



HelixQAC

Helix QAC Plug-in for Jenkins Manual

Version: 3.3.3

This page Intentionally left blank

This page Intentionally left blank

Important Notice

Disclaimer of Warranty

This document should only be used in conjunction with Helix QAC Plug-in for Jenkins .

Programming Research Ltd have taken due care in preparing this document which it has endeavored to ensure is accurate at the time of printing. However, no liability can be accepted for errors or omissions; nor should the document be considered as an expressed or implied warranty of accuracy or completeness, fitness for a particular purpose, or that the products described perform as specified within.

Copyright Notice

All rights reserved. No part of this document may be reproduced, stored in a retrieval system of any nature, or transmitted in any form or by any means, including photocopying and recording, without the prior written permission of Programming Research Ltd., the copyright owner. If any unauthorized acts are carried out in relation to this copyrighted work, a civil claim for damages may be made and/or a criminal prosecution may result. Copyright ©Programming Research Ltd 2020.

Trademarks

Helix QAC, the Helix QAC logo, Helix QAC for C, Helix QAC for C++ and High Integrity C++ (HIC++) are trademarks of Programming Research Ltd.

"MISRA", "MISRA C" and "MISRA C++" are registered trademarks of HORIBA MIRA Limited, held on behalf of the MISRA Consortium.

"AUTOSAR" is a registered trademark of AUTOSAR GBR, held on behalf of the AUTOSAR Development Partnership.

Yices is a registered trademark of SRI International.

Windows is a registered trademark of Microsoft Corporation.

Contacting Programming Research Ltd

For technical support, contact your nearest Programming Research Ltd. authorized distributor or, alternatively, the Programming Research Ltd head office as follows

by telephone on +44 (0) 1932 888 080

or by webpage: <https://www.perforce.com/support/request-support>

This page Intentionally left blank

This page Intentionally left blank

Contents

Important Notice	3
Disclaimer of Warranty	3
Copyright Notice	3
Trademarks	3
Contacting Programming Research Ltd	3
Contents	1
1 Installation	1
Prerequisites	1
Recommended installation order	1
Jenkins Requirements	1
Supported OS	1
Pipeline installation for Jenkins	2
Installing Helix QAC Plug-in for Jenkins	3
Link Helix QAC Plug-in to Helix QAC	3
2 Using the Helix QAC Plug-in for Jenkins	5
Creating a Freestyle Project	5
Thresholds and Metrics	6
Message Compliance Threshold	6
Project Compliance Threshold	7
File Compliance Threshold	7
Compliance Metric Output from the Helix QAC Plug-in for Jenkins	8
Creating a Pipeline Job	9

This page Intentionally left blank

This page Intentionally left blank

1 | Installation

Note:

It is assumed that you already have the Helix QAC 2020.2 (or later) installed where a Jenkins instance will be able to access it. This should be on the same set of build servers as Jenkins is installed upon.

Prerequisites

Before installing the Helix QAC Plug-in for Jenkins, the following applications must be installed:

- Helix QAC
- Reprise license manager

the following application is optional:

- Helix QAC Dashboard

Recommended installation order

It is recommended that you follow the order below when installing the Helix QAC Plug-in for Jenkins:

1. Install Jenkins
2. Install Pipeline Plugin
3. Install Helix QAC Plug-in for Jenkins
4. Link Helix QAC Plug-in for Jenkins to the Helix QAC Installation

Jenkins Requirements

Your Jenkins version must be 2.164.3 or later for the Helix QAC Plugin to work.

Caution: If you use an older version of Jenkins compared to the latest release, it must already have the Pipeline plugin installed, or else Jenkins will request the latest Pipeline plugin version which will necessitate an upgrade of Jenkins.

Information: For information on installing Jenkins see <https://www.jenkins.io>

Supported OS

- Linux: Red Hat Enterprise Linux 7
- Windows: Windows 10.

Pipeline installation for Jenkins

Login to Jenkins with your administrator login name and password.

To check if the Pipeline Plug-in is installed

From the Jenkins home page do the following:

1. Click **Manage Jenkins**
The Manage Jenkins page is displayed
2. Click **Manage Plugins**
The Plugin Manager page is displayed
3. Open the **Installed** tab
If the Pipeline Plug-in is already installed then continue with [Installing Helix QAC Plug-in for Jenkins](#)

To Install the Pipeline Plug-in.

From the Jenkins home page do the following:

1. Click **Manage Jenkins**
The Manage Jenkins page is displayed
2. Click **Manage Plugins**
The Plugin Manager page is displayed
3. Open the **Available** tab
4. Search for "**Pipeline**"
5. Select **Pipeline**

Note: If you see red warning text then check that you have the latest version of Jenkins installed.

6. Click **Download now and install after restart.**
Jenkins will then download all plug-in dependencies and then the Pipeline plug-in itself.
You may be prompted to restart Jenkins to complete the installation

Note: Please refer to Jenkins documentation if there were any problems at this stage.

Installing Helix QAC Plug-in for Jenkins

From the Jenkins home page do the following:

1. Click **Manage Jenkins**

The Manage Jenkins page is displayed

2. Click **Manage Plugins**

The Plugin Manager page is displayed

3. Open the **Available** tab

4. Search for "**Helix QAC**"

5. Select Helix QAC Plug-in for Jenkins

6. Click **Download now and install after restart.**

Jenkins will then download all plug-in dependencies and then the Helix QAC Plug-in for Jenkins itself.

You may be prompted to restart Jenkins to complete the installation

Link Helix QAC Plug-in to Helix QAC

From the Jenkins home page do the following:

1. Click **Manage Jenkins**

The Manage Jenkins page is displayed

2. Click **Global Tool Configuration**

The Global Tool Configuration page is displayed

3. Select **Helix QAC Installations**

4. The Helix QAC Installations section is displayed

5. Enter a name (it can be anything you like) that identifies a Helix QAC installation and its version in the **Helix QAC installation name** field.

6. Enter the absolute path to the root of the Helix QAC installation folder in the Helix QAC installation path field.

Note: The direction of separator is dependent on your Jenkins OS. Use '\' (backslash) on Windows and '/' (forward slash) on Linux.

7. Click **Save**

If you have more than one version of Helix QAC installed, you can repeat this process by clicking **Add Helix QAC** and repeating steps 1–7 until all your Helix QAC Installations have been linked.

This page Intentionally left blank

This page Intentionally left blank

2 | Using the Helix QAC Plug-in for Jenkins

You may use the Helix QAC Plug-in for Jenkins in two ways. As part of Jenkins Freestyle projects, or as part of Pipeline projects.

Creating a Freestyle Project

To create a freestyle project

From the Jenkins home page do the following:

1. Click **New Item**
The **New Item** page is displayed
2. Enter a name for the Project.
3. Click **Freestyle Project**
4. Click **OK**

Information: For information on configuring Freestyle Projects please see the general Jenkins Documentation

Before completing the Freestyle Project, we need to create a **Post Build Action** to request Helix QAC runs and creates a Helix QAC Report.

To create a Post Build Action

From the Freestyle Projects page do the following:

5. Click **Add post-build action**

The post build page is displayed.

6. Select **Helix QAC Report**

The Helix QAC Installation field will show a dropdown list pre-populated with all the Helix QAC installations you configured in the Global Tool Configuration.

Only the Helix QAC Project is required to then be populated for the post build action to be successful.

Note: If you have not created a Helix QAC project for a source code project at this point, it must be initially created via the Helix QAC standalone UI, or one of the IDE plugins.

Each of the other text fields or, non-report options, have help icons on the right which explain their use.

7. Click **Save** to save your changes and return you to the **New Item** page.
8. Click **Apply** to save your changes and stay on the **Post Build** page.

Thresholds and Metrics

The number of messages generated by Helix QAC can be compared against thresholds for build stability. If a threshold is exceeded, the build status is set to unstable. This can be used as a gateway to prevent subsequent tasks from running.

The messages are organized into message levels according to the severity of the message, with Level 0 being the lowest.

The thresholds can be set to operate at a message level, allowing the severity of the messages to be considered. Code that contains low-severity maintenance and style issues may pass but code that contains critical issues would fail.

Thresholds are added as part of the Helix QAC post build action, see [To create a Post Build Action](#). There are three types:

- File Compliance Threshold
- Message Compliance Threshold
- Project Compliance Threshold

Note: The Helix QAC Plug-in for Jenkins will be set by default to **No thresholds**.

Message Compliance Threshold

You can choose either **Maximum messages target** or **Continuous improvement**.

Maximum messages target

Choosing this will set the total number of messages that are allowed on all levels for all the files in the project. Messages in header files are de-duplicated (i.e. they are only counted once and not for each source file with which the header is included).

Continuous improvement

Selecting this states that the **Message Compliance value** for the build will be less than or equal to the **Message Compliance value** found in the previous build.

The **Message Compliance value** is the total number of messages for all the files in the project that are at the **Message Threshold Level** or above.

Information: If the **Message Compliance value** for the build is higher than the **Maximum messages target** then the build is marked as unstable.

Message Threshold Level

This sets the message level from which the message total is calculated. Only messages from this level and above are summed to give the message total that is used for the **Messages Threshold** comparison.

Note: The **File Compliance Threshold** and **Project Compliance Threshold** are not affected by this setting.

Information: If the **Message Compliance value** for the build is higher than the '**Message Threshold Level**', the build is marked as unstable.

Note: If any of these thresholds are not met, then the build will be marked as unstable. Only configuration errors or analysis errors will mark the build as failed.

Project Compliance Threshold

You can choose either **Project compliance target** or **Continuous improvement**.

Project compliance target

Choosing this sets the **Project compliance level** for the project.

The **Project compliance level** is the percentage of message groups (a coding standard 'rule') that have no messages in the project.

Information: If the **Project Compliance level** for the build is lower than the **Project Compliance target** then the build is marked as unstable.

Continuous improvement

Selecting this states that the **Project Compliance level** for this build will be greater than or equal to the **Project Compliance level** for the previous build.

Information: If the **Project Compliance level** for the build is lower than the **Project Compliance level** for the previous build the build is marked as unstable.

If the **File Compliance level** and **Project Compliance level** are close, then each file is violating more or less the same rules.

Note: If any of these thresholds are not met, then the build will be marked as unstable. Only configuration errors or analysis errors will mark the build as failed.

File Compliance Threshold

You can choose either 'File Compliance target' or 'Continuous improvement'.

File Compliance target

Choosing this sets the minimum **File Compliance level** for the project.

The **File Compliance level** is the mean across all files of the percentage of message groups that have no messages in each file.

Information: If the **File Compliance level** for the build is lower than the **File Compliance target** then the build is marked as unstable.

Continuous improvement

Selecting this states that the **File Compliance level** for this build will be greater than or equal to the **File Compliance level** for the previous build.

Information: If the File Compliance level is lower than the File Compliance level for the previous build then the build is marked as unstable.

A high **File Compliance level** and low **Project Compliance level** indicate that each file violates only a small number of rules but that each file violates a different rule.

If the **File Compliance level** and **Project Compliance level** are close, then each file is violating more or less the same rules.

Note:

If any of these thresholds are not met, then the build will be marked as unstable.

Only configuration errors or analysis errors will mark the build as failed.

Compliance Metric Output from the Helix QAC Plug-in for Jenkins

The plug-in uses configured settings to call Helix QAC to analyse as part of a Freestyle project post build action, or as a step within a Pipeline job.

The output from Helix QAC is captured and utilized to be presentable to users within Jenkins.

Note: A Freestyle project, or Pipeline job that invokes Helix QAC Analysis, must successfully run twice before results can be viewed.

When viewing a Freestyle/Pipeline job page, the results will be displayed based on the configuration of the Freestyle or Pipeline project.

Clicking **Helix QAC Results** will just display the graphs collectively on screen without the peripheral information. The report output types match those from Helix QAC output and are described in the Helix QAC manual.

Creating a Pipeline Job

Pipeline jobs are discrete units of work that can be chained together and configured to require all/some upstream jobs to have been successful before execution. They may also start multiple downstream jobs depending on the outcome. This provides more flexibility to configure a Continuous Integration / Continuous Delivery environment according to the needs of your business. Allowing a far greater level of automation than a Freestyle project will permit.

Note: Pipeline job doesn't know the outcome of upstream jobs – the assumption is that the upstream jobs met whatever criteria was necessary to invoke the job currently running.

To create a Pipeline Job

From the Jenkins home page do the following:

1. Click **Pipeline**

2. The **pipeline project's configuration** screen is displayed

Refer to the Jenkins documentation for information on the general Pipeline configuration options.

3. Click **Pipeline Syntax** in the **Pipeline** section

The **Pipeline Syntax** screen is displayed in a new browser tab.

4. In the **Sample Step** dropdown select **qacReport: Helix QAC Report**

The Helix QAC Installation field will show a dropdown list pre-populated with all the Helix QACk installations you configured in the Global Tool Configuration.

Only the Helix QAC Project is required to then be populated for the post build action to be successful.

Note: If you have not created a Helix QAC project for a source code project at this point, it must be initially created via the Helix QAC standalone UI, or one of the IDE plugins.

The text fields or, non-report options, have help icons on the right which explain their use.

Once configured.

5. Click **Generate Pipeline Script**, which will populate the text field below.

Example :If we configured the Pipeline options with Analysis Settings, Stop Analysis On Failure selected then Generated a HIS Metrics Report – the generated output would look like this

```
qacReport runWhenSuccess: false, sourceQAFramework: [addBuildNumber: true,
analysisSettings: true, assembleSupportAnalytics: false, cmaProjectName: '',
customCpuThreads: false, customLicenseServerAddress: '',
downloadUnifiedProjectDefinition: false, enableDependencyMode: false, generateCrr:
false, generateHis: true, generateMdr: false, generatePreprocess: false, generateReport:
false, generateReportOnAnalysisError: false, generateSup: false, loginToQAV: false,
maxNumThreads: '', performCrossModuleAnalysis: false, projectConfiguration: '',
publishToQAV: false, qaInstallation: 'Helix-QAC-2020.1', qaProject:
'C:\\Perforce\\kazlib-prqa', qaVerifyProjectName: '', reuseCmaDb: false, stopWhenFail:
true, unifiedProjectName: '', uploadSnapshotName: '', uploadSourceCode: 'NONE',
uploadWhenStable: false, useCustomLicenseServer: false, useDiskStorage: false]
```

Tip: Copy this text somewhere safe in addition to the clipboard.

Switch back to the **pipeline project's configuration** screen.

6. In the script editor dropdown select **Hello World**.
7. Replace the entry `echo 'Hello World'` with the script previously generated .

```
pipeline {
  agent any

  stages {
    stages('Hello') {
      steps {
        qacReport runWhenSuccess: false, sourceQAFramework: [addBuildNumber:
true, analysisSettings: true, assembleSupportAnalytics: false, cmaProjectName: '',
customCpuThreads: false, customLicenseServerAddress: '',
downloadUnifiedProjectDefinition: false, enableDependencyMode: false, generateCrr:
false, generateHis: true, generateMdr: false, generatePreprocess: false, generateReport:
false, generateReportOnAnalysisError: false, generateSup: false, loginToQAV: false,
maxNumThreads: '', performCrossModuleAnalysis: false, projectConfiguration: '',
publishToQAV: false, qaInstallation: 'Helix-QAC-2020.1', qaProject:
'C:\\Perforce\\kazlib-prqa', qaVerifyProjectName: '', reuseCmaDb: false, stopWhenFail:
true, unifiedProjectName: '', uploadSnapshotName: '', uploadSourceCode: 'NONE',
uploadWhenStable: false, useCustomLicenseServer: false, useDiskStorage: false]
      }
    }
  }
}
```

```
}  
}
```

8. Click **Save** to save your changes and return you to the **Pipeline Syntax** page.
9. Click **Apply** to save your changes and stay on the **pipeline project's configuration** page.

This page Intentionally left blank

This page Intentionally left blank