs for app-related system data.

```
[Default App]


Enter the default app:


Configures the app that will be used by default when platform extensions are developed.

Ex. [ORGANIZATION]App
>
```

Select whether to enable integration with QAD Enterprise Asset Management requisitions. If enabled, QAD EAM-based requisitions will be used.

This integration is typically disabled if QAD EAM is installed, but not in use or configured.

 This question is only provided when QAD EAM is installed.

 To enable this integration, QAD EAM 2017.1 must be installed. If you are using an earlier version of QAD EAM and want to use this functionality, you must upgrade QAD EAM before proceeding.

```
[EAM Requisitions Integration]

Enable EAM Requisitions integration?

[ ] y - Yes
[X] n - No

> n
```

Select whether to install QAD Reporting Framework on Linux. If you select to install this feature, additional prerequisites are required and the location of Wine must be entered (which will configure the following YAB setting: 'qrf.default.wine.dir')

Refer to the section [QAD Reporting Framework on Linux](#) for additional details.

```
[QAD Reporting Framework on Linux]

Install QAD Reporting Framework on Linux?

[ ] y - Yes
[ ] n - No

>
```

Select whether to install QAD Sales Configurator. This option must be selected to install the QAD Adaptive-based Sales Configurator app.

```
[QAD Sales Configurator]

Install QAD Sales Configurator?

[ ] y - Yes
[ ] n - No

>
```

Select whether to proceed with the installation or not.

If you continue, the QAD Adaptive packages will be added into the environment and configured.

If you do not continue at this point, no changes are made to the environment.

```
[Start Installation]

Start installing QAD Adaptive ERP 2024 (Web UI)?

n - No
y - Yes

>
```

## Review the Current Configuration Using yab config to Determine If the Configuration Needs to be Adjusted

Apply any configuration changes in the `configuration.properties` file.

Some common configuration options are discussed in the section "QAD Adaptive tuning and configuration options."

## Review the appserver Configuration Using yab config to Determine If the Configuration Needs to be Adjusted

Review the appserver configuration and ensure that it is configured correctly for your database connections. Refer to the *Review Appserver startup parameters* section of [Preparation Steps](#) for background.

If you have accepted the default:

```
bootstrap._useparameterfile=true
```

Review the appserver startup parameters defined in your environment:

```
> yab config pas.*.openedge.session-mgr.agentStartupParam
```

If any of these definitions do not include the relevant pf file, review the location of the definition using `-trace` and update it to include the bootstrap connection.

For example, run:

```
> yab config pas.mfg.openedge.session-mgr.agentStartupParam -trace
appserver.mfg.srvrstartuppam=-T /tmp -d mdy -yy 1950 -s 32768 -mmax 10000 -inp 32000 -
rereadnolock -c 30 -D 1000 -Bt 350 -nb 200 -h 25 -tok 4096 -tmpbsize 8 -TB 31 -TM 32 -cpinternal
utf-8 -cpstream utf-8 -cpcoll ICU-UCA -cpcase basic -mmax 20408 -Bt 20000 -tmpbsize 4

[1: build/config/system/environment.properties]

appserver.mfg.srvrstartuppam=${appserver._base.srvrstartuppam} -mmax 20408 -Bt 20000 -
tmpbsize 4

[2: build/config/packages/yab-ee-foundation/1/20/0/9/default/yab-ee-foundation.properties]

# @if bootstrap._useparameterfile=true
pas.mfg.openedge.session-mgr.agentStartupParam=-T "${base}/temp" -pf ${bootstrap.default.
parameterfile} -mmax 16384 -Bt 10000 -tmpbsize 4 -lkwmo 1800
```

Edit `build/config/system/environment.properties` and update the entry to include `${bootstrap.default.parameterfile}`

```
pas.mfg.openedge.session-mgr.agentStartupParam=-pf ${bootstrap.default.parameterfile} -mmax 20408
-Bt 20000 -tmpbsize 4
```

Review the updated value using `yab config` and ensure that it now includes the bootstrap pf file:

```
> yab config pas.mfg.openedge.session-mgr.agentStartupParam
```

Repeat this for other appserver definitions.

### Optional: Execute a Financials backwards compatible upgrade (only required if upgrading from Adaptive UX 2022 or earlier Adaptive UX releases)

```
> yab conversion-ee-upgrade-reset  
> yab -conversion._type:fin conversion-ee-upgrade
```

In an OE11 to OE12 upgrade

```
> yab conversion-ee-upgrade-reset  
> yab -conversion._type:fin conversion-ee-upgrade -openedge.dir:<OE11 location> -validation.error:  
false
```

## Apply the Update



Before applying the update to the environment, refer to the installation documentation for any additional updates for any pre-update steps.

```
> yab database-truncate -openedge.dir:<OE11 location> -validation.error:false (needed only in an  
OE11 to OE12 upgrade)  
> yab update
```

The update installs QAD Adaptive, as well as any additional updates included in the configuration.

The status of the update can be monitored in `build/logs/yab.log`.

## Apply Post-Update Steps

Once the update completes, refer to the installation documentation for any other products being installed for any additional steps needed.

# Installation Steps - Prepared Context



Certain types of changes are not supported in a Prepared Context. If you are performing an OpenEdge upgrade as part of the installation, you must use the direct installation technique.

A prepared context is created and referenced by providing the `-c` parameter to each YAB command. The following examples use the context name `ciapps` for the installation.

Each command must use the same `-c` parameter to link them together into a single prepared change.

## Prerequisite Installation

### Operations Patch

If an Operations patch is required and not installed, install it now:

```
> yab -c:ciapps -install-update:false install <mfg-patch>.zip
```

### QAD .NET UI Cumulative Patch

If a .NET UI cumulative patch is required and not installed, install it now:

```
> yab -c:ciapps -install-update:false install netui-cumulative-3.8.x.x.zip
```

### QAD Financials

QAD Financials 2025 will automatically be installed with the QAD Adaptive installation.

## QAD Adaptive Installation

Download the QAD Adaptive image zip file, `channel-islands-apps-image`, into a location available on the server.

Execute the QAD Adaptive Installation by running a `yab install` command against the downloaded image.

```
> yab -c:ciapps -install-update:false install <path to image zip above>
```

The QAD Adaptive installer will now execute and guide you through the installation of QAD Adaptive. If you are upgrading from a previous Channel Islands or Adaptive UX release, each prompt will be populated from your previous installation. Review and adjust the values, as needed.

Press Enter to confirm that you want to begin the installation process:

```
This installer will guide you through the process of installing QAD Adaptive.

The installation is being recorded in the file:
/tmp/install.mfg.log

At the prompt the ENTER key submits a response and moves to the next question.
'!' will enter command mode to navigate through the questions.

Hit ENTER to continue
>
```

Enter the location of the Progress OpenEdge installation.

The current value from the environment will be populated, but can be updated, if required.

```
[Progress OpenEdge]
```

```
Enter the location of Progress OpenEdge:  
> /tech/progress/dlc1289
```

Enter the location of the Java 11 installation.

The current value from the environment will be populated, but can be updated, if required:

```
[Java v11]
```

```
Enter the location of Java version 11 (minimum 11.0.5)  
> /tech/java/openjdk11.0.5
```

Enter the location of the default Java 17 installation.

The current value from the environment will be populated, but can be updated, if required:

```
[Java default (v17)]
```

```
Enter the location of default Java (minimum 17.0.7)  
> /tech/java/openjdk17.0.7.7
```

Enter the location of the Java 21 installation.

This value will only be used for the NiFi 2.4 installation component.

```
[Java v21]
```

```
Enter the location of Java version 21 (minimum 21.0.7)  
> /tech/java/openjdk21.0.7_6
```

Enter the location of the Python installation.

The current value from the environment will be populated, but can be updated, if required.

```
[Python]
```

```
Enter the location of the Python interpreter:  
> /tech/python/3.12.1/bin/python3
```

Enter the QAD end user number for the customer associated with this environment:

```
[Customer End User Number]
```

```
Enter the QAD End User Number for the customer associated with this environment:
```

```
Ex. 00012345  
> 12345678
```

Enter the vertical market for the customer associated with this environment. This information is used to provide vertical-specific help content.

```
[Customer Vertical]

Enter the vertical market for the customer associated with this environment:

Used to provide help content specific to your vertical.

Valid values: - auto (Automotive) - food (Food & Beverage) - comp (Consumer Products) - elec
(Electric/High Tech) - ind (Industrial) - lsm (Life Science Medical) - lsp (Life Science Pharma)
> auto
```

Enter the type of environment being installed. The type of environment being installed is used to configure the appropriate functionality.

```
[Environment Type]

Enter the type of environment being installed:

Describes the role of the environment and is used to configure appropriate functionality.

Valid values: - development - test - production
>
```

Enter the Namespace of the environment into which you are installing. The Namespace controls the set of apps that can be maintained in the environment.

When installing into multiple environments (for example: Dev, Test, and Production), the same Namespace should be used.

```
[Environment Namespace]

Enter the namespace to associate with this environment:

The namespace identifies the organization using "Reverse Domain Notation" and is used to control
which Apps can be maintained using this environment.

Ex. com.[ORGANIZATION]
>
```

Enter the default app to be created. This is the app that is used as the default by the maintenance programs for app-related system data.

```
[Default App]


Enter the default app:

Configures the app that will be used by default when platform extensions are developed.

Ex. [ORGANIZATION]App
>
```

Select whether to enable integration with QAD EAM requisitions. If enabled, QAD EAM-based requisitions will be used.

This integration is typically disabled if QAD EAM is installed, but not in use or configured.

 This question is only provided when QAD EAM is installed.



To enable this integration, QAD EAM 2017.1 must be installed. If you are using an earlier version of QAD EAM and want to use this functionality, you must upgrade QAD EAM before proceeding.

```
[EAM Requisitions Integration]

Enable EAM Requisitions integration?

[ ] y - Yes
[X] n - No

> n
```

Select whether to install QAD Reporting Framework on Linux. If you select to install this feature, additional prerequisites are required and the location of Wine must be entered (which will configure the following YAB setting: 'qrf.default.wine.dir')

Refer to the section *QAD Reporting Framework on Linux* below for additional details.

```
[QAD Reporting Framework on Linux]

Install QAD Reporting Framework on Linux?

[ ] y - Yes
[ ] n - No

>
```

Select whether to install QAD Sales Configurator. This option must be selected to install the QAD Adaptive-based Sales Configurator app.

```
[QAD Sales Configurator]

Install QAD Sales Configurator?

[ ] y - Yes
[ ] n - No

>
```

Select whether to proceed with the installation or not.

If you continue, the QAD Adaptive packages will be added into the context and configured.

If you do not continue at this point, no changes are made to the context.

```
[Start Installation]

Start installing QAD Adaptive ERP 2024 (Web UI)?

n - No
y - Yes

>
```

## Review and Verify the Packages and Configuration

You can review the updates installed into the context by running the `info` command in the context:

```
> yab -c:ciapps info
```

Review the current configuration using `yab config` to determine if the configuration needs to be adjusted.

Apply any configuration changes in the `configuration.properties` file within your context. To locate the file, run the command:

```
> yab -c:ciapps -pull system-context-locate build/config/configuration.properties
```

Some common configuration options are discussed in [QAD Adaptive Tuning and Configuration Options](#).

## Review the Appserver Configuration Using `yab config` to Determine If the Configuration Needs to Be Adjusted

Review the appserver configuration and ensure that it is configured correctly for your database connections. Refer to the *Review Appserver startup parameters* section of *Preparation Steps* for background.

If you have accepted the default:

```
bootstrap._useparameterfile=true
```

Review the appserver startup parameters defined in your environment:

```
> yab config pas.*.openedge.session-mgr.agentStartupParam
```

If any of these definitions do not include the relevant pf file, review the location of the definition using `-trace` and update it to include the bootstrap connection.

For example, run:

```
> yab config pas.mfg.openedge.session-mgr.agentStartupParam -trace
appserver.mfg.srvrstartupparam=-T /tmp -d mdy -yy 1950 -s 32768 -mmax 10000 -inp 32000 -
rereadnolock -c 30 -D 1000 -Bt 350 -nb 200 -h 25 -tok 4096 -tmpbsize 8 -TB 31 -TM 32 -cpinternal
utf-8 -cpstream utf-8 -cpcoll ICU-UCA -cpcase basic -mmax 20408 -Bt 20000 -tmpbsize 4

[1: build/config/system/environment.properties]

appserver.mfg.srvrstartupparam=${appserver._base.srvrstartupparam} -mmax 20408 -Bt 20000 -
tmpbsize 4

[2: build/config/packages/yab-ee-foundation/1/20/0/9/default/yab-ee-foundation.properties]

# @if bootstrap._useparameterfile=true
pas.mfg.openedge.session-mgr.agentStartupParam=-T "${base}/temp" -pf ${bootstrap.default.
parameterfile} -mmax 16384 -Bt 10000 -tmpbsize 4 -lkwmo 1800
```

Edit `build/config/system/environment.properties` in your context and update the entry to include `${bootstrap.default.parameterfile}`. To locate the file, run the command:

```
> yab -c:ciapps -pull system-context-locate build/config/system/environment.properties
```

Edit the entry:

```
pas.mfg.openedge.session-mgr.agentStartupParam=-pf ${bootstrap.default.parameterfile} -mmax 20408
-Bt 20000 -tmpbsize 4
```

Review the updated value using `yab config` and ensure that it now includes the bootstrap pf file:

```
> yab -c:ciapps config pas.mfg.openedge.session-mgr.agentStartupParam
```

Repeat this for other appserver definitions.

## Run an Update in the Context

The update installs QAD Adaptive, as well as any additional updates in the context. These changes are isolated to the context, but limited to commands that are currently supported in this mode.

These consist of OpenEdge code compilation, schema load preparation, and data load preparation.

The status of the update can be monitored in `build/contexts/prepared/ciapps/logs/yab.log`.

```
> yab -c:ciapps update
```

## Review the Changes in the Context

Use the `explain-apply` command to get a detailed view of what would happen if the changes in the context were applied, without actually applying the changes.

The detailed output of this command is recorded in `build/contexts/prepared/ciapps/logs/yab.log`.

```
> yab -c:ciapps explain-apply
```

## Apply the Context to the Environment

When you are ready to apply the changes prepared in the context to the environment, use the `apply` command. This applies the prepared changes to the environment and executes any deferred steps. This consists of an update in the environment, including downtime.

```
> yab -c:ciapps apply
```

## Complete Any Additional Installation Steps

If you have installed any additional updates during the installation process, such as new patches or version upgrades, refer to the appropriate Installation documents for any additional steps.

# Post-Installation Steps

## Update File Permissions of Newly Added Databases

The QAD Adaptive installation can introduce a number of new database instances. If you did not accept the default `bootstrap._useparameterfile=true` setting, the database files of these new databases must be writable to all users who will initiate sessions, which includes the user who runs Connection Manager sessions and any CHUI users.

If you are using `bootstrap.xml` database connections, edit the file permissions of these databases to ensure that these permissions are set. For example, update the bisgen database:

```
> yab stop
> chmod 666 /dr01/qadapps/qa/databases/bis*
> yab start
```



Depending on the products installed in your initial environment, a number of databases may be created. Ensure that you review and update permissions on each.

## Remove Unneeded Databases

If you are upgrading from a Channel Islands version prior to the March 2019 release, the `crm-app` app has been obsoleted and removed from the QAD Adaptive release. If QAD CRM is not in use, the bisgen database is no longer required and has been removed from the application configuration.

The bisgen database files are not automatically deleted and should be reviewed to ensure that they are no longer required. If no longer required, the database files can be deleted.

If QAD CRM is in use, then the database has not been removed from the configuration and no action is required.

## Re-compile Any Code Not Managed by YAB

The QAD Adaptive installation can include schema changes to existing tables. Therefore, any third-party products or similar code managed outside of YAB that reference the QAD Enterprise Edition schema may need to be re-compiled.

## Apply the PLA License Code

To use the Action Centers in QAD Adaptive, you need to apply the PLA license code that you received from QAD. If you do not apply this license code, you will not have access to Action Centers or KPIs in the Web UI.

Log into the QAD .NET UI and enter the license using the License Registration screen. Add the necessary users to this license. See the "Registering QAD Enterprise Edition" section of the *QAD Enterprise Edition Installation Guide* for step-by-step instructions.



To support the batch update of Action Center data, the batch user defined in YAB must have the PLA license added.

## Install the QAD Reporting Framework Service

If you did not choose to install the "QRF on Linux" feature, then you must install the QAD Reporting Framework Service on one or more Windows machines to allow users to run reports within the QAD Web UI.

See the "Setting Up QAD Reporting Framework Service" section of the *QAD Reporting Framework User Guide* for detailed information on installation and configuration.

## Logi Composer Migration

The QAD Adaptive release no longer includes built-in support for Logi Platform Services (LPS) migration processes. If you need to migrate LPS artifacts from the source environment, you must first install AUX2024 to perform this migration.

## Review and Enable New Action Center KPIs

New action center KPIs are loaded with the Inactive status. When using a new action center, newly added KPIs show no data and the message "KPI is inactive." This is done to save system resources for querying and caching data for the KPIs as there is no need to perform these activities if the KPIs are not actively used.

To enable the use of these KPIs, select the Active checkbox on the KPIs screen. You can perform this step for multiple KPIs at once using the Assign Domains & Entities bulk action in the KPIs screen.

## **Update the Promotion Process for Action Centers**

In QAD Adaptive, Action Centers should be packaged and shared across environments as Configuration Data and should be exported/imported using the standard tools.

## **Additional Post-Installation Steps**

If additional products are installed as part of the QAD Adaptive install, refer to those products' installation guides for additional post-installation steps.

For example, if a Financials Backward Compatibility installation has been performed, a conversion must be executed at this point.

# QAD Reporting Framework on Linux

## Overview

Optionally, QAD Adaptive customers can run the QAD Reporting Framework on the Linux machine through Wine. Previously, the reporting framework server had to be run on a Windows host.

## Prerequisite

Running QAD Reporting Framework on Linux requires that Wine version 5 is installed on your Linux server before the QAD Adaptive installation is executed. Wine must also be on the PATH of the user executing the YAB installation.

Refer to <https://www.winehq.org/> for details of installing Wine on your server. In addition to Wine, X virtual framebuffer (Xvfb) must be installed.

After installing Wine and QAD Adaptive, use the `yab validate` command to confirm that the prerequisites are set up correctly.

## Installation

During the installation of QAD Adaptive, choose to install QAD QRF on Linux, which will create the default configuration with a single "service mode" report agent.

If you choose not to install QRF on Linux during the installation process and wish to use QRF on Linux, you must configure your own Wine prefix with the appropriate prerequisites, before later enabling QRF on Linux.

## Configuration

The QAD Reporting Framework is initially set up with one agent to service report requests. Review the help for QRF using `yab config-help qrf` to access details of the configuration options that are available.

```
# configure a "service mode" agent to process all reports scheduled from the WEBUI and reports
scheduled from the .NET UI associated with the "gadsvc" batch ID
qrf.default.agents.service.report-mode=service
```

```
[mfg@vmlszcco01 aux]$ yab config-help qrf

SETTINGS

    qrfprinters.INSTANCE.uncpath                The QRF Printer's UNCPATH. Ex.
qrfprinters.1.uncpath=\\corp23\rm01

    qrfprinters.INSTANCE.description            The QRF Printer's Description. Ex.
qrfprinters.1.description=Kevin printer

    qrf                                          Configures the QAD Reporting
Framework (QRF) to run on Linux using Wine.

                                                Consult the 'QAD Reporting Framework
User Guide' for details on configuring QRF.

                                                NOTE: Wine and Xvfb must be
installed on the hosts where QRF is deployed.
```

## Security

The standard installation of QAD Reporting Framework on Linux has some built-in security measures such as the enabling of "winetricks sandbox" and the removal of the "z:" drive mapping to the UNIX file system. For an additional level of file system security, you can configure the `runas` setting in YAB:

```

> yab help qrf.runas

SETTINGS

    qrf.INSTANCE.runas      The operating system user and group to run Wine as.

                            Syntax: [USER]:[GROUP]

                            Ex.
                            wine:wineusers

                            This is a security measure to limit what is allowed in a Wine
session. If 'runas' is not configured, Wine is run using
                            the operating system user used to run YAB (i.e. the YAB user) and
files are created in that user's primary group.
                            The 'runas' user and group should be created specifically for
securing Wine.

                            The following restrictions apply:

                            1. The 'runas' user must be a member of the group.
                            2. The YAB user must be a member of the group.
                            3. The YAB user must be able to sudo to the 'runas' user.

                            Furthermore the 'runas' user must be the owner of the 'prefix'
directory. If the prefix directory does not have the
                            correct owner, but the YAB user has the authority to change the
owner, it will be changed, otherwise the administrator
                            is responsible for changing the ownership of the directory (chown).
The administrator may create the directory
                            beforehand if it does not exist.

```

## Configuring Your Own Wine Prefix

QAD Adaptive contains a preconfigured Wine prefix, with all software prerequisites included, that will be used to run QRF on Linux. This prefix has been validated on Red Hat Enterprise Linux Server release 8.

However, if any compatibility issues are found, you can configure your own Wine prefix and use that. To do this, you must first select "no" when asked if you wish to install QRF on Linux during the QAD Adaptive installation and then create your own Wine prefix.

Before creating the Wine prefix, you must have X11 [Forwarding](#) enabled and an [Xming](#) X server running. Refer to the Wine documentation for detailed instructions on creating a prefix. When creating the prefix, cancel the installation of Mono and install the Gecko when asked. When the configuration UI displays, set the "Windows Version" on the Applications tab to "Windows 10".

When the Wine prefix is created, install the following prerequisites into it:

1. .NET Framework 4.5
2. gdiplus
3. allfonts

Create a `tar.gz` file of the prefix and place it in a directory accessible from your QAD Adaptive installation.

Configure the location of this prefix and a QRF agent:

```

qrf.default.prebuilt=/dr01/qadapps/qrf-prefix/qrf-prefix.tar.gz
qrf.default.agents.service.report-mode=service

```

Execute a `yab update` to install the prefix and QRF on Linux.

# Initial Environment Validation

To verify the environment installation, first configure the necessary users to allow access to the QAD Web UI.

Refer to the *QAD Security Administration Guide* for information on how to set up the necessary access.

Once a user has been granted access to the Web UI, log in and verify that the user can access the correct Web UI screens, based on the roles assigned.

- Hybrid browses, such as Sales Orders or Items
- Reports
- Action Centers
- QAD Inbox



Installing QAD Adaptive introduces changes that impact the existing QAD .NET UI and character clients. These should also be validated after the installation.

# QAD Adaptive Installation Considerations

## Extension Database

The initial installation of QAD Adaptive introduces a new extension database, which is configured in YAB using the db.extension instance. The extension database is empty upon creation and is intended to be used for any platform business components created. It is connected to all standard YAB runtime sessions, by default, and should be added to any custom sessions that may execute these platform extensions.

The detailed configuration of this database can be reviewed by executing:

```
> yab config db.extension*
```

## Data Stores and Environment Type

When installing QAD Adaptive, a database, or databases, can be marked as being available as a development data store for platform development. Development data stores can only be configured when the environment is identified as a development environment and should not be configured in a test or production environment.

To configure an environment as a development environment, update the environment type:

```
environment.type=development
```

To configure a database as a development data store, ensure that it is configured as a data store:

```
db.extension.datastore=true
```

Configure the associated data store as type development:

```
datastore.extension.type=development
```

To support platform development, a database must have an SQL Broker enabled. Refer to "Configuring a SQL-92 Secondary Login Broker" in the *QAD Configuration and Administration Guide YAB 1.19* for additional details of how to enable an SQL Broker.



Standard QAD databases, such as db.qaddb, should not be configured as development data stores. The new db.extension is provided for this purpose or a custom database can be used.

If these settings have been changed after the initial installation, they are applied by running:

```
> yab update
```

# Customizing the QAD Adaptive Installation

## Running a Non-Interactive Headless Installation

The main installation section of this document describes how to run an interactive installation of QAD Adaptive. This installation can also be executed in a non-interactive headless mode by providing the necessary configuration before running the installation.

To do this, first extract the QAD Adaptive Installation media and locate the `install.conf` file at the root of the directory. Copy this file into a working directory.

Edit this file and complete the entries in the `USER INTERFACE` section.



Do not edit or remove any of the annotation lines before the entries beginning with the `#` symbol. Changing or removing these lines may result in an unsuccessful or invalid installation.

In the following example, the `openedge.dir` and `java.default.dir` locations have been completed:

```
# -----
# USER INTERFACE
#
# When the installer is run interactively, the installer will prompt for the
# settings in this section.
# -----

# [Progress OpenEdge]
#
# Enter the location of Progress OpenEdge:
# @path
# @validation req
# @validation dir
# @validation openEdge [12.8.9)
# @validation openSSL
openedge.dir=/dr01/openedge

# [Java v11]
#
# Enter the location of Java version 11 (minimum 11.0.5)
# @path
# @validation req
# @validation dir
# @validation java [11.0.5,12) jdk * false openjdk
java.v11.dir=/dr01/java11

# [Java default (v17)]
#
# Enter the location of default Java (minimum 17.0.7)
# @path
# @validation req
# @validation dir
# @validation java [17.0.7,18) jdk * false openjdk
java.default.dir=/dr01/java17

# [Java v21]
#
# Enter the location of Java version 21 (minimum 21.0.7)
# @path
# @validation req
# @validation dir
# @validation java [21.0.7,22) jdk * false openjdk
java.v21.dir=/dr01/java21
```

Review and complete all entries, as required, up to the entry:

```
install-ui-end=
```

Save and exit the file.

Run the installation, skipping the interactive UI, and providing your preconfigured `install.conf` file:

```
> yab -install-update:false -install-ui:false -install-conf:<path to configured install.conf>
install <path to channel-islands-apps-image zip file>
```



The sample command uses a direct installation. To install using a prepared context, refer to the [Installation Steps - Prepared Context](#) section of this document.

The installation will now execute with the preconfigured values and will not prompt you for further input.



Running the installation in a non-interactive, headless mode does not skip the validations on your input values and errors may still be received if invalid values have been provided.

## Customizing Tuning Parameters

Installing QAD Adaptive may introduce a number of new databases into the system and also provides a number of new configuration and tuning parameters. Many of these can be changed after installation, as needed, but you can also provide your customized values before installing.

To do this, first extract the QAD Adaptive Installation media and locate the `install.conf` file at the root of the directory. Copy this file into a working directory.

Edit this file and go to the "QAD Adaptive specific environment tuning" section. Review and edit the values, as required, in this section. If necessary, new values can be added to this section.



Do not edit or remove any of the annotation lines beginning with the `#` symbol. Changing or removing these lines may result in an unsuccessful or invalid installation.



Do not change any of the values in the later "Installer Features" or "Installer Settings" sections. Changing or removing these entries may result in an unsuccessful or invalid installation.

Once complete, save and exit the file. Then, execute the installation, providing your preconfigured `install.conf` file:


```
> yab -c:ciapps -install-update:false -install-conf:<path to configured install.conf> install
<path to channel-islands-apps-image zip file>
```

The installation now executes with your preconfigured values.

# QAD Adaptive Tuning and Configuration Options

## Configuring QAD Adaptive to Use SSL

By default, QAD Adaptive is installed with http access enabled but it is recommended that you disable http access and instead use SSL access.

 If enabling SSL on an existing environment, ensure that the environment is stopped by executing `yab stop` before making any changes.

If you do not stop the environment, the Tomcat instance continues running and must be stopped directly before proceeding. This can be done by running:

```
> ./scripts/tomcat-webui-stop
```

Add the following YAB configuration to disable http access and to enable SSL:

```
tomcat.webui.usessl=true
tomcat.webui.usehttp=false
```

To support SSL, you must also configure an SSL certificate in a Java keystore and configure your Tomcat instance to use this.

Refer to the "Secured Configuration" chapter of the *QAD Security Administration Guide* for steps on how to configure the keystore and SSL certificate.

To configure YAB to use this keystore, add the appropriate configuration and apply it. The section below shows an example of the configuration, but should be adjusted for your environment.

```
tomcat.webui.keystore=/dr01/java/certificate/keystore.jks
tomcat.webui.keystorepassword=samplepassword
```


To apply these changes to your environment, run a `yab update`.

## Secure Configuration

To deploy a fully secure configuration, refer to the *QAD Security Administration Guide*.

## Database Security

To enable database security, refer to the *QAD Security Administration Guide*. For details on configuring database users and table permissions using YAB, refer to the "Configuring Database Security" section of the *QAD Configuration and Administration Guide YAB 1.19*.

 When database security is enabled, the `qaddb` database SQL admin user defined by the properties `dbserver.qaddb.sql.admin.user` and `dbserver.qaddb.sql.admin.password` must be a valid user with the same credentials in all databases that are used as QAD Adaptive datastores.

To review the databases that are defined as datastores, run the YAB command:

```
> yab config db.*.datastore
```

This list should be used as a reference but the value of the properties for standard QAD databases should not be changed.

This requirement is a known OpenEdge driver limitation when querying multiple databases if they have different passwords.

```
> yab update
```

## Installing New Versions of QAD Products

When installing QAD Adaptive, a number of apps are conditionally installed, based on the QAD products that are already installed in your environment. If you add or upgrade products in your environment, it may be necessary to rerun the QAD Adaptive installer to install the related updates.

When you run the QAD Adaptive installer for a second time, the full `install` command is executed, but only the necessary changes are applied. Previously installed packages were not reinstalled. Refer to the Installation Steps section for details on running the installer.

The following sections address the steps required when installing a new version of some QAD products:

- [Installing QAD Production Orders](#)
- [Installing QAD Enterprise Asset Management](#)

# Installing QAD Production Orders

## Introduction

From Adaptive UX 2024, no additional QAD Adaptive steps are required when installing QAD Production Orders. Refer to the *QAD Production Orders Installation Guide* for details on the specific process required when installing QAD Production Orders.

# Installing QAD Enterprise Asset Management

## Introduction

QAD Enterprise Asset Management has a corresponding app included in the QAD Adaptive installation. If you install QAD EAM into your QAD Adaptive environment, you must rerun the QAD Adaptive installation.

## Preparing the Update

Before installing an update, review the release notes for the release that you are installing. Review and apply any required prerequisites before proceeding.

## Installing the Update

To install QAD Enterprise Asset Management, download the media from your Download Center and stage it in a location accessible from your QAD Adaptive ERP location.

Review the release notes for your new QAD EAM release before proceeding with the installation. Apply any necessary YAB configuration changes or other upgrade considerations, as outlined in the release notes.



This example uses the `-install-update:false` option of the `install` command, which allows multiple products to be installed, before applying the changes.

To install QAD EAM, execute the command:

```
> yab -install-update:false install <path to QAD EAM zip>
```

Then, rerun the QAD Adaptive installation:

```
> yab -install-update:false install channel-islands-apps-image-4.12.0.x.zip
```

You can also install both products using the same `install` command:

```
> yab -install-update:false install <path to QAD EAM zip> channel-islands-apps-image-4.12.0.x.zip
```

Review the versions added using `yab info`.

Apply the changes to the environment by executing a `yab update`:

```
> yab update
```



A `yab update` will include restarting the environment so downtime should be scheduled.

## Verifying the Update

After the update has been applied, use `yab info` to verify that the version is now installed in the environment.

Refer to the release notes for the newly installed QAD EAM for additional verification steps or for post-installation steps.

# Installing Updates

After the initial QAD Adaptive installation, future updates can be applied to the environment. These updates may consist of patch releases for individual apps, new releases of individual apps, or a new release of QAD Adaptive.

## Preparing to Apply an Update

Before installing an update, review the release notes for the app that you are installing. If any dependencies are documented, then compare the dependencies to the current environment to determine if additional updates are needed. These dependencies should be installed before, or as part of, this update.

## Installing the Update

To execute the installation of an individual app release:

1. Download the zip file from QAD Fulfillment.
2. Stage the zip file into a directory that is accessible from your QAD Adaptive installation.
3. Execute the command:

```
> yab install <zip file>
```

For example:

```
> yab install sales-app-2.2.1.2.zip
```



The `yab install` command executes a `yab update`, which includes restarting the environment.



This example shows the installation of an update using a direct installation. A prepared context can also be used to install an update. Refer to the *QAD Configuration and Administration Guide YAB 1.19* for details.

When the installation has dependencies or you have multiple updates to install, you can use the `-install-update: false` option to delay the update of the environment until all updates are included. If you take this approach, you must execute a `yab update` to apply the changes and to complete the installation.

For example:

```
> yab install -install-update:false sales-app-2.2.1.2.zip  
> yab install -install-update:false service-app-2.2.1.0.zip
```

Use the `yab info` command to review the versions installed before proceeding:

```
> yab update
```

## Verifying the Update

After the update has been applied, use `yab info` to verify that the versions that are now installed in the environment.

Refer to the release notes for the newly installed apps for additional verification steps.

# Troubleshooting

## Log Files

### Installation Log

When the installation is executing, the output is recorded in a log file in the temp directory. The file name includes the name of the user executing the installation.

Assuming the default temp location and mfg user, the file is:

```
/tmp/install.mfg.log
```

The location of this file is noted when the installation is executed.

If you encounter errors during the initial installation of a product, review this log file for any additional details.

### YAB Log

When executing any YAB command, the status is written to the `build/logs/yab.log` file.

If errors are received when running YAB commands, including a `yab update`, review this log file for any additional details.

For additional details on YAB logging and log options, refer to the *QAD Configuration and Administration Guide YAB 1.19*.

### Application Logs

Each component of the QAD Adaptive installation maintains its own log files. By default, many of these logs are stored in the `build/logs` directory but each location can be configured, as needed.

The Tomcat log files are stored in the logs directory of each Tomcat instance. If errors are encountered, review these logs for additional details.

The location of each log can be confirmed by querying the YAB configuration.

#### Tomcat Logs

Tomcat logs are stored in the logs directory of each Tomcat instance. To find the root directory of each Tomcat instance, query the YAB configuration:

```
> yab config tomcat.*.base
```

#### Appserver Logs

Progress application server logs are located under `${appdir}/servers/pas-*/logs/`.

#### MongoDB Log

To find the MongoDB log file, query the YAB configuration:

```
> yab config mongodb.qad-collaboration.logfile
```

#### Cassandra Log

To view the Cassandra log file:

```
> yab cassandra-logs
```

#### Spark Logs

To find the Spark log files, query the YAB configuration:

```
> yab config spark.masterdefault.env.spark.log.dir
```

```
> yab config spark.slavedefault.env.spark.log.dir
```

## Common Installation Errors

### Incorrect YAB Version

If you have not successfully updated your YAB version to 1.19 before beginning the installation, an error may occur:

```
INSTALL FAILED

BUILD FAILED (0.828 s)

Unable to resolve the validation class [OpenSSLValidation].
```

Install YAB 1.19 to resolve this error and continue.

### Errors when upgrading YAB from 1.15 or older

Use the following approach if this error is encountered when upgrading YAB:

*Caused by: java.lang.NoClassDefFoundError: org/apache/log4j/net/SocketAppender  
at com.qad.yab.loader.LoggingConfigurator.configure(LoggingConfigurator.java:80)*

### Extract the image contents

```
unzip yab-ee-image-[VERSION].zip -d yab-ee-image-[VERSION]
cd yab-ee-image-[VERSION]
unzip data.zip
```

### Use the install.py script to upgrade YAB (/install.py -install-update:false -install-instance:<path to environment>):

```
./install.py -install-update:false -install-instance:/dr01/qadapps/systest/
```

### Dependency Errors

If you receive an error related to dependencies or "The package request could not be satisfied:", review the installed prerequisites and ensure that all necessary updates have been applied.

The package highlighted in the error will provide guidance as to the missing dependency.

The following is a sample error caused by a missing QAD Enterprise Financials dependency:

```
Please review the following constraints:
financials-2.20171.3.11 is out of scope
financials-2.20180.0.104 is out of scope
inventory-2.2.0.207 is required
inventory-2.2.0.207 requires financials[2.20171.0.30,3.0)

Resolving and copying packages           FAIL
Restoring configuration                   OK
Cleaning up                               OK

INSTALL FAILED

BUILD FAILED (28.949 s)

Unable to process the merged properties.
```

### Login/Session Errors During Installation

A number of steps in the QAD Adaptive installation involve logging into the application to call an API. If these steps fail with a login- or session-related error, review the Appserver log files for any errors.

For example, the `fin-sync-run-update` or `metadata-gracore-update` step may fail with these errors.

The error in these cases can vary, but may be similar to:

```
ExecuteAction end: Action= synchronize Full * ReturnStatus= -1 * ErrorMsg= Session.Login
failed: vcSyncLogin=sysadmin, ReturnStatus=-1, vlValid=no
Error=<BLF-348,An error occurred while authenticating user. Look at the following messages for
details.>
Error=<BLF-347,tt_msg_sev = * tt_msg_nbr = * tt_msg_data = * tt_msg_context = *
tt_msg_field = * tt_msg_index =>
Return status: -1
```

```
ExecuteAction end: Action= synchronize Full * ReturnStatus= -98 * ErrorMsg= Session.Login
failed: vcSyncLogin=mfg, ReturnStatus=-98, vlValid=no
S=<,AUTHENTICATION_FAILURE>
Return status: -98
```

## Timeout Error on webapp-webshell-started

The `webapp-webshell-started` process is used to verify that the QAD Adaptive webapp has successfully started before continuing with the remaining steps. If this step errors on a timeout, it means that the `qad-central` webapp in `tomcat.webui` has not started successfully in the allocated time.

Review the `catalina.out` log file of the `tomcat.webui` instance for additional details.

The following is a sample error:

```
2020-01-31 00:46:23,114 DEBUG [main:af62] APPLY - webapp-webshell-started
2020-01-31 00:48:10,624 DEBUG [main:af62] SpinWait - Testing Web Application [webshell] for up to
10:00 m
2020-01-31 00:56:23,530 DEBUG [main:af62] APPLY - webapp-webshell-started ERROR
2020-01-31 00:56:23,532 ERROR [main:af62] Main - BUILD FAILED (35:25 m)
java.lang.RuntimeException: The task has exceeded the configured timeout of [10:00 m]. This
timeout may be individually configured by the [timeout.webapp.start.webshell] property and
currently is defaulted by the [timeout] property.
    at com.qad.build.utilities.SpinWait.waitFor(SpinWait.java:305)
```

## HTTP 404 error on action-center-dashboard-update

If an HTTP 404 error is encountered in the `action-center-dashboard-update`, review the log file of the `tomcat.webui` instance.

The error suggests that the Tomcat instance has not started successfully and additional details will be available in the `catalina.out` log file of the Tomcat instance.

The following is a sample error:

```
2018-09-19 23:43:22,097 DEBUG [main] APPLY - action-center-dashboard-update ERROR
2018-09-19 23:43:22,102 ERROR [main] Main - BUILD FAILED (25:23 m)
java.lang.RuntimeException: HTTP/1.1 404 Not Found
```

## HTTP 401 error on action-center-dashboard-update

If an HTTP 401 error is encountered in the `action-center-dashboard-update`, review the log file of the `tomcat.webui` instance.

The error can be encountered when SSL is enabled on the `tomcat.webui` instance and if the keystore or SSL Certificate has not been correctly configured.

The following is a sample error:

```

2018-09-14 09:57:17,004 DEBUG [main] HttpCallCommand - HttpResponseProxy{HTTP/1.1 401
Unauthorized [Server: Apache-Coyote/1.1, X-Frame-Options: SAMEORIGIN, Cache-Control: no-store,
Content-Type: text/plain;charset=ISO-8859-1, Content-Length: 113, Date: Fri, 14 Sep 2018 13:57:16
GMT] ResponseEntityProxy{[Content-Type: text/plain;charset=ISO-8859-1,Content-Length: 113,
Chunked: false]}}
2018-09-14 09:57:17,073 DEBUG [main] APPLY - action-center-dashboard-update ERROR
2018-09-14 09:57:17,635 ERROR [main] Main - BUILD FAILED (42:11 m)
java.lang.RuntimeException: HTTP/1.1 401 Unauthorized

```

## HTTP 403 error on lps-<module>-referencedataview-update

The following is a sample error in yab.log:

```

2019-08-26 06:38:30,002 DEBUG [main] HttpCallCommand - POST https://vmlszz0102.qad.com:22011/qad-
central/api/analytics/lps/deployment/artifacts?type=referenceDataview&appUri=urn:app:com.qad.
assetmgmt
2019-08-26 06:38:33,099 DEBUG [main] HttpCallCommand - HttpResponseProxy{HTTP/1.1 403 Forbidden
[Server: Apache-Coyote/1.1, Set-Cookie: JSESSIONID=7DB09EA721966A1DF93C9F2BC95BC191; Path=/qad-
central;-Type: application/json;charset=UTF-8, Transfer-Encoding: chunked, Date: Mon, 26 Aug 2019
13:38:33 GMT] ResponseEntityProxy{[Content-Type: application/json;charset=UTF-8,Chunked: true]}}
2019-08-26 06:38:33,100 DEBUG [main] APPLY - lps-assetmgmt-referencedataview-update ERROR
2019-08-26 06:38:33,111 ERROR [main] Main - BUILD FAILED (3:11 h)
java.lang.RuntimeException: HTTP/1.1 403 Forbidden

```

If an HTTP 403 error is encountered in the lps-<module>-referencedataview-update process, review the log file of the tomcat.webui instance.

The encountered error may be similar to:

```

[19/08/26@06:38:33.088-0700] ERROR mfg com.qad.qrview.mvc.controller.
ControllerExceptionHandler: Error thrown by controller. SessionID=..C95BC191 ; Request URL=/qad-
central/api/analytics/lps/deployment/artifacts?type=referenceDataview&appUri=urn:app:com.qad.
assetmgmt
[19/08/26@06:38:33.089-0700] ERROR mfg com.qad.qrview.mvc.controller.
ControllerExceptionHandler: Error details:
org.springframework.security.access.AccessDeniedException: Access is denied
    at org.springframework.security.access.vote.AffirmativeBased.decide(AffirmativeBased.java:
84)
    at org.springframework.security.access.intercept.AbstractSecurityInterceptor.
beforeInvocation(AbstractSecurityInterceptor.java:233)
    at org.springframework.security.access.intercept.aopalliance.MethodSecurityInterceptor.
invoke(MethodSecurityInterceptor.java:65)
    at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:179)
    at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:
213)

```

This error is caused by the YAB user not having the PLA license code assigned.

## PRS Fails to Start

If the prs-status process shows as stopped, refer to the log file defined by the prs.logging.file.name property.

```
> yab config prs.logging.file.name
```

If you encounter an error similar to the following example, ZeroMQ has not been installed. Refer to the Preparation Steps section for details on ZeroMQ installation.

```
java.lang.UnsatisfiedLinkError: /dr01/qadapps/systest/servers/prs/jzmq/libjzmq.so: libjzmq.so.5:
cannot open shared object file: No such file or directory
```