



# Automatic Release Automation

ONE Automation

## JENKINS Plugin User Guide

Version: 1.2.0

Publication Date:

**Automatic Software GmbH**

# Copyright

Automic® and the Automic logo® are trademarks owned by Automic Software GmbH (Automic). All such trademarks can be used by permission only and are subject to the written license terms. This software/computer program is proprietary and confidential to Automic Software and is only available for access and use under approved written license terms.

This software/computer program is further protected by copyright laws, international treaties and other domestic and international laws and any unauthorized access or use gives rise to civil and criminal penalties. Unauthorized copying or other reproduction of any form (in whole or in part), disassembly, decompilation, reverse engineering, modification, and development of any derivative works are all strictly prohibited, and any party or person engaging in such will be prosecuted by Automic Software.

No liability is accepted for any changes, mistakes, printing or production errors. Reproduction in whole or in part without permission is prohibited.

© Copyright Automic Software GmbH. All rights reserved.

Jenkins is a registered trademark of *Software in the Public Interest, Inc.*

Other names may be trademarks of their respective holders.

---

---

# Contents

<b>1 Introduction</b> .....	<b>1</b>
1.1 About Jenkins .....	1
1.2 About ARA .....	1
<b>2 Links &amp; Guides</b> .....	<b>3</b>
<b>3 Installing the ARA Plugin</b> .....	<b>4</b>
<b>4 Updating the ARA Plugin</b> .....	<b>5</b>
<b>5 Using the ARA Plugin</b> .....	<b>7</b>
5.1 Creating a Package in ARA from a Post-build Action in Jenkins .....	7
5.2 Executing General Workflows in ARA from a Post-build Action in Jenkins .....	11
5.3 Executing Application Workflows in ARA from a Post-build Action in Jenkins .....	12
<b>6 Troubleshooting</b> .....	<b>15</b>
<b>7 About Automic Software, Community and Support</b> .....	<b>16</b>
7.1 About Automic .....	16
7.2 Automic Community .....	17
7.3 Download Center .....	17
7.4 Marketplace .....	17
7.5 Technical Support .....	18
7.6 Training Services .....	18

---

# 1 Introduction

The ARA plugin for Jenkins allows to automatically create packages and trigger general and application deployment workflows in ARA after a build has ended successfully in Jenkins.

## Target Audience

This guide is intended for administrators and managers who wish to integrate powerful tools for agile development, continuous integration and automated deployment with Jenkins and Atomic Release Automation.

Readers should have basic knowledge of the administration of Jenkins and Atomic Release Automation.

## 1.1 About Jenkins

*Jenkins is a project-oriented build, test and track software for continuous integration (CI) and continuous delivery (CD).*

Jenkins helps developers find and solve defects in code and automate build testing. Furthermore, it makes easier to integrate changes and speeds up the software development process. ARA can seamlessly integrate with Jenkins to manage the post-build steps of an integrated deployment.

### See also:

- <https://wiki.jenkins-ci.org/display/JENKINS/Meet+Jenkins>

## 1.2 About ARA

*Atomic Release Automation is an end-to-end solution for planning, coordinating and automating software release processes, including automated deployment of applications across large-scale server environments and covers all mentioned use cases.*

### Purpose

The purpose of Atomic Release Automation is to unify enterprise application and infrastructure automation functionality onto a single platform - without the need of managing multiple tools. Users first architect and control the execution of application process flows, then orchestrate the underlying infrastructure to meet required service levels.

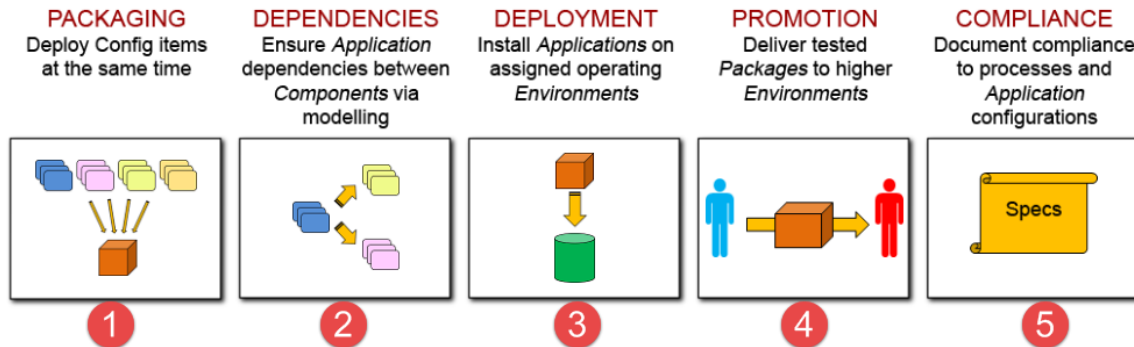
ARA streamlines application release workflow managements for even the most highly-scaled environments while reducing deployment cost across the organization including development, testing and production.

The particular aim is to enable consistent, traceable and auditable deployment for multi-tier applications across heterogeneous and distributed IT infrastructures, reaching from physical or virtual to cloud computing platforms. To improve efficiency, quality and release cycle time, a highly automated system like Atomic Release Automation is preferred.

---

## Main Functions

ARA does the following:



1. It allows modelling the various components of an application, which are defined as meta-data elements in ARA, with all necessary technical references to the physical elements, and then the packaging of these components for simultaneous deployment. By using the Automation Engine's powerful synchronization capabilities, it is able to deploy, at the same time, the artifacts that are intended to stay together in a single package.
2. As components of an application generally carry important dependencies across one another, ARA provides native capabilities to build those dependencies via modelling tools, so that the application is deployed in the right sequence. A dependency of a component on another can be enforced using ARA's modeling tools, rather than makeshift orchestration mechanisms.
3. ARA deploys the right components to the right environments and it does so natively. Teams no longer have to worry about pushing the proper artifacts to their assigned systems. ARA does that for them.
4. ARA is capable of staging by allowing users to design structured deployment environments. Hence, each environment like DEV, QA and PRODUCTION exists in ARA as distinct entities and the same component can be deployed seamlessly across these environments without the need for manual handling and configuration.
5. ARA is particularly adept at enforcing governance and compliance. It provides all the necessary tools to comply with a team's internal processes without having to design checks and balances and fail-safe mechanisms. It will further audit deployments to ensure that everything goes smoothly and without error, all the time.

---

## 2 Links & Guides

You can find links to resources for Jenkins and Automic Release Automation in the table below.

Component	Link
ARA for Jenkins plugin	The ARA for Jenkins plugin can be downloaded from the <a href="#">Automic Marketplace</a> .
Automic Release Automation	You can download ARA v12 from: <a href="http://downloads.automic.com/">http://downloads.automic.com/</a> . Compatibility Information and system requirements can be found on our website Automic Compatibility Checker ( <a href="https://docs.automic.com/tools/compatibility_checker">https://docs.automic.com/tools/compatibility_checker</a> ). ARA v12 documentation is available on <a href="#">Automic Docs</a>

---

# 3 Installing the ARA Plugin

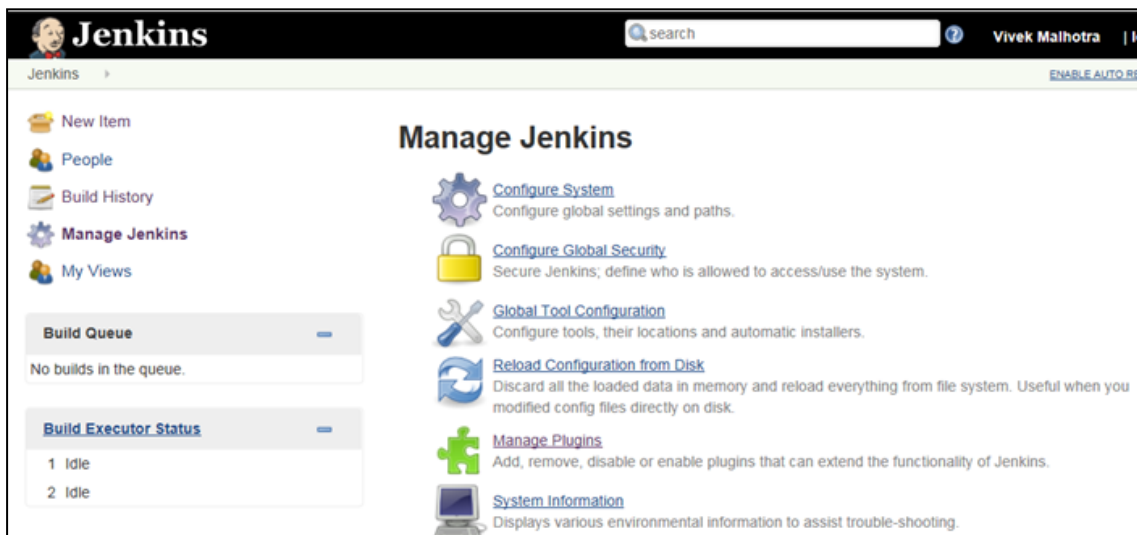
The ARA plugin can be easily installed by following the steps below.

## Preparation Steps

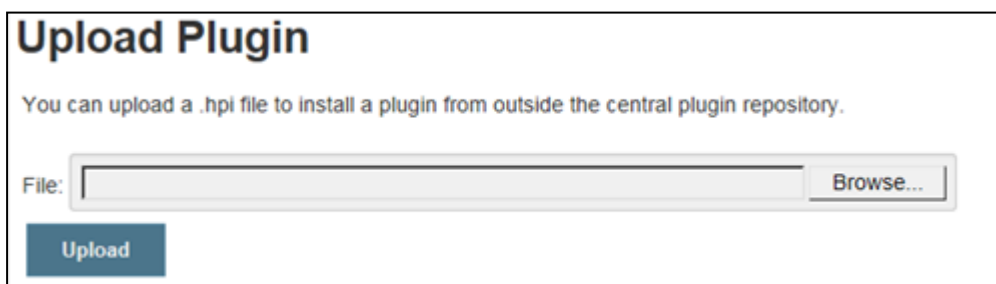
1. Download the ARA for Jenkins plugin from the [Automic Marketplace](#).
2. Extract the file to a folder and locate the `PLG.AUTOMIC_JENKINS.hpi` file.

## To Install the ARA Plugin

1. Log in to Jenkins and select **Manage Jenkins**.
2. Click on **Manage Plugins** as shown below:



3. Select the **Advanced** tab and scroll down to the **Upload Plugin** section.



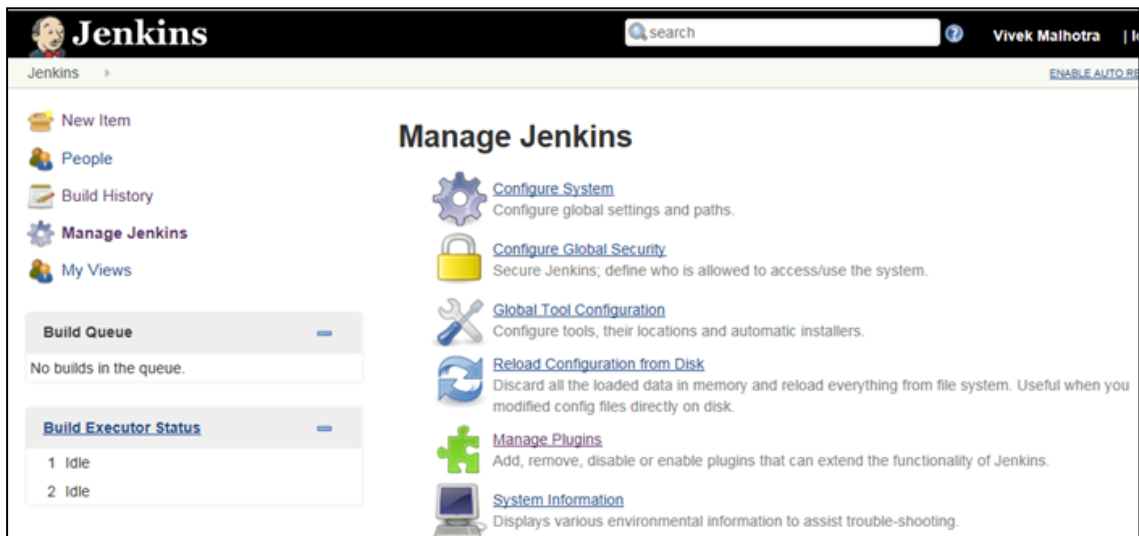
4. Browse to the location where the extracted `.hpi` file was saved.
  5. Select the file and click **Upload**.
  6. Restart the servlet container.
- ✓ Now your Jenkins for Automic Release Automation integration is ready for operation.

# 4 Updating the ARA Plugin

Updates can be performed by simply uninstalling the existing version of the plugin and installing a new one.

## To Update the ARA Plugin

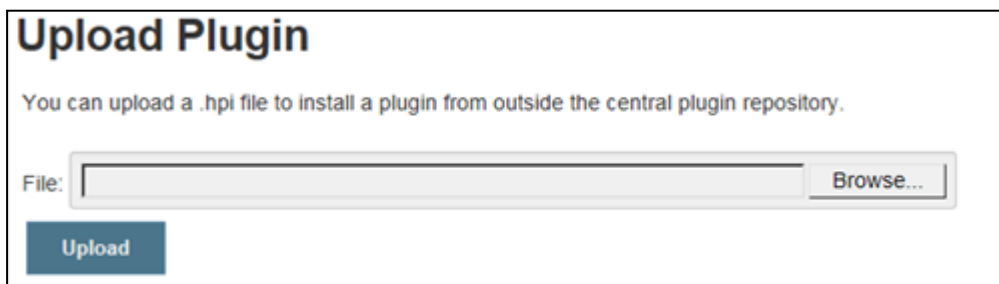
1. Download the latest version of the ARA for Jenkins plugin from the [Automic Marketplace](#).
2. Extract the file to a folder and get the latest `PLG.AUTOMIC_JENKINS.hpi` file.
3. Log in to Jenkins and select **Manage Jenkins**.
4. Click on **Manage Plugins** as shown below:



5. Select the **Installed** tab.
6. Locate the `PLG.AUTOMIC_JENKINS` plugin and click **Uninstall**.



7. Select the **Advanced** tab and scroll down to the **Upload Plugin** section.



8. Browse to the location where the extracted `.hpi` file was saved.



9. Select the file and click **Upload**.
10. Restart the servlet container.

✓ The ARA plugin has been successfully updated.

# 5 Using the ARA Plugin

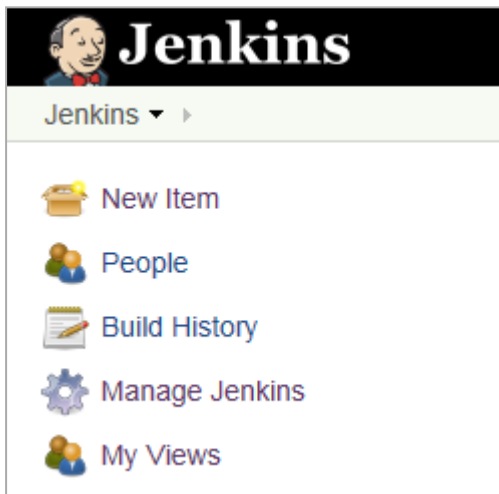
*You can use the ARA plugin from any existing or new Jenkins jobs.*

## 5.1 Creating a Package in ARA from a Post-build Action in Jenkins

*This topic describes how to create ARA packages from Jenkins after a build has ended successfully.*

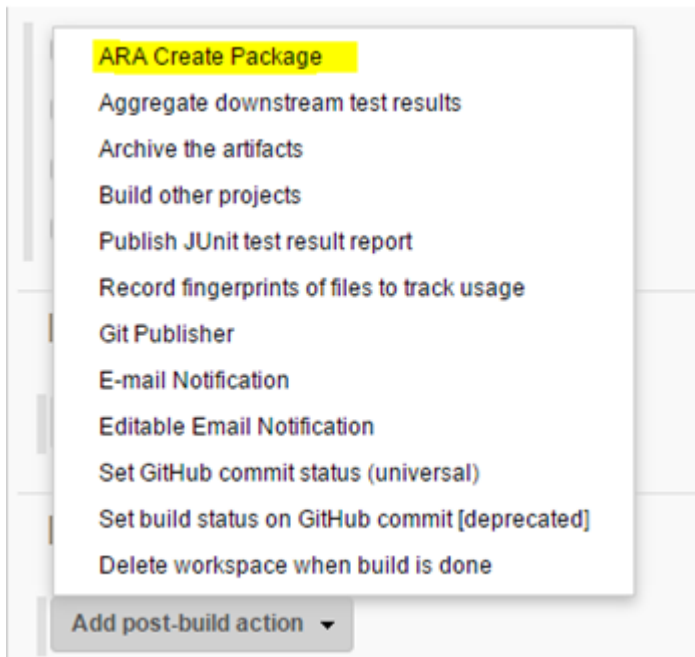
### *To Create a Package from Jenkins*

1. Go to the main page and create a new build job (**New Item**) or use an existing one.

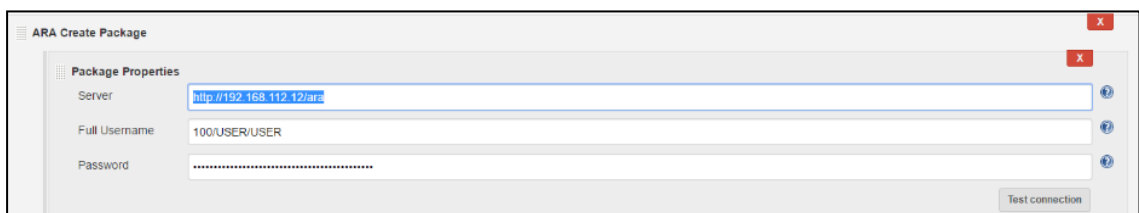


2. If you are creating a new build job, enter the necessary information in the **General**, **Source Code Management**, **Build Triggers** and **Build** sections.
3. At the bottom of the page, click the **Add post-build action** button in the **Post-build Actions** section.

4. Add **ARA Create Package** as a post-build action.






5. Enter the URL of the Automic Release Automation installation. E.g.: `http://192.000.000.00/ara`



6. Enter the user name used to log in to Automic Release Automation with the following format: `client/username/department` or `system/client/username/department`. E.g. `1000/USER/DEP` or `AUTOMIC/1000/USER/DEP`.
  7. Enter the password used to log in to Automic Release Automation.
  8. Click **Test connection** to verify the connection. If the connection details are correct, the plugin pulls all required data from ARA.
-

9. Once the connection has been checked, fill in the other fields as follows:

1. **Package Name:** name of the package which will be created in Automatic Release Automation.
  -  This value can also contain variables (e.g. Build environment variable). Variables must start with \$.
2. **Package Type:** Deployment or Generic (additional types may be added by the ARA administrator).
3. **Application:** select the application by its name from a list provided by Automatic Release Automation.
4. **Folder:** the folder in which the package should be created.
  -  Folders are used in Automatic Release Automation to manage user rights
5. **Owner:** select the owner by its name from a list provided by Automatic Release Automation.
  -  Owners are used in Automatic Release Automation for filtering entities and to manage user rights as they are stored in folders
6. **Components:** by default, all components of the application are assigned to the package. If you click **Add Component**, only the selected components will be assigned.
  1. Optionally, you can specify one or more conditions for each component, so that only the ones which comply with the conditions are added to the package. To do so.
    1. Click **Add condition**.
    2. Select the condition operation from the dropdown list and provide a

value.

The screenshot shows a configuration window with a red 'X' in the top right corner. It contains two rows: 'Build Log' and 'Value'. The 'Build Log' dropdown is open, showing 'Contains' selected. The 'Value' dropdown is also open, showing 'Contains' and 'Does not contain'. Below the dropdowns, a red error message reads 'The value is mandatory'. At the bottom left, there is an 'Add condition' button. Information icons are visible on the right side of the dropdowns.

Values are case insensitive and can contain regular expressions and variables. Variables must start with \$. If the value is not specified, the corresponding condition will be skipped.

A component is added into a package only if all conditions are met.

2. **Properties:** click **Add Property** if you want to overwrite the custom properties of the package.
3. **Dynamic Properties:** click **Add Dynamic Property** if you want to overwrite the dynamic properties of the package.

Custom and dynamic property values can also contain variables (e.g. Build environment variable). Variables must start with \$.

10. You can create multiple ARA packages by clicking **Add ARA Package** as many times as needed.

11. Click **Save**.

✓ As of now, after each successful build the packages defined in the **ARA Create Package** post-build action will be automatically triggered in ARA.

### Post Build Step Behavior

Please note the following:

- If the build fails, ARA packages will not be created at all.
- If the package creation failed for some reason, the build will fail as well.
- If multiple packages are added and one of them cannot be created, the subsequent packages will not be created either. E.g. If you want to create Package1, Package2, Package3 and Package4 and for some reason the creation of Package2 fails, the subsequent packages (Package3 and Package4) will not be created and the build will be marked as failed.
- Generic packages can also be created. For this type of packages neither applications nor components will be associated to the package created.
- If the association of components/properties/dynamic properties to the package fails, the build will fail but the package will still be created in ARA.

## 5.2 Executing General Workflows in ARA from a Post-build Action in Jenkins

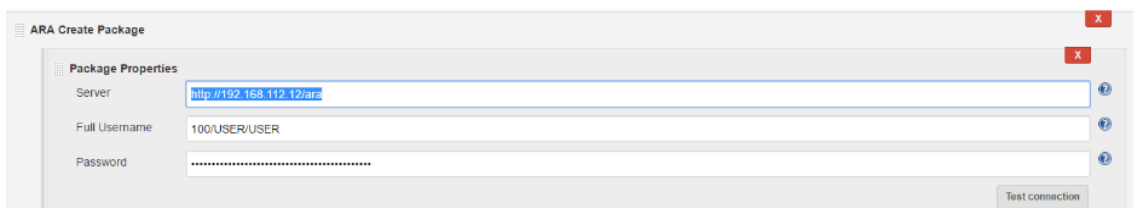
This topic describes how to automatically execute ARA general workflows from Jenkins after a build has ended successfully.

### To Define Execution Properties of a General Workflow

1. Go to the main page and create a new build job (**New Item**) or use an existing one.



2. If you are creating a new build job, enter the necessary information in the **General**, **Source Code Management**, **Build Triggers** and **Build** sections.
3. At the bottom of the page, click the **Add post-build action** button in the **Post-build Actions** section.
4. Add **ARA General Workflow Execution** as a post-build action.
5. Enter the URL of the Automatic Release Automation installation. <http://192.000.000.00/ara>



6. Enter the user name used to log in to Automatic Release Automation with the following format: system/client/name/department. E.g. 1000/USER/DEP or AUTOMIC/1000/USER/DEP.
7. Enter the password used to log in to Automatic Release Automation.
8. Click **Test connection** to verify the connection. If the connection details are correct, the plugin pulls all required data from ARA.

9. Once the connection has been checked, review or edit the other fields as follows:

The screenshot shows the configuration form for an ARA workflow execution. The fields are as follows:

- Workflow:** UlansApp
- Start At:**  Now  At  Queue
- Execute At:** 2016-11-30 16:23
- Queue:** [Empty]
- Manual Confirmation:**  Yes  No
- User/Group:**  User  Group
- Installation Mode:**  Overwrite  Skip
- Properties:**
  - Property Name: Version, Property Value: 1.639
  - Property Name: Owner, Property Value: JenkinsInc

1. Select the name of an existing ARA workflow from the dropdown list.
  2. Select the execution start time:
    - **Now**
    - **At:** if selected, you have to specify the planned start time.
    - **Queue:** if selected, you have to specify the queue (the container for workflow executions) where you want to add the execution.
  3. **Manual Confirmation:** select if manual confirmation is required for the execution.
  4. **User/Group:** if manual confirmation is needed, you have to define a user or group to confirm the execution.
  5. **Installation Mode:** you can specify whether you want already successful installations to be overwritten or skipped. The decision whether a component is installed or skipped is based on the installation records (see *ARA User Guide - Working with Packages*), which are automatically created during all deployments.
  6. **Properties:** enter the name and value of the dynamic properties to be applied to the execution workflow.
10. Click **Validate** to ensure that the data you entered is in a valid format.
  11. You can create multiple ARA general workflow executions by clicking **Add ARA General Workflow Execution** as many times as needed.
  12. Click **Save**.

- ✓ As of now, after each successful build the general deployment workflows configured in the **ARA General Workflow Execution** post-build action will be automatically triggered in ARA.

## 5.3 Executing Application Workflows in ARA from a Post-build Action in Jenkins

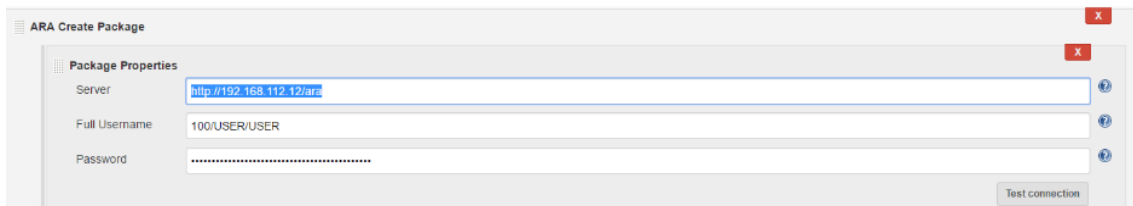
*This topic describes how to automatically execute ARA application workflows from Jenkins after a build has ended successfully.*

### *To Define Execution Properties of an Application Workflow*

1. Go to the main page and create a new build job (**New Item**) or use an existing one.



2. If you are creating a new build job, enter the necessary information in the **General**, **Source Code Management**, **Build Triggers** and **Build** sections.
3. At the bottom of the page, click the **Add post-build action** button in the **Post-build Actions** section.
4. Add **ARA Deployment Workflow Execution** as a post-build action.
5. Enter the URL of the Automic Release Automation installation.



6. Enter the user name used to log in to Automic Release Automation with the following format: system/client/name/department.
7. Enter the password used to log in to Automic Release Automation.
8. Click **Test connection** to verify the connection. If the connection details are correct, the plugin pulls all required data from ARA.



9. Once the connection has been checked, review or edit the other fields as follows:

The screenshot shows the 'ARA General Workflow Execution' configuration form. It contains the following fields and options:

- Server:** http://localhost/ara
- Full Username:** 100/AUTOMIC/AUTOMIC
- Password:** Masked with asterisks
- Application:** UlansApp
- Workflow:** UlansApp
- Package:** 1.0
- Profile:** MSI-Installation
- Start At:** Radio buttons for Now, At, Queue
- Execute At:** 2016-11-30 16:23
- Queue:** Dropdown menu
- Manual Confirmation:** Radio buttons for Yes, No
- User/Group:** Radio buttons for User, Group
- Installation Mode:** Radio buttons for Overwrite, Skip
- Properties:**
  - Property Name: Version, Property Value: 1.639
  - Property Name: Owner, Property Value: JenkinsInc

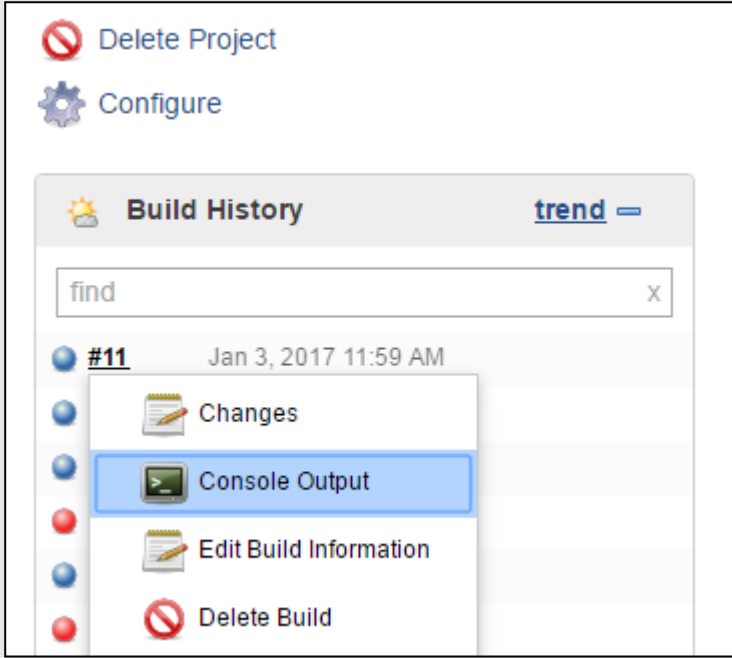
A 'Test connection' button is located to the right of the Password field.

1. Select the name of the application you want to deploy.
2. Select the name of the ARA workflow from the dropdown list.
3. Select the name of the package (an instance of the application) from the dropdown list.
4. Select the name of the deployment profile (the entity that links the application to one specific environment).
5. Select the execution start time:
  - **Now**
  - **At:** if selected, you have to specify the planned start time.
  - **Queue:** if selected, you have to specify the queue (the container for workflow executions) where you want to add the execution.
6. **Manual Confirmation:** select if manual confirmation is required for the execution.
7. **User/Group:** if manual confirmation is needed, you have to define a user or group to confirm the execution.
8. **Installation Mode:** you can specify whether you want already successful installations to be overwritten or skipped. The decision whether a component is installed or skipped is based on the installation records (see *ARA User Guide - Working with Packages*), which are automatically created during all deployments
9. **Properties:** enter the name and value of the dynamic properties to be applied to the execution workflow.
10. Click **Validate** to ensure that the data you entered is in a valid format.
11. You can create multiple ARA general workflow executions by clicking **Add ARA General Workflow Execution** as many times as needed.
12. Click **Save**.

✓ As of now, after each successful build the application deployment workflows configured in the **ARA Deployment Workflow Execution** post-build action will be automatically triggered in ARA.

## 6 Troubleshooting

This section provides information about general troubleshooting steps.

Error	Possible solution
<p>I can't find the build output.</p>	<ul style="list-style-type: none"> <li>To check the build output, click the <b>Console Output</b> of the build.</li> </ul>  <p>The screenshot shows a 'Build History' window with a search bar containing 'find'. Below the search bar, a list of build entries is shown. The first entry is '#11' dated 'Jan 3, 2017 11:59 AM'. A context menu is open over this entry, with 'Console Output' highlighted in blue. Other options in the menu include 'Changes', 'Edit Build Information', and 'Delete Build'. Above the menu, there are 'Delete Project' and 'Configure' options.</p>
<p>I don't know where the logs for the ARA create package post build step are displayed.</p>	<ul style="list-style-type: none"> <li>The logs are displayed as follows:           <pre>" .....Post build step (ARA create package) starts....."  //post build step log  ".....Post build step (ARA create package) ends....."</pre> </li> </ul>

# 7 About Automic Software, Community and Support

*This topic introduces the Automic Software company and how to leverage the full potential of our solutions to you.*

You can also obtain Automic documentation online from [docs.automic.com](https://docs.automic.com).

## 7.1 About Automic

*Automic Software is dedicated to business automation.*

Automic is the world's most comprehensive platform in automating businesses. Founded 1985, Automic pioneered the largest, independent, globally deployed automation platform which powers the enterprise, application and infrastructure. Now, as the consumerization of IT accelerates, Automic is re-imagining how organizations integrate next generation service models such as Cloud, DevOps and Big Data. Today, our software automates tens of millions of operations a day for over 2,000 customers worldwide. We challenge conventional thinking, enabling our customers to be faster, smarter, in control. Automic – the standard in business automation.

Find out more at our website [www.automic.com](http://www.automic.com).

### Download Center

The Automic Download Center ([downloads.automic.com](https://downloads.automic.com)) is the place where you find product downloads, documentation, release notes and information on new releases and hot-fixes about your Automic solution.

### Marketplace

The [Automic Marketplace \(https://marketplace.automic.com/\)](https://marketplace.automic.com/) features hundreds of business automation templates and solutions for enterprise automation needs including workload automation, service orchestration, DevOps initiatives, big data operations and social media outreach. DevOps engineers can also access a broad range of open-source containers to facilitate rapid application release automation. Users are able to provide ratings, review and feedback on existing plugins, get support and request new plugins.

### Automic Community

Join the Automic Community ([community.automic.com](https://community.automic.com)) to talk with other users from around the world to learn how they optimize their business automation with Automic. Interact with the Automic Team to get ONE Automation tips and tricks straight from the source.

---

## Technical Support

Our Technical Support Team ([support@automic.com](mailto:support@automic.com)) is ready to support you, anytime and anywhere. Three support centers located in Europe, the United States, and Asia Pacific build the core of the Automic support organization.

## Training Services

We offer a range of training options on how to get the most out of your Automic solution. Depending on your location, either open training sessions at an Automic Software Service Center, or personalized training sessions at your company's site suits best. Visit the training site and get detailed information about currently offered courses: <http://automic.com/about/training/>

Global Headquarter: Automic Software GmbH, Am Europlatz 5, 1120 Vienna, Austria

## 7.2 Automic Community

*Want to connect with other Automic users to compare notes or learn how others are tackling problems that you're running into?*

Talk with other users from around the world to learn how they optimize their business automation with Automic. Interact with the Automic Team to get ONE Automation tips and tricks straight from the source.

Join the Automic Community ([community.automic.com](http://community.automic.com)) and become an Automic Insider and be among the first to get news about new products and events, even before they are generally announced!

## 7.3 Download Center

*Make sure that you are using our products to their fullest potential.*

The Automic Download Center ([downloads.automic.com](http://downloads.automic.com)) is the place where you find product downloads, documentation and information on new releases and hot-fixes about your Automic solution. It's all in one place: from service hotfixes, release notes, and all guides. You will also find patch descriptions, known bugs or workarounds.

## 7.4 Marketplace

*Access largest marketplace of templates and solutions for automating any part of your enterprise!*

The [Automic Marketplace \(https://marketplace.automic.com/\)](https://marketplace.automic.com/) features hundreds of business automation templates and solutions for enterprise automation needs including workload automation, service orchestration, DevOps initiatives, big data operations and social media outreach. DevOps engineers can also access a broad range of open-source containers to facilitate rapid application release automation. Users are able to provide ratings, review and feedback on existing plugins, get support and request new plugins.

---

## 7.5 Technical Support

*We have a support team you can trust.*

Our team of professionals is ready to support you, anytime and anywhere. Three support centers located in Europe, the United States, and Asia Pacific build the core of the Automic support organization.

Our Technical Support Team ([support@automic.com](mailto:support@automic.com)) makes sure that your closest Automic experts are never more than a few hours flight away, no matter on which continent your subsidiaries and data centers are located. Automic software is designed to provide global connectivity for international companies. You are employing Automic software on a global scale and therefore you can expect global service.

## 7.6 Training Services

*Do you want to learn even more about Automic solutions?*

We offer a range of training options on how to get the most out of your Automic solution. Depending on your location, either open training sessions at an Automic Software Service Center, or personalized training sessions at your company's site suits best. Visit the training site and get detailed information about currently offered courses.

See also: <http://automic.com/about/training/>

---