



# Database and Middleware Automation

Software Version: 10.60.000.000

Linux, Solaris, AIX, and HP-UX

## Workflows for Red Hat JBoss

Document Release Date: July 2018

Software Release Date: August 2018



**Hewlett Packard**  
Enterprise

## Legal Notices

### Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

### Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### Copyright Notice

© 2012-2015 Hewlett Packard Enterprise Development LP

### Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

## Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: <https://softwaresupport.hpe.com/>.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

## Support

Visit the HPE Software Support site at: <https://softwaresupport.hpe.com/>.

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: <https://softwaresupport.hpe.com/web/softwaresupport/access-levels>.

**HPE Software Solutions Now** accesses the HPSW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is <https://softwaresupport.hpe.com/km/KM01702731>.

## About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

# Contents

Red Hat JBoss .....	5
Provision Open Source JBoss 7 StandAlone Mode .....	6
Prerequisites for this Workflow .....	7
How this Workflow Works .....	8
How to Run this Workflow .....	11
Sample Scenario .....	13
Parameters for Provision Open Source JBoss 7 StandAlone Mode .....	14
JBoss - Create and Configure Data Source v2 .....	15
Prerequisites for this Workflow .....	15
How this Workflow Works .....	15
How to Run this Workflow .....	18
Sample Scenarios .....	20
Parameters for JBoss - Create and Configure Data Source v2 .....	23
JBoss - Code Release v2 .....	24
Prerequisites for this Workflow .....	25
How this Workflow Works .....	25
How to Run this Workflow .....	30
Sample Scenario .....	33
Parameters for JBoss - Code Release v2 .....	35
JBoss - Provision Software v3 .....	37
Prerequisites for this Workflow .....	38
How this Workflow Works .....	39
How to Run this Workflow .....	42
Sample Scenarios .....	44
Parameters for JBoss - Provision Software v3 .....	46
JBoss - Patch Software v3 .....	48
Prerequisites for this Workflow .....	49
How this Workflow Works .....	50
How to Run this Workflow .....	52
Sample Scenario .....	54
Parameters for JBoss - Patch Software v3 .....	55
JBoss - Rollback Patch Software v2 .....	56
Prerequisites for this Workflow .....	57
How this Workflow Works .....	58
How to Run this Workflow .....	60
Sample Scenario .....	62
Parameters for JBoss - Rollback Patch Software v2 .....	63

Send documentation feedback .....	64
-----------------------------------	----

# Red Hat JBoss

This section contains the following topics:

Workflow type	Workflow name
Provisioning	<a href="#">"Provision Open Source JBoss 7 StandAlone Mode" on the next page</a>
	<a href="#">"JBoss - Provision Software v3" on page 37</a>
Patching	<a href="#">"JBoss - Patch Software v3" on page 48</a>
	<a href="#">"JBoss - Rollback Patch Software v2" on page 56</a>
Configuring	<a href="#">"JBoss - Create and Configure Data Source v2" on page 15</a>
Release Management	<a href="#">"JBoss - Code Release v2" on page 24</a>

# Provision Open Source JBoss 7 StandAlone Mode

Use this workflow to install the open source JBoss Application Server 7 Community version (JBoss AS 7) and start a single, default profile application server.

The workflow performs checks to determine whether the JBoss and Java binaries exist on the target server. If they do not, the workflow downloads them from the software repository.

The workflow also performs validation checks at the operating system level, including file system space checks and Java version level checks.

To use this workflow in your environment, see the following information:

Topic	Information Included
<a href="#">"Prerequisites for this Workflow"</a>	List of prerequisites that must be satisfied before you can run this workflow
<a href="#">"How this Workflow Works"</a>	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
<a href="#">"How to Run this Workflow"</a>	Instructions for running this workflow in your environment
<a href="#">"Sample Scenario"</a>	Examples of typical parameter values for this workflow
<a href="#">"Parameters for Provision Open Source JBoss 7 StandAlone Mode"</a>	List of input parameters for this workflow

The process of deploying and running this workflow is the same for all scenarios, but the parameters required will differ depending on the specific scenario that you are implementing.

The workflow provides default values for most parameters. These default values are usually sufficient for a typical provisioning scenario. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios.

Any parameters not explicitly specified in the deployment will have the default values listed in ["Parameters for Provision Open Source JBoss 7 StandAlone Mode" on page 14](#).

**Note:** For information about the steps in this workflow, see the [Steps in this Workflow](#).

## Prerequisites for this Workflow

The following prerequisites must be satisfied before you can run the "[Provision Open Source JBoss 7 StandAlone Mode](#)" workflow:

1. The workflow requires unchallenged `sudo` access to a user (typically `root`) who can access all required files and directories.
2. The workflow requires the Java Development Kit (JDK) version 1.6 update 24.
3. Adequate disk space must be available to install the JBoss and Java binaries.

For information about prerequisites for JBoss AS 7, refer to the [JBoss Product Documentation](#).

## How this Workflow Works

This topic contains the following information about the "[Provision Open Source JBoss 7 StandAlone Mode](#)" workflow:

### Validation Checks Performed

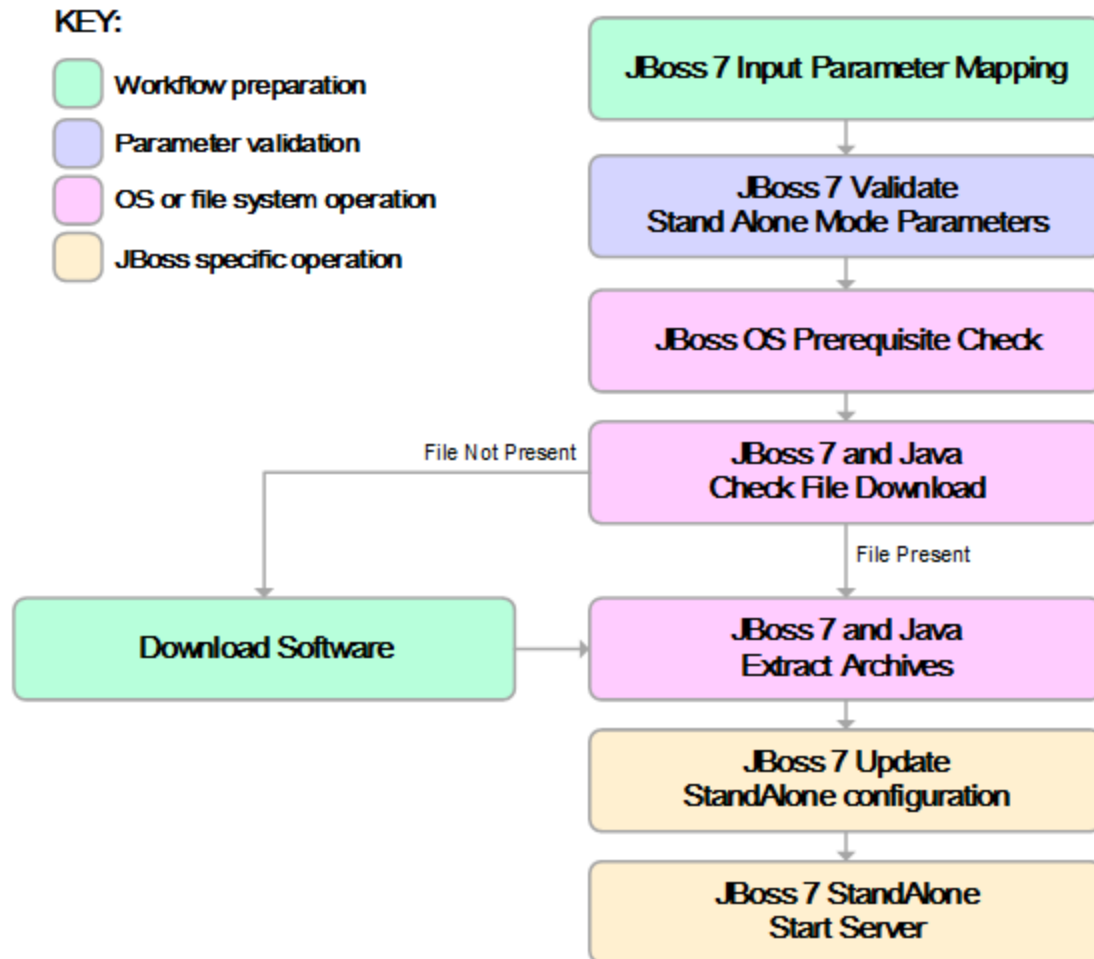
The workflow checks the following things prior to extracting the binaries. If any of these checks fails, the workflow fails.

1. All required parameters have values. If any required parameter does not have a value—either a value that you specify or a default value—the workflow fails.
2. All required libraries are present (see "[Prerequisites for this Workflow](#)" on the previous page).
3. Sufficient disk space is available to extract the binary files from the compressed archive.
4. Sufficient disk space is available to install JBoss and Java.



## Steps Executed

The "Provision Open Source JBoss 7 StandAlone Mode" workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure, and all subsequent steps are skipped.



## Process Flow

This workflow performs the following tasks:

1. Creates the call wrapper and determines the target server platform type.
2. Validates the parameters needed to install JBoss and Java and create a stand-alone profile (see the [validation checks](#) performed).
3. Checks the following:
  - a. File system space requirements where JBoss and Java will be installed.
  - b. Temporary space requirements where the compressed software will be extracted before it is installed.
4. Determines whether the JBoss and Java binary archives are present on the target server. If either archive is not present, the workflow downloads it from the software repository.
5. Extracts the JBoss and Java binary archives to the specified directories.
6. Creates a default profile for a stand-alone application server.
7. Starts the new stand-alone JBoss application server.
8. Cleans up any files that were downloaded.

# How to Run this Workflow

This topic explains how to customize and run the ["Provision Open Source JBoss 7 StandAlone Mode"](#) workflow in your environment.

**Note:** Prior to running this workflow, review the ["Prerequisites for this Workflow"](#), and ensure that all requirements are satisfied.

**To customize and run the Provision Open Source JBoss StandAlone Mode workflow:**

1. Create a deployable copy of the workflow
2. Determine the values that you will specify for the following parameter. These are the parameters that are visible in the deployment by default.

Parameter Name	Default Value	Description
Install Dir	UNIX: /opt/jboss  Windows: c:\jboss	Fully qualified path where the JBoss and Java binaries will be uncompressed..
JBoss Binary Archive	no default	Fully qualified path where the compressed Java software package should be found on the target server. If the Java software package is not available in this location, it will be downloaded from the SASoftware repository and placed in this location.
Java Binary Archive	no default	Fully qualified path where the compressed Java software package should be found on the target server. If the Java software package is not available in this location, it will be downloaded from the SASoftware repository and placed in this location.
JBoss User	root	The user who will install and run JBoss. This user must have write permission on the install directory.

See ["Parameters for Provision Open Source JBoss 7 StandAlone Mode" on page 14](#) for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.

6. On the Parameters tab, specify values for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment.

## Sample Scenario

This topic shows you how to use various parameters to achieve the following provisioning scenario in your environment using the ["Provision Open Source JBoss 7 StandAlone Mode"](#) workflow:

### Install JBoss Application Server 7 Community version

Specify values for the following parameters to install JBoss AS 7 and start a single, default profile application server. This is the simplest scenario, and it uses only those parameters that are visible in the deployment by default (out of the box).

Step Name	Parameter Name	Example Value
JBoss 7: Validate Stand Alone Mode Parameters	Install Dir	/opt/jboss/jboss-as7
	JBoss Binary Archive	/opt/jboss/jboss-as-7.1.1.Final.zip
	Java Binary Archive	/opt/jboss/jdk-6u29-linux-x64.bin
	JBoss User	root

Be sure that the default values for all remaining parameters are appropriate for your environment (see ["Parameters for Provision Open Source JBoss 7 StandAlone Mode"](#) on the next page).

# Parameters for Provision Open Source JBoss 7

## StandAlone Mode

The following tables describe the required and optional input parameters for this workflow. Most of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned

### Parameters Defined in this Step: JBoss 7: Validate Stand Alone Mode Parameters

Parameter Name	Default Value	Required	Description
Call Wrapper	see description	optional	Command that will execute this step (or subsequent steps) as a specific user.  For UNIX targets, the default is:  <code>/opt/hp/dma/client/jython.sh</code> running as root.  For Windows targets, the default is: <code>jython</code> running as Administrator.
File List	no default	optional	Comma-separated list of fully qualified files ( JBoss Binary Archive, Java Binary Archive) that must either exist on the target server or be downloaded from the software repository.
HostName	no default	required	Fully qualified hostname or IP address of the server where JBoss will be installed.
Install Dir	UNIX: <code>/opt/jboss</code>  Windows: <code>c:\jboss</code>	optional	Fully qualified path where the JBoss and Java binaries will be uncompressed..
JBoss Binary Archive	no default	required	Fully qualified path where the compressed Java software package should be found on the target server. If the Java software package is not available in this location, it will be downloaded from the SAsoftware repository and placed in this location.
JBoss Home	no default	optional	Fully qualified path from which JBoss will run.
JBoss User	root	optional	The user who will install and run JBoss. This user must have write permission on the install directory.
Java Binary Archive	no default	optional	Fully qualified path where the compressed Java software package should be found on the target server. If the Java software package is not available in this location, it will be downloaded from the SAsoftware repository and placed in this location.
Java Home	no default	optional	Fully qualified path from which Java will run.

# JBoss - Create and Configure Data Source v2

This workflow creates a data source for a given JBoss Application Server. The data source can be used later by applications deployed to the configured database. The workflow can create a data source for databases on the same machine as well as on remote machines.

To use this workflow in your environment, see the following information:

Topic	Information Included
<a href="#">"Prerequisites for this Workflow"</a>	List of prerequisites that must be satisfied before you can run this workflow
<a href="#">"How this Workflow Works"</a>	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
<a href="#">"How to Run this Workflow"</a>	Instructions for running this workflow in your environment
<a href="#">"Sample Scenarios"</a>	Examples of typical parameter values for this workflow
<a href="#">Parameters</a>	List of input parameters for this workflow

## Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the JBoss - Create and Configure Data Source v2 workflow.

### Product Platform

This workflow creates a new JBoss Data Source connection for JBoss EAP 6.x and JBoss WildFly in standalone mode.

### Dependencies

The JBoss Application server must be provisioned, up, and running. The database to which the connection is being created must already be installed.

## How this Workflow Works

The following information describes how the JBoss - Create and Configure Data Source v2 workflow works:

## Overview

This workflow does the following things in the order shown:

1. Initially, the workflow inputs all parameters, set defaults for optional parameters, and validates all parameters.
2. Next the workflow creates and configures the JDBC provider and data source on JBoss Application Server.
3. Finally, the workflow verifies that the connection to the data source was successful.

## Validation Checks Performed

The workflow performs the following checks on the input parameters:

Implementation Type	Must be xa or non-xa
JBoss Home Data Source Name Driver Name Connection URL Driver Class Path Implementation Type JNDI Name	Must be specified if Implementation Type is set to non-xa
XA Datasource Class Name XA DataSource Database Name XA DataSource Port XA DataSource Server Name JNDI Name Connection URL JBoss Home Implementation Type Data Source Name	Must be specified if Implementation Type is set to xa
JNDI Name	Must start with java:/ or java:jboss/

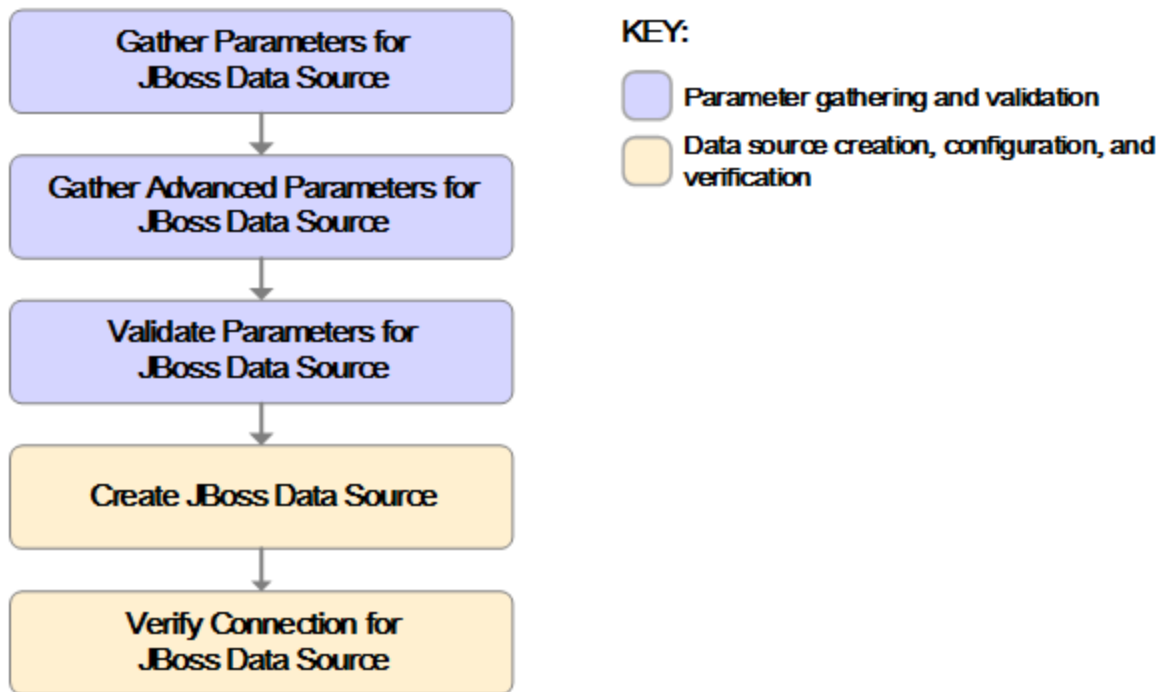
The JBoss - Create and Configure Data Source v2 workflow also checks the environment for the following:

- The operating system is supported.
- The JBoss version is EAP 6.x or later.
- The JBoss installation location is valid.
- The driver class path must exist.

## Steps Executed

The JBoss - Create and Configure Data Source v2 workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.





#### Steps Used in the JBoss - Create and Configure Data Source v2 Workflow

Workflow Step	Description
Gather Parameters for JBoss Data Source	This step gathers mandatory input parameters from the user to create a data source on JBoss Application Server standalone setup.
Gather Advanced Parameters for JBoss Data Source	This step prepares and validates the parameters needed to configure a JDBC provider, J2C alias, and data source for a WebSphere Application Server. This step gathers the advanced input parameters to create a data source on a JBoss Application Server. Input parameters specified in this step are optional.
Validate Parameters for JBoss Data Source v2	This step validates the basic and advanced parameters provided by the user, and checks the prerequisites for creating a data source on a JBoss Application Server on standalone setup.
Create JBoss Data Source v3	This step creates and configures the JDBC provider and data source on JBoss Application Server.
Verify Connection for JBoss Data Source v2	This step verifies the connection created by the workflow. If the test connection fails, then the workflow fails as well.

For parameter descriptions and defaults, see ["Parameters for JBoss - Create and Configure Data Source v2" on page 23](#).

## How to Run this Workflow

The following instructions show you how to customize and run the JBoss - Create and Configure Data Source v2 workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios. Any parameters not explicitly specified in the deployment will have the default values listed in ["Parameters for JBoss - Create and Configure Data Source v2" on page 23](#).

**Note:** Before following this procedure, review the ["Prerequisites for this Workflow" on page 15](#), and ensure that all requirements are satisfied.

**To use the JBoss - Create and Configure Data Source v2 workflow:**

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

### Input Parameters for Gather Parameters for JBoss Create and Configure Data Source

Parameter Name	Default Value	Required	Description
Connection URL	no default	required	The URL used by the data source to connect to the database.  For example: jdbc:oracle:thin:@//localhost:1521
Data Source Name	no default	required	The name given to the data source when it is created.
Driver Class Path	no default	required	The JAR file name for the resource provider classes.  For example: /app/oracle/jdbc/ojdbc6.jar
Driver Name	no default	required	The full path of the driver name.  For example: oracle.jdbc.OracleDriver
Driver Type	no default	required	The name of the driver type. For example: 'oracle', 'MS-SQL', etc.
Implementation Type	no default	required	The implementation type for the JDBC (Java Database Connectivity) provider. Use Connection pool data source if your application runs in a single phase or a local transaction. Otherwise, use XA data source to run in a global transaction. Valid

**Input Parameters for Gather Parameters for JBoss Create and Configure Data Source, continued**

Parameter Name	Default Value	Required	Description
			values are Connection pool data source or XA data source.
JBoss Home	no default	required	The JBoss installation location.
JNDI Name	no default	required	Java Name Directory Interface (JNDI) name. This is a user specified string specific to the application component calls to the data source. For example: jdbc/myDatasource
Password	no default	required	The database password.
User Name	no default	required	The user name of the database.

**Input Parameters for Gather Advanced Parameters for JBoss Create and Configure Data Source**

Parameter Name	Default Value	Required	Description
XA DataSource Database Name	no default	optional	If XA is the Implementation Type, then provide the database name.
XA DataSource Port	no default	optional	If XA is the Implementation Type, then provide the port number.
XA DataSource Server Name	no default	optional	If XA Datasource is provided, then add the fully-qualified Server Name.
XA Datasource Class Name	no default	optional	If XA is the Implementation Type, then provide the Datasource Class.  For example:  'com.mysql.jdbc.jdbc2.optional.MysqlXADataSource'

**Tip:** To avoid having to re-enter passwords whenever they change, you can create a policy to provide them to the workflow.

**Note:** This is the minimum set of parameters required to run this workflow. You may need to expose additional parameters depending on your objectives.

See ["Parameters for JBoss - Create and Configure Data Source v2" on page 23](#) for detailed descriptions of all input parameters for this workflow, including default values.

2. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment.
3. Save the changes to the workflow (click **Save** in the lower right corner).
4. Create a new deployment.
5. On the Parameters tab, specify values for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
6. On the Targets tab, specify one or more targets for this deployment.
7. Save the deployment (click **Save** in the lower right corner).
8. Run the workflow using this deployment.

## Sample Scenarios

This topic shows you typical parameter values for different use cases for the JBoss - Create and Configure Data Source v2 workflow. For a complete list of all parameters used in this workflow, including default values, see ["Parameters for JBoss - Create and Configure Data Source v2" on page 23](#).

The sample scenarios assume that Web Service URL has the value of DMA.URL. This is the default value mapped from the DMA metadata.

**Tip:** To avoid having to re-enter passwords whenever they change, you can create a policy to provide them to the workflow.

### Scenario 1: Create an Oracle data source using a connection pool data source (non-XA)

In this scenario we will create an Oracle data source using a connection pool data source, where both JBoss and Oracle are installed on same machine.

#### Input Parameters for Gather Parameters for JBoss Data Source

Parameter Name	Example Value	Description
Connection URL	jdbc:oracle:thin:@localhost:1521/orcl	The URL used by the data source to connect to the database.  For example: jdbc:oracle:thin:@//localhost:1521
Data Source Name	myOraclePool	The name given to the data source when it is created.

**Input Parameters for Gather Parameters for JBoss Data Source, continued**

Parameter Name	Example Value	Description
Driver Class Path	/tmp/jar/ojdbc6.jar	The JAR file name for the resource provider classes.  For example: /app/oracle/jdbc/ojdbc6.jar
Driver Name	oracle.jdbc.OracleDriver	The full path of the driver name.  For example: oracle.jdbc.OracleDriver
Driver Type	Oracle	The name of the driver type. For example: 'oracle', 'MS-SQL', etc.
Implementation Type	non-xa	The implementation type for the JDBC (Java Database Connectivity) provider. Use Connection pool data source if your application runs in a single phase or a local transaction. Otherwise, use XA data source to run in a global transaction. Valid values are Connection pool data source or XA data source.
JBoss Home	/opt/jboss/wildfly-8.1.0/	The JBoss installation location.
JNDI Name	java:/jboss/MyOracleCpool	Java Name Directory Interface (JNDI) name. This is a user specified string specific to the application component calls to the data source. For example: jdbc/myDataSource
Password	Test	The database password.
User Name	User_test	The user name of the database.

**Scenario 2: Create an Oracle data source using a connection pool data source (XA)**

In this scenario we will create an Oracle data source using a connection pool data source, where Oracle is installed on a remote machine.

**Input Parameters for Gather Parameters for JBoss Data Source**

Parameter Name	Example Value	Description
Connection URL	jdbc:oracle:thin:@remoteHost.xyz.com:1521/orcl	The URL used by the data source to connect to the database.  For example: jdbc:oracle:thin:@//localhost:1521
Data Source Name	myOraclePool	The name given to the data source when it is created.
Driver Class Path	/tmp/jar/ojdbc6.jar	The JAR file name for the resource provider classes.

**Input Parameters for Gather Parameters for JBoss Data Source, continued**

Parameter Name	Example Value	Description
		For example: /app/oracle/jdbc/ojdbc6.jar
Driver Name	oracle.jdbc.OracleDriver	The full path of the driver name.  For example: oracle.jdbc.OracleDriver
Driver Type	Oracle	The name of the driver type. For example: 'oracle', 'MS-SQL', etc.
Implementation Type	xa	The implementation type for the JDBC (Java Database Connectivity) provider. Use Connection pool data source if your application runs in a single phase or a local transaction. Otherwise, use XA data source to run in a global transaction. Valid values are Connection pool data source or XA data source.
JBoss Home	/opt/jboss/wildfly-8.1.0/	The JBoss installation location.
JNDI Name	java:/jboss/MyOracleCpool	Java Name Directory Interface (JNDI) name. This is a user specified string specific to the application component calls to the data source. For example: jdbc/myDatasource
Password	Test	The database password.
User Name	User_test	The user name of the database.

**Input Parameters for Gather Advanced Parameters for JBoss Data Source**

Parameter Name	Example Value	Description
XA DataSource Database Name	remoteHost.xyz.com	If XA is the Implementation Type, then provide the database name.
XA DataSource Port	Orcl	If XA is the Implementation Type, then provide the port number.
XA DataSource Server Name	1521	If XA Datasource is provided, then add the fully-qualified Server Name.
XA Datasource Class Name	oracle.jdbc.xa.client.OracleXADataSource	If XA is the Implementation Type, then provide the Datasource Class.  For example:  'com.mysql.jdbc.jdbc2.optional.MysqlXADataSource'

# Parameters for JBoss - Create and Configure Data Source v2

The following tables describe the required and optional input parameters for this workflow. Most of these parameters are not initially visible in a deployment. For most parameters, if you do not specify a value for a parameter, a default value is assigned.

## Input Parameters for Gather Parameters for JBoss Data Source

Parameter Name	Default Value	Required	Description
Connection URL	no default	required	The URL used by the data source to connect to the database.  For example: <code>jdbc:oracle:thin:@//localhost:1521</code>
Data Source Name	no default	required	The name given to the data source when it is created.
Driver Class Path	no default	required	The JAR file name for the resource provider classes.  For example: <code>/app/oracle/jdbc/ojdbc6.jar</code>
Driver Name	no default	required	The full path of the driver name.  For example: <code>oracle.jdbc.OracleDriver</code>
Driver Type	no default	required	The name of the driver type. For example: 'oracle', 'MS-SQL', etc.
Implementation Type	no default	required	The implementation type for the JDBC (Java Database Connectivity) provider. Use Connection pool data source if your application runs in a single phase or a local transaction. Otherwise, use XA data source to run in a global transaction. Valid values are Connection pool data source or XA data source.
JBoss Home	no default	required	The JBoss installation location.
JNDI Name	no default	required	Java Name Directory Interface (JNDI) name. This is a user specified string specific to the application component calls to the data source. For example: <code>jdbc/myDatasource</code>
Password	no default	required	The database password.
User Name	no default	required	The user name of the database.

**Input Parameters for Gather Advanced Parameters for JBoss Data Source**

Parameter Name	Default Value	Required	Description
XA DataSource Database Name	no default	optional	If XA is the Implementation Type, then provide the database name.
XA DataSource Port	no default	optional	If XA is the Implementation Type, then provide the port number.
XA DataSource Server Name	no default	optional	If XA Datasource is provided, then add the fully-qualified Server Name.
XA Datasource Class Name	no default	optional	<p>If XA is the Implementation Type, then provide the Datasource Class.</p> <p>For example:</p> <p>'com.mysql.jdbc.jdbc2.optional.MysqlXADataSource'</p>

**Tip:** To avoid having to re-enter passwords whenever they change, you can create a policy to provide them to the workflow.

**Note:** This is the minimum set of parameters required to run this workflow. You may need to expose additional parameters depending on your objectives.

## JBoss - Code Release v2

This workflow automates the deployment of applications in a JBoss Application Server. In addition to deployment, this workflow can update the JVM Generic Arguments and JVM System Properties on the Web Server, and also provides install options for the deployment of applications.

Some of the install options are provided as parameters to the workflow, or users can specify install options within a file for each of the applications to be deployed. Note, though, that the value provided for parameters takes higher precedence. This workflow supports the verification of the application deployments by providing the URLs.

For successful application deployments, verifications and a list of the applications are maintained in the History file. In cases of unsuccessful application deployments, the workflow rolls back the deployment and restores the last successfully deployed application (if any).



To use this workflow in your environment, see the following information:

Topic	Information Included
<a href="#">"Prerequisites for this Workflow"</a>	List of prerequisites that must be satisfied before you can run this workflow
<a href="#">"How this Workflow Works"</a>	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
<a href="#">"How to Run this Workflow"</a>	Instructions for running this workflow in your environment
<a href="#">"Sample Scenario"</a>	Examples of typical parameter values for this workflow
<a href="#">Parameters</a>	List of input parameters for this workflow

## Prerequisites for this Workflow

Be sure that the following prerequisites are satisfied before you run the JBoss - Code Release v2 workflow.

### Product Platform

This workflow deploys an application archive on the Red Hat Enterprise Linux platform only .

### Dependencies

This workflow requires the JBoss Application Server to be installed beforehand.

For more information about prerequisites for JBoss - Code Release, refer to the [JBoss Product Documentation](#).

## How this Workflow Works

The following information describes how the JBoss - Code Release v2 workflow works:

## Overview

This workflow does the following things in the order shown:

1. Initially, the workflow inputs all parameters, set defaults for optional parameters, and validates all parameters. If input files do not exist in the specified locations, they are downloaded from the software repository. The workflow performs a checksum to verify that the archive files should be deployed in the Application Server.
2. Next, the workflow creates the installation options. The workflow updates the JVM settings (if any) and then takes a configuration backup. The workflow deploys the specified Application Archive files in the Application Server.
3. If the application deployment succeeds, the workflow tests the URLs for the web servers and copies the application archives.
4. If the application deployment fails, the workflow rolls back the deployment and restores the last successfully deployed application (if any).
5. Finally, the workflow cleans up downloaded files based on the Cleanup on Success and Cleanup on Failure parameters.

**Validation Checks Performed**

The workflow performs the following checks on the input parameters:

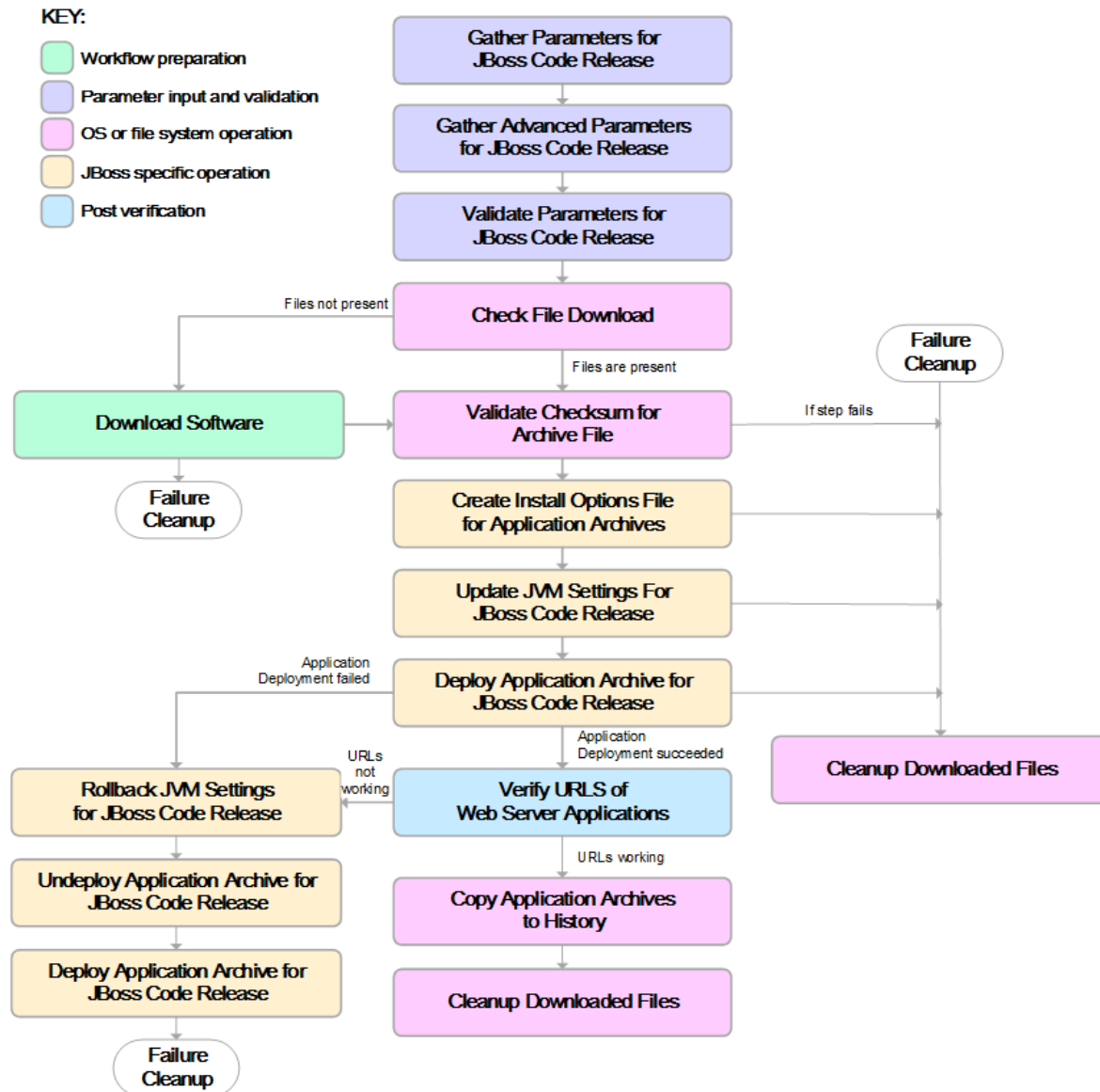
JBoss Home Application Archive File List MD5 Checksum JBoss Code Release History Location JBoss Staging Location	Required parameters must have values specified
Archive Install Option Force Deploy	If set to True (Yes, Y, or T), Archive Install Option All Server Groups and Archive Install Option Server Groups are not specified
Application Archive File List Md5 Checksum	There must be a checksum for each Application Archive file The Application Archive files must be type .ear or .war and have valid absolute paths Checksums must be valid hexadecimal numbers
JBoss Staging Location JBoss Code Release History Location	Must be valid absolute paths Cannot have the same values

The JBoss - Code Release v2 workflow also checks the environment for the following:

- The operating system is supported.
- The JBoss version is EAP 6.x or later.
- The JBoss installation location is valid.

## Steps Executed

The JBoss - Code Release v2 workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and subsequent steps are skipped, except for the Cleanup Downloaded Files step.



**Steps Used in the JBoss- Code Release v2 Workflow**

<b>Workflow Step</b>	<b>Description</b>
Gather Parameters for JBoss Code Release	This step gathers mandatory input parameters (user-provided) used to deploy a list of application archives in a JBoss Application Server.
Gather Advanced Parameters for JBoss Code Release	This step gathers the advanced input parameters (user-provided) used to deploy an application archive for a JBoss Application Server. Input parameters specified in this step are optional.
Validate Parameters for JBoss Code Release	This step validates the basic and advanced parameters provided by the user, and checks the prerequisites for deploying a list of application archives for a JBoss Application Server.
Check File Download	<p>This step checks for the existence of a file before downloading from the Server Automation software repository.</p> <ul style="list-style-type: none"> <li>• Checks if file is in the expected location.</li> <li>• If the file is not in the expected location, generates a list of files for file download.</li> </ul>
Download Software	This step downloads a list of files to a specified location on the target server.
Validate Checksum for Archive File	This step verifies the checksum for the archive files and archive setting file (if any) to ensure that the file has not changed and that the correct archives are deployed in the Application Server.
Create Install Options File for Application Archives	This step creates a setting file that includes the install options for the list of application archive files being deployed by the application server.
Update JVM Settings For JBoss Code Release v2	This step updates the JVM settings for the JBoss Application Server, and also performs a backup of the JBoss server configuration.
Deploy Application Archive for JBoss Code Release	Using the user-provided Application Archive files: This step deploys the list of application archives (.war and .ear) in a JBoss Application Server.
<b>If the application deployment succeeds, the following steps are executed</b>	
Verify URLs of Web Server Applications	This step verifies the checksum for the archive files and archive setting file (if any) to ensure that the file has not changed and that the correct archives are deployed in the Application Server.
Copy Application Archives to History	This step creates a setting file that includes the install options for the list of application archive files being deployed by the application server.
Cleanup Downloaded Files	Using the user-provided Application Archive files: This step deploys the list of application archives (.war and .ear) in a JBoss Application Server.
<b>If the application deployment fails, the following steps are executed</b>	
Rollback JVM Settings for JBoss Code Release v2	This step restores the backup of a JBoss Application server configuration.
Undeploy Application Archive for JBoss	This step uninstalls the list of application archives from a JBoss Application Server.

**Steps Used in the JBoss- Code Release v2 Workflow, continued**

Workflow Step	Description
Code Release v2	
Deploy Application Archive for JBoss Code Release	Using the backup of the Application Archive files: This step deploys the list of application archives (.war and .ear) in a JBoss Application Server.
Cleanup Downloaded Files	For workflow failure—and if Cleanup on Failure is set to True (default)—this step removes all downloaded files and archives. Dependencies: Run as file/directory owner.

For parameter descriptions and defaults see, ["Parameters for JBoss - Code Release v2" on page 35](#).

## How to Run this Workflow

The following instructions show you how to customize and run the JBoss - Code Release v2 workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment.

**Note:** Before following this procedure, review the ["Prerequisites for this Workflow"](#), and ensure that all requirements are satisfied.

Before you run this workflow, you can perform the following optional advance configuration to deploy applications JBoss application servers.

Create a configuration file on the target machine or the SA Server. The file should contain the advanced parameters for all the application servers being deployed. If no configuration file is provided, the target will be defaulted to admin server of the domain. The options that are to be used in this file are listed below.

```
AdderEAR.ear = {
  Runtime Name = adder_app
}
myServletWAR.war = {
  Force Deploy=Yes Runtime Name=myservletAPP Keep Content=True
}
```

List of Options:

- Force Deploy
- All Server Groups
- Runtime Name
- Server Groups
- Disabled
- Keep Content

**To use the JBoss - Code Release workflow:**

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters: show

**Parameters Defined in this Step: Gather Parameters for JBoss - Code Release**

Parameter Name	Default Value	Required	Description
Application Archive File List	no default	required	Comma-separated list of the Application Archive files to be deployed. Example: xxx.war or yyy.ear
JBoss Code Release History Location	/opt/hp/dma/jboss/history	required	Fully qualified path name of the location where the application archive will be saved (for history purposes) on the target machine. This location cannot be the same as the Code Release Staging Location.
JBoss Home	/opt/jboss	required	The location of the JBoss installation.
JBoss Staging Location	/tmp/jboss/stage	required	The temporary location in which to store the application archive.  Note that the workflow will fail if the directory does not exist in the location specified.
MD5 Checksum	no default	required	Comma-separated list of the MD5 Checksum of the Application Archive files to be deployed.

**Parameters Defined in this Step: Gather Parameters for JBoss - Code Release , continued**

Parameter Name	Default Value	Required	Description
Test URLs	no default	required	Comma-separated list of URLs used to test whether or not the list of applications deployed successfully.  Example: http://mytestdb.com, http://yourtest.com

**Note:** This is the minimum set of parameters required to run this workflow. You may need to expose additional parameters depending on your objectives.

See ["Parameters for JBoss - Code Release v2" on page 35](#) for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment.

**To verify the results:**

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

*Optional:* If you want to further verify the results:

Use the WebSphere administrative console interface to check that the web server is configured.



## Sample Scenario

This topic shows you typical parameter values for different use cases for the JBoss - Code Release v2 workflow. For a complete list of all parameters used in this workflow, including default values, see ["Parameters for JBoss - Code Release v2" on page 35](#).

**Tip:** To avoid having to re-enter passwords whenever they change, you can create a policy to provide them to the workflow.

### Scenario 1: Install an application archive (for example stockanalysis.war) on a running JBoss Application Server on a standalone setup.

In this scenario we will deploy the stockanalysis.war file on a running JBoss Application Server. We will install the application using the default installation options. If the application archive files are not present in the Code Release Staging Location, then they will be downloaded from the SA Repository.

#### Parameters Defined in this Step: Gather Parameters for JBoss - Code Release

Parameter Name	Example Value	Description
Application Archive File List	stockanalysis.war	Comma-separated list of the Application Archive files to be deployed.  Example: xxx.war or yyy.ear
JBoss Code Release History Location	/opt/hp/dma/jboss/history	Fully qualified path name of the location where the application archive will be saved (for history purposes) on the target machine. This location cannot be the same as the Code Release Staging Location.
JBoss Staging Location	/tmp/jboss/stage	The temporary location in which to store the application archive. Note that the workflow will fail if the directory does not exist in the location specified.
JBoss Home	/opt/jboss/wildfly-9.0.0.Alpha1/	The location of the JBoss installation.
MD5 Checksum	4477e994e9d457ad2214a3d36b1bb16b	Comma-separated list of the MD5 Checksum of the Application Archive files to be deployed.
Test URLs	http://<server:port>/stockanalysis/<xyx.html>	Comma-separated list of URLs used to test whether or not the list of applications deployed successfully.  Example: <a href="http://mytestdb.com">http://mytestdb.com</a> , <a href="http://yourtest.com">http://yourtest.com</a>

## Scenario 2: Install an application archive (for example stockanalysis.war) on a running JBoss Application Server on a standalone setup.

In this scenario we will deploy the stockanalysis.war file on a running JBoss Application Server. We will install the application using the default installation options. The JVM settings are also applied to the Application server. If the application archive files are not present in the Code Release Staging Location, then they will be downloaded from the SA Repository.

### Parameters Defined in this Step: Gather Parameters for JBoss - Code Release

Parameter Name	Example Value	Description
Application Archive File List	stockanalysis.war	Comma-separated list of the Application Archive files to be deployed.  Example: xxx.war or yyy.ear
JBoss Code Release History Location	/opt/hp/dma/jboss/history	Fully qualified path name of the location where the application archive will be saved (for history purposes) on the target machine. This location cannot be the same as the Code Release Staging Location.
JBoss Staging Location	/opt/hp/dma/jboss/history	The temporary location in which to store the application archive. Note that the workflow will fail if the directory does not exist in the location specified.
JBoss Home	/opt/jboss/wildfly-9.0.0.Alpha1/	The location of the JBoss installation.
MD5 Checksum	4477e994e9d457ad2214a3d36b1bb16b	Comma-separated list of the MD5 Checksum of the Application Archive files to be deployed.
Test URLs	http://<server:port>/stockanalysis/<xyx.html>	Comma-separated list of URLs used to test whether or not the list of applications deployed successfully.  Example: <a href="http://mytestdb.com">http://mytestdb.com</a> , <a href="http://yourtest.com">http://yourtest.com</a>

### Parameters Defined in this Step: Gather Advanced Parameters for JBoss - Code Release

Parameter Name	Example Value	Description
JVM Generic Arguments	<ul style="list-style-type: none"> <li>Dclient.encoding.override=UTF-8</li> <li>Dsun.rmi.dgc.client.gcInterval=3600000000</li> <li>Dsun.rmi.dgc.server.gcInterval=3600000000</li> </ul>	Specifies the JVM generic arguments. Provide values as standard JVM settings.

**Parameters Defined in this Step: Gather Advanced Parameters for JBoss - Code Release , continued**

Parameter Name	Example Value	Description
JVM System Properties	stockanalysis_home, /opt/stockanalysis/bin, Home path for the stock analysis	Specifies the JVM System Properties. Provide the string in the following format: 'PropertyName, PropertyValue  PropertyName, PropertyValue'

## Parameters for JBoss - Code Release v2

The following tables describe the required and optional input parameters for this workflow. For most parameters, if you do not specify a value for a parameter, a default value is assigned.

Determine the values that you will specify for the following parameters: show

**Parameters Defined in this Step: Gather Parameters for JBoss - Code Release**

Parameter Name	Default Value	Required	Description
Application Archive File List	no default	required	Comma-separated list of the Application Archive files to be deployed.  Example: xxx.war or yyy.ear
JBoss Code Release History Location	/opt/hp/dma/jboss/history	required	Fully qualified path name of the location where the application archive will be saved (for history purposes) on the target machine. This location cannot be the same as the Code Release Staging Location.
JBoss Home	/opt/jboss	required	The location of the JBoss installation.
JBoss Staging Location	/tmp/jboss/stage	required	The temporary location in which to store the application archive. Note that the workflow will fail if the directory does not exist in the location specified.
MD5 Checksum	no default	required	Comma-separated list of the MD5 Checksum of the Application Archive files to be deployed.
Test URLs	no default	required	Comma-separated list of URLs used to test whether or not the list of applications deployed successfully.  Example: <a href="http://mytestdb.com">http://mytestdb.com</a> , <a href="http://yourtest.com">http://yourtest.com</a>

**Parameters Defined in this Step: Gather Advanced Parameters for JBoss - Code Release**

Parameter Name	Default Value	Required	Description
Archive Install Option All Server Groups	no default	optional	This parameter specifies whether or not the deployment is applicable to all available server groups. <b>Note:</b> This argument is unavailable in standalone mode.
Archive Install Option Force Deploy	True	optional	This parameter specifies whether or not the existing application is replaced by the new one. A value of True replaces the application, and a value of False ensures that it is not replaced.
Archive Install Option Runtime Name	no default	optional	Specifies the runtime name of the deployment.
Archive Install Option Server Groups	no default	optional	This parameter specifies a comma-separated list of server group names to which the deployment should apply. <b>Note:</b> This argument is unavailable in standalone mode.
Archive Settings File	no default	optional	The file containing the install options for all the archive files.
Cleanup on Failure	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon failure of the workflow.
Cleanup on Success	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon successful execution of the workflow.
JVM Generic Arguments	no default	optional	Specifies the JVM generic arguments. Provide values as standard JVM settings.
JVM System Properties	no default	optional	Specifies the JVM System Properties. Provide the string in the following format: 'PropertyName, PropertyValue  PropertyName, PropertyValue'

## JBoss - Provision Software v3

Use this workflow to install a new instance of a standalone JBoss Application Server supporting community edition 7.1.1 and 7.2.0, Enterprise Application Platform (EAP), and WildFly versions.

The workflow performs checks to determine whether the JBoss and Java binaries exist on the target server. If they do not, the workflow downloads them from the software repository.

The workflow also performs validation checks at the operating system level, including file system space checks and Java version level checks.

Topic	Information Included
<a href="#">"Prerequisites for this Workflow"</a>	List of prerequisites that must be satisfied before you can run this workflow
<a href="#">"How this Workflow Works"</a>	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
<a href="#">"How to Run this Workflow"</a>	Instructions for running this workflow in your environment
<a href="#">"Sample Scenarios"</a>	Examples of typical parameter values for this workflow
<a href="#">"Parameters for JBoss - Provision Software v3"</a>	List of input parameters for this workflow

The process of deploying and running this workflow is the same for all scenarios, but the parameters required will differ depending on the specific scenario that you are implementing.

The workflow provides default values for most parameters. These default values are usually sufficient for a typical provisioning scenario. You can override the defaults by specifying parameter values in the deployment. You can also expose additional parameters in the workflow, if necessary, to accomplish more advanced scenarios.

Any parameters not explicitly specified in the deployment will have the default values listed in ["Parameters for JBoss - Provision Software v3"](#).

**Note:** For information about the steps in this workflow, see the [Steps in this Workflow](#).

**Note:** The documentation for this workflow contains steps that are referred to by their base names. The names in the DMA user interface may have a version appended, for example, v2.

## Prerequisites for this Workflow

The following prerequisites must be satisfied before you can run the ["JBoss - Provision Software v3"](#) workflow:

1. The workflow requires unchallenged `sudo` access to a user (typically `root`) who can access all required files and directories.
2. The workflow requires the Java Development Kit (JDK) version 1.7 (or later).
3. Adequate disk space must be available to install the JBoss and Java binaries.

For information about prerequisites for JBoss, refer to the [Red Hat JBoss Product Documentation](#).

## How this Workflow Works

This topic contains the following information about the "JBoss - Provision Software v3" workflow:

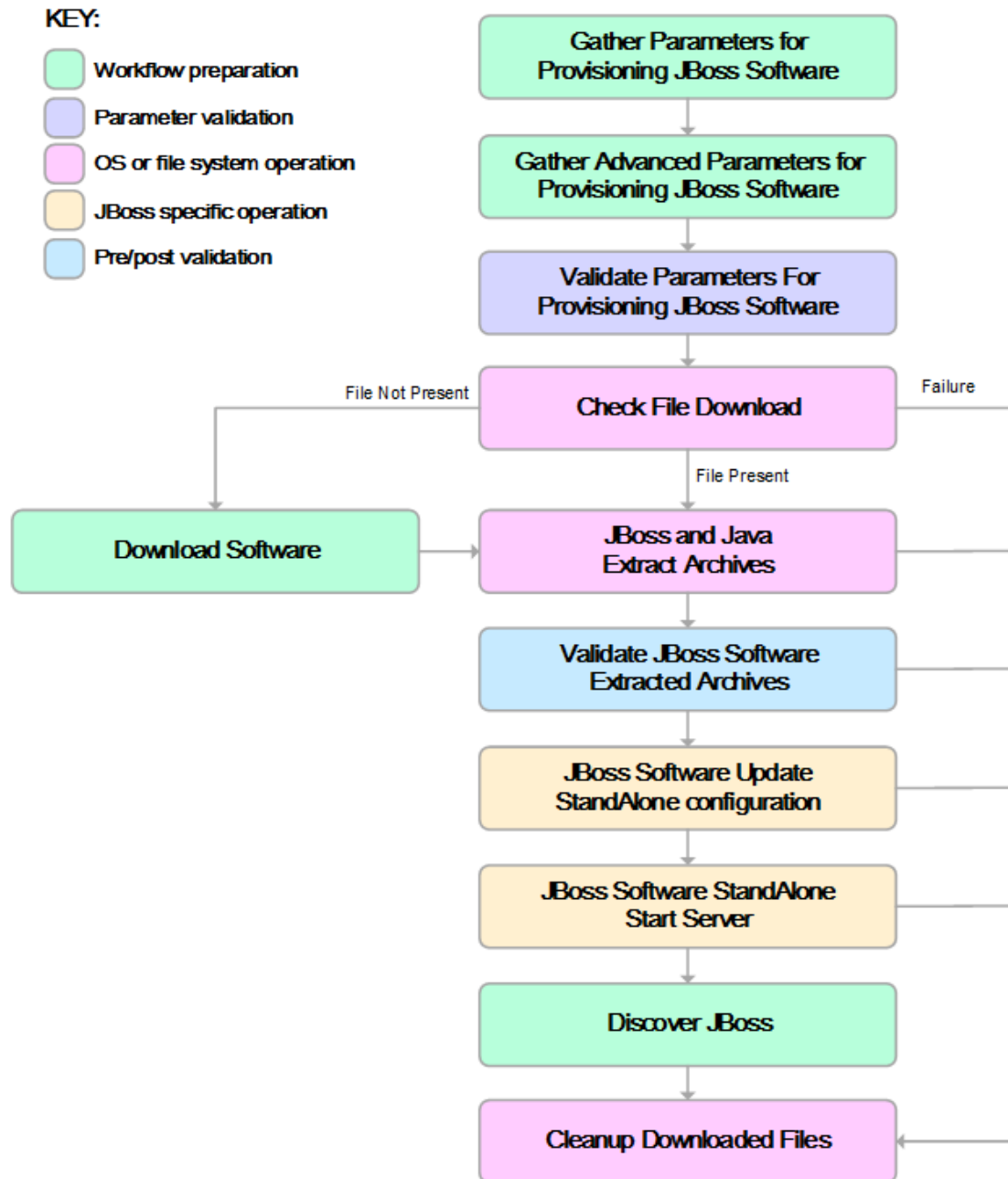
### Validation Checks Performed

The workflow checks the following things prior to extracting the binaries. If any of these checks fails, the workflow fails.

1. All required parameters have values. If any required parameter does not have a value—either a value that you specify or a default value—the workflow fails.
2. Directories and host names are valid. No illegal characters are included. The fully qualified paths specified for downloading JBoss and Java binaries exist and point to the same folder, for example: `/example/downloads/`.
3. The Java version is 1.7 or later and the revision is 5 or later.
4. The operating system is a supported platform.
5. Sufficient disk space is available to extract the binary files from the compressed archive.
6. Sufficient disk space is available to install JBoss and Java.

## Steps Executed

The "JBoss - Provision Software v3" workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure, and subsequent steps are skipped.





## Process Flow

This workflow performs the following tasks:

1. Creates the call wrapper and determines the target server platform type.
2. Inputs the required and optional parameters for the workflow.
3. Validates the parameters needed to install JBoss and Java in standalone mode.
4. Determines whether the JBoss and Java binary archives are present on the target server. If either archive is not present, the workflow downloads it from the software repository.
5. Extracts the JBoss and Java binary archives to the specified directories.
6. Validates that the Java version is 1.7 or later.
7. Updates the `standalone.xml` configuration file—required to start the server in standalone mode.
8. Starts the new standalone JBoss application server.
9. Captures information learned during the provisioning process in DMA metadata fields.
10. Cleans up downloaded files that are no longer required, based on user-specified flags in the event of workflow success or failure.

## How to Run this Workflow

This topic explains how to customize and run the ["JBoss - Provision Software v3"](#) workflow in your environment.

**Note:** Prior to running this workflow, review the ["Prerequisites for this Workflow"](#), and ensure that all requirements are satisfied.

### To customize and run the JBoss - Provision Software workflow:

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters. These are the parameters that are visible in the deployment by default.

Parameter Name	Default Value	Description
Java Binary Archive	no default	Fully qualified path to where the compressed Java software package should be located on the target machine. The directory should exist on the target machine and must be the same directory specified in the JBoss Binary Archive parameter. If the Java software package is not available in this directory, then it will be downloaded from the software repository and placed in this directory.  For example: <code>/example/downloads/jdk-7u71-linux-x64.gz</code>
Install Dir	<code>/opt/jboss</code>	Fully qualified path to where the binary files will be uncompressed.  For example: <code>/opt/jboss</code>
JBoss Binary Archive	no default	Fully qualified path to where the compressed JBoss software package should be located on the target machine. The directory should exist on the target machine and must be the same directory that is provided in the Java Binary Archive Parameter. If the JBoss software package is not available in this directory, then it will be downloaded from the software repository and placed in this directory.  For example: <code>/example/downloads/wildfly-9.0.0.Alpha1.tar.gz</code>
Download Location	no default	The location where the Java and JBoss binaries will be downloaded and saved in the target machine.
Web Service Password	no default	Password for the DMA Discovery web service API.
Web Service URL	no default	URL for the DMA Discovery web service API.
Web	no default	User capable of modifying the managed environment through the

Parameter Name	Default Value	Description
Service User		DMA Discovery web service API.

See "[Parameters for JBoss - Provision Software v3](#)" for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment.
4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values for the required parameters listed in step 2 and any additional parameters that you have exposed. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.
8. Save the changes to the workflow (click **Save** in the lower right corner).
9. Run the workflow using this deployment.

The workflow will complete and report "Success" on the Console if it has run successfully. If an invalid parameter value is specified, an error is logged, and the workflow terminates in the "Failure" state.

## Sample Scenarios

This topic shows you how to use various parameters to achieve the following provisioning scenarios in your environment using the "JBoss - Provision Software v3" workflow:

### Scenario 1: Install JBoss Enterprise Application Platform 6 (EAP)

Specify values for the following parameters to install the EAP 6.3.0 version of JBoss. The downloaded JBoss and Java binaries will be removed upon successful or unsuccessful execution of the workflow.

Step Name	Parameter Name	Example Value
Gather Parameters for Provisioning JBoss Software v2	Install Dir	/opt/jboss
	Download Location	/opt/downloads
	Java Binary Archive	/example/downloads/jdk-7u71-linux-x64.gz
	JBoss Binary Archive	/example/downloads/jboss-eap-6.3.0.zip
	Web Service Password	mypwd
	Web Service URL	https://mydmaservername:8443/dma
	Web Service User	myusername
Gather Advanced Parameters for Provisioning JBoss Software	Clean on Failure	True
	Clean on Success	True
	Install Dir	/opt/jboss

Be sure that the default values for all remaining parameters (the advanced parameters) are appropriate for your environment.

**Scenario 2: Install JBoss WildFly**

Specify values for the following parameters to install the WildFly 9.0.0 Alpha1 version of JBoss in the `/opt/wildfly` folder. The downloaded JBoss and Java binaries will not be removed upon successful or unsuccessful execution of the workflow. You need to expose the advanced parameters in your copy of the workflow in order to implement this scenario.

Step Name	Parameter Name	Example Value
Gather Parameters for Provisioning JBoss Software v2	Install Dir	<code>/opt/jboss</code>
	Download Location	<code>/opt/downloads</code>
	Java Binary Archive	<code>/example/downloads/jdk-7u71-linux-x64.gz</code>
	JBoss Binary Archive	<code>/example/downloads/wildfly-9.0.0.Alpha1.tar.gz</code>
	Web Service Password	<code>mypwd</code>
	Web Service URL	<code>https://mydmaservername:8443/dma</code>
	Web Service User	<code>myusername</code>
Gather Advanced Parameters for Provisioning JBoss Software	Clean on Failure	<code>True</code>
	Clean on Success	<code>True</code>
	Install Dir	<code>/opt/jboss</code>

## Parameters for JBoss - Provision Software v3

The following tables describe the required and optional input parameters for this workflow. Several of these parameters are not initially visible in a deployment. For many parameters, if you do not specify a value for a parameter, a default value is assigned.

### Parameters Defined in this Step: Gather Parameters for Provisioning JBoss Software

Parameter Name	Default Value	Required	Description
Install Dir	/opt/jboss	required	Fully qualified path to where the binary files will be uncompressed.
Download Location	no default	required	The location where the Java and JBoss binaries will be downloaded and saved in the target machine.
Java Binary Archive	no default	required	Fully qualified path to where the compressed Java software package should be located on the target machine. The directory should exist on the target machine and must be the same directory specified in the JBoss Binary Archive parameter. If the Java software package is not available in this directory, then it will be downloaded from the software repository and placed in this directory.  For example: /example/downloads/jdk-7u71-linux-x64.gz
JBoss Binary Archive	no default	required	Fully qualified path to where the compressed JBoss software package should be located on the target machine. The directory should exist on the target machine and must be the same directory that is provided in the Java Binary Archive Parameter. If the JBoss software package is not available in this directory, then it will be downloaded from the software repository and placed in this directory.  For example: /example/downloads/wildfly-9.0.0.Alpha1.tar.gz
Web Service Password	no default	required	Password for the DMA Discovery web service API.
Web Service URL	no default	required	URL for the DMA Discovery web service API.
Web Service User	no default	required	User capable of modifying the managed environment through the DMA Discovery web service API.

### Parameters Defined in this Step: Gather Advanced Parameters for Provisioning JBoss Software

Parameter Name	Default Value	Required	Description
Clean on Failure	True	optional	Flag that determines whether to clean up on workflow failure. If set to 'True', the workflow will clean up the downloaded

**Parameters Defined in this Step: Gather Advanced Parameters for Provisioning JBoss Software , continued**

Parameter Name	Default Value	Required	Description
			files. Valid values are 'True' and 'False'. The default value is 'True'.
Clean on Success	True	optional	Flag that determines whether to clean up on workflow success. If set to 'True', the workflow will clean up the downloaded files. Valid values are 'True' and 'False'. The default value is 'True'.
JBoss Group	no default	optional	Group ID to install JBoss software. If not specified, group ID of the root will be used.
JBoss User	no default	optional	The user installing the JBoss. If not specified, root user ID will be used.

## JBoss - Patch Software v3

This workflow applies one or more patches to the specified JBoss EAP standalone server and WildFly application server. It also supports patching the Java that is used by WebLogic domains.

To use this workflow in your environment, see the following information:

Topic	Information Included
<a href="#">"Prerequisites for this Workflow"</a>	List of prerequisites that must be satisfied before you can run this workflow
<a href="#">"How this Workflow Works"</a>	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
<a href="#">"How to Run this Workflow"</a>	Instructions for running this workflow in your environment
<a href="#">"Sample Scenario"</a>	Examples of typical parameter values for this workflow
<a href="#">"Parameters for JBoss - Patch Software v3"</a>	List of input parameters for this workflow



## Prerequisites for this Workflow

The following prerequisites must be satisfied before you can run the JBoss patching workflow:

- The latest DMA solution packs require the latest DMA platform. To use the latest solution packs, update the DMA platform. DMA 10.60.000.000 solution packs are supported on DMA 10.60.000.000 (and later).
- You have installed JBoss.
- You have provisioned a JBoss or WildFly server. You can do this by running workflows found in the DMA Application Server Provisioning Solution Pack:
  - Provision JBoss standalone EAP server.
- You have verified that the patches to be installed are appropriate for your version of JBoss or WildFly.
- You have added a link to the Java folder and added the link in the **setDomain.sh** file.

For more information about prerequisites for JBoss or WildFly, refer to the [JBoss Product Documentation](#).

## How this Workflow Works

The following information describes how the JBoss - Patch Software V3 workflow works:

### Overview

The JBoss - Patch Software workflow first prepares to apply the patch. It creates the commands that will be used to execute subsequent steps, gathers and validates the necessary input parameters, and creates additional utility parameters.

The workflow then makes sure that all necessary files exist, have valid specifications, and are in the expected locations.

Next, the workflow applies the patches. On the Console page, the workflow reports whether each patch succeeded or failed. It collects the patch identifiers of the patches that were successfully installed.

The workflow ends cleanly. It returns all JBoss components to the state they were in when the workflow started. If required, it restarts the JBoss (EAP) standalone server.

This workflow also supports patching the Java that is used by WebLogic domains. A symbolic link to the Java parent directory must be provided and specified in the `setDomain.sh` file. The Java binaries will be extracted in this folder.

### Steps Executed

The JBoss - Patch Software workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.

**Workflow Steps****Steps Used in JBoss - Patch Software**

Workflow Step	Description
Gather Parameters for Patching JBoss Software	This step gathers mandatory input parameters (user-provided) used to apply a list of patches for JBoss (EAP) Standalone Server.
Gather Advanced Parameters for Patching JBoss Software v2	This step gathers the advanced input parameters (user-provided) used to deploy a patch for JBoss (EAP) Standalone Server. Input parameters specified in this step are optional. Appropriate default values are specified.
Validate Parameters for Patching JBoss Software v3	This step validates the basic and advanced parameters provided by the user, and checks the prerequisites for patching JBoss (EAP) Standalone Server.
Check File Download	<p>This step checks for the existence of a file on the target server before downloading that file from the software repository. For each file in the list, this step does the following things:</p> <ol style="list-style-type: none"> <li>1. Determines whether the file is in the expected location on the target machine.</li> <li>2. If the file is not in the expected location, adds that file to a list of files that need to be downloaded.</li> </ol>
Decompress Wildfly Patch Archive	This step decompresses WildFly patch archive file to a specified staging location.
Download Software	This step downloads a list of files to a specified location on the target server.
Apply Patch for JBoss Software v2	This step applies the patches to JBoss(EAP) Application Server.
Cleanup Downloaded Files v2	This step removes all temporary downloaded files and archives.
Apply Patch for JBoss Java Home	This Step Extracts the Java JDK or JRE file into the Java home of the JBoss installation.
Cleanup Downloaded Files v2	This step removes all temporary downloaded files and archives.
Restore JBoss Java Home	This step restores the Java Home of the JBoss installation if the Java patching fails.
Discover JBoss	This step examines the target server's physical environment to discover information about JBoss or WildFly.

For parameter descriptions and defaults, see ["Parameters for JBoss - Patch Software v3"](#).

## How to Run this Workflow

The following instructions show you how to customize and run the JBoss - Patch Software V3 workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. Any parameters not explicitly specified in the deployment will have the default values listed in ["Parameters for JBoss - Patch Software v3"](#).

**Note:** Before following this procedure, review the ["Prerequisites for this Workflow"](#), and ensure that all requirements are satisfied.

**To use the JBoss - Patch Software workflow:**

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

### Input Parameters for Gather Parameters for Patching JBoss Software

Parameter Name	Default Value	Required	Description
JBoss Home	no default	required	Fully qualified path to the product installation directory in which to install JBoss patches. Example: /opt/jboss/jboss-as-7.0.1.Final/
JBoss Patch Binary Archives	no default	required	Fully qualified path to the comma-separated list of JBoss patch files.
Patch Staging Location	no default	required	The temporary location in which to store the patch archive. Note that the workflow fails if the directory does not exist in the location specified.

### Input Parameters for Gather Advanced Parameters for Patching JBoss Software v2

Parameter Name	Default Value	Required	Description
Clean on Failure	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon failure of the workflow.
Clean on Success	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon successful execution of the workflow.
Java JDK File	no default	optional	Name of the Java JDK File. If not present on the target will be downloaded to staging location from SA, for example, jdk-7u80-linux-x64.tar.gz.

**Input Parameters for Gather Advanced Parameters for Patching JBoss Software v2, continued**

Parameter Name	Default Value	Required	Description
Java JRE File	no default	optional	Name of the Java JRE File. If not present on the target will be downloaded to staging location from SA, for example, jre-7u80-linux-x64.tar.gz.

**Note:** This is the minimum set of parameters required to run this workflow. You may need to expose additional parameters depending on your objectives.

See "[Parameters for JBoss - Patch Software v3](#)" for detailed descriptions of all input parameters for this workflow, including default values.

3. Save the changes to the workflow (click **Save** in the lower right corner).
4. Create a new deployment.
5. On the Parameters tab, specify values for the required parameters listed in step 2. You do not need to specify values for those parameters whose default values are appropriate for your environment.
6. On the Targets tab, specify one or more targets for this deployment.

**Note:** Specify all the targets associated with your JBoss (EAP) standalone server. The first target specified must be the Administration Server.

7. Save the deployment (click **Save** in the lower right corner).
8. Run the workflow using this deployment.

**To verify the results:**

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

See the Console page output for error messages that indicate whether problems occurred during the application of the patches. Specifically, look at the JBoss Patch Server step to see the results of applying each individual patch.

## Sample Scenario

It is very straightforward to run the JBoss - Patch Software workflow. This topic shows you typical parameter values to use.

### Input Parameters for Gather Parameters for Patching JBoss Software

Parameter Name	Example Value	Description
JBoss Home	see description	Fully qualified path to the product installation directory in which to install JBoss patches. Example: /opt/jboss/jboss-eap-6.3
JBoss Patch Binary Archive	see description	Fully qualified path to the comma-separated list of JBoss patch files. Example: /root/jboss-eap-6.3.1-patch.zip,/root/jboss-eap-6.3.2-patch.zip
Patch Staging Location	see description	The temporary location in which to store the patch archive. Note that the workflow fails if the directory does not exist in the location specified. Example: /root

### Input Parameters for Gather Advanced Parameters for Patching JBoss Software v2

Parameter Name	Default Value	Required	Description
Clean on Failure	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon failure of the workflow.
Clean on Success	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon successful execution of the workflow.
Java JDK File	no default	optional	Name of the Java JDK File. If not present on the target will be downloaded to staging location from SA, for example, jdk-7u80-linux-x64.tar.gz.
Java JRE File	no default	optional	Name of the Java JRE File. If not present on the target will be downloaded to staging location from SA, for example, jre-7u80-linux-x64.tar.gz.

## Parameters for JBoss - Patch Software v3

The following tables describe the required and optional input parameters for this workflow.

### Parameters Defined in this Step: Gather Parameters for Patching JBoss Software

Parameter Name	Default Value	Required	Description
JBoss Home	no default	required	Fully qualified path to the product installation directory in which to install JBoss or WildFly patches. Example: /opt/jboss/jboss-as-7.0.1.Final/
JBoss Patch Binary Archive	no default	required	Fully qualified path to the comma-separated list of JBoss patch files.  For WildFly, only one patch binary archive is allowed.
Patch Staging Location	no default	required	The temporary location in which to store the patch archive. Note that the workflow fails if the directory does not exist in the location specified.

### Input Parameters for Gather Advanced Parameters for Patching JBoss Software v2

Parameter Name	Default Value	Required	Description
Clean on Failure	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon failure of the workflow.
Clean on Success	True	optional	Determines whether or not to remove downloaded and extracted files. This parameter also cleans up the installed directory upon successful execution of the workflow.
Java JDK File	no default	optional	Name of the Java JDK File. If not present on the target will be downloaded to staging location from SA, for example, jdk-7u80-linux-x64.tar.gz.
Java JRE File	no default	optional	Name of the Java JRE File. If not present on the target will be downloaded to staging location from SA, for example, jre-7u80-linux-x64.tar.gz.

## JBoss - Rollback Patch Software v2

This workflow rolls back one or more patches from the specified JBoss or WildFly application server.

To use this workflow in your environment, see the following information:

Topic	Information Included
"Prerequisites for this Workflow"	List of prerequisites that must be satisfied before you can run this workflow
"How this Workflow Works"	Information about what the workflow does, including validation checks performed, steps executed, and a high-level process flow
"How to Run this Workflow"	Instructions for running this workflow in your environment
" Sample Scenario"	Examples of typical parameter values for this workflow
"Parameters for JBoss - Rollback Patch Software v2"	List of input parameters for this workflow



## Prerequisites for this Workflow

The following prerequisites must be satisfied before you can run the JBoss or WildFly rollback patching workflow:

- This solution requires DMA version 10.50 (or later).

The latest DMA solution packs require the latest DMA platform. To use the latest solution packs, update the DMA platform. DMA10.60.000.000 solution packs are supported on DMA10.60.000.000 (and later).

- You have installed JBoss.
- You have provisioned a JBoss EAP server or WildFly application server. You can do this by running workflows found in the DMA Application Server Provisioning Solution Pack:
  - Provision JBoss standalone EAP server.
- You have verified that one or more patches are installed are applied to JBoss or WildFly application server.

For more information about prerequisites for JBoss or WildFly rollback patch, refer to the [JBoss Product Documentation](#).

## How this Workflow Works

The following information describes how the JBoss - Rollback Patch Software v2 workflow works:

### Overview

The JBoss - Rollback Patch Software workflow first prepares to roll back the patch. It creates the commands that will be used to execute subsequent steps, gathers and validates the necessary input parameters, and creates additional utility parameters.

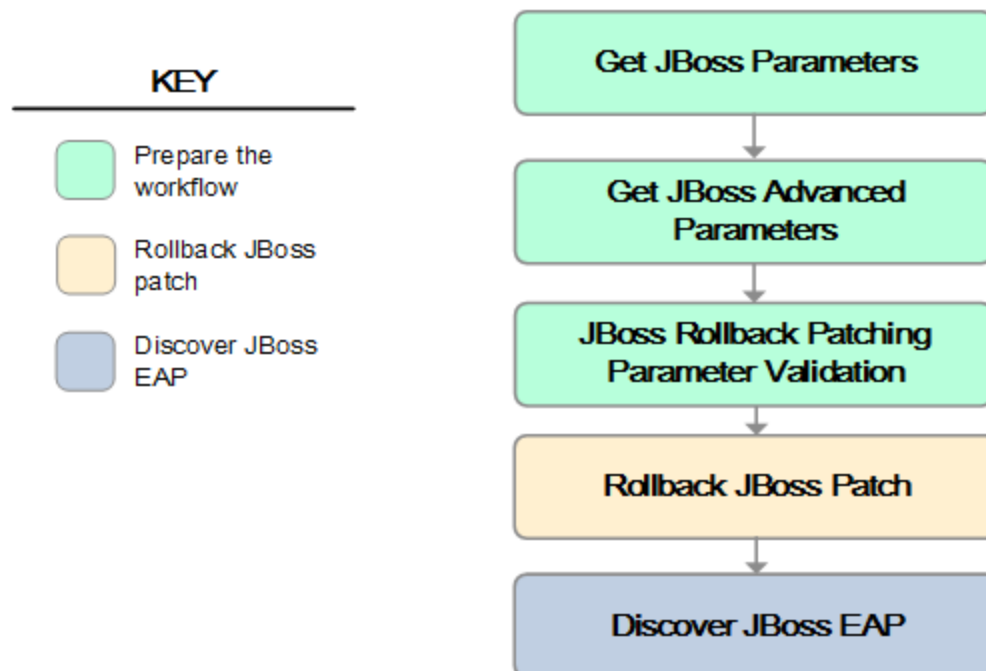
The workflow then makes sure that all necessary files exist, have valid specifications, and are in the expected locations.

Next, the workflow rolls back the patch. On the Console page, the workflow reports if the patch roll back succeeded or failed. It collects the patch identifiers of the patches that were successfully removed.

The workflow ends cleanly. It returns all JBoss or WildFly components to the state they were in when the workflow started. If required, it restarts the JBoss or WildFly application server.

### Steps Executed

The JBoss - Rollback Patch Software workflow includes the following steps. Each step must complete successfully before the next step can start. If a step fails, the workflow reports a failure and all subsequent steps are skipped.



**Workflow Steps****Steps Used in JBoss - Rollback Patch Software v2**

Workflow Step	Description
Gather Parameters for Rollback Patching JBoss Software	This step gathers mandatory input parameters (user-provided) used to apply a list of patches for JBoss (EAP) Standalone Server or WildFly application server.
Gather Advanced Parameters for Rollback Patching JBoss Software	This step gathers the advanced input parameters (user-provided) used to deploy a patch for JBoss or WildFly application server. Input parameters specified in this step are optional. Appropriate default values are specified.
Validate Parameters for Rollback Patching JBoss Software	This step gathers and validates the parameters required to remove patches from a JBoss or WildFly application server.
Rollback Patch for JBoss Software v2	The step rolls back patches from the specified target.
Discover JBoss	This step examines the target server's physical environment to discover information about JBoss or WildFly.

For parameter descriptions and defaults, see ["Parameters for JBoss - Rollback Patch Software v2" on page 63](#).

## How to Run this Workflow

The following instructions show you how to customize and run the JBoss - Rollback Patch Software workflow in your environment.

The workflow provides default values for some parameters. These default values are usually sufficient for a "typical" installation. You can override the defaults by specifying parameter values in the deployment. Any parameters not explicitly specified in the deployment will have the default values listed in ["Parameters for JBoss - Rollback Patch Software v2"](#).

**Note:** Before following this procedure, review the ["Prerequisites for this Workflow"](#), and ensure that all requirements are satisfied.

### To use the JBoss - Rollback Patch Software

1. Create a deployable copy of the workflow.
2. Determine the values that you will specify for the following parameters:

#### Input Parameters for Gather JBoss Parameters

Parameter Name	Default Value	Required	Description
JBoss Home	no default	required	Fully qualified path to the product installation directory from which to roll back JBoss patches. Example: /opt/jboss/jboss-as-7.0.1.Final/
JBoss Patch IDs	no default	required	Fully qualified path to the comma-separated list of JBoss patch IDs.

#### Input Parameters for Gather JBoss Advanced Parameters

Parameter Name	Default Value	Required	Description
Override all	True	optional	Bypasses any content verification on the miscellaneous items changed by the patch that is rolled back.
Reset Configuration	True	optional	Updates the installation configuration and resets the snapshots that were taken when the patch was applied.

**Note:** This is the minimum set of parameters required to run this workflow. You may need to expose additional parameters depending on your objectives.

See ["Parameters for JBoss - Rollback Patch Software v2"](#) for detailed descriptions of all input parameters for this workflow, including default values.

3. In the workflow editor, expose any additional parameters that you need. You will specify values for those parameters when you create the deployment.

4. Save the changes to the workflow (click **Save** in the lower right corner).
5. Create a new deployment.
6. On the Parameters tab, specify values for the required parameters listed in step 2. You do not need to specify values for those parameters whose default values are appropriate for your environment.
7. On the Targets tab, specify one or more targets for this deployment.

**Note:** Specify all the targets associated with your JBoss (EAP) standalone server. The first target specified must be the Administration Server.

8. Save the deployment (click **Save** in the lower right corner).
9. Run the workflow using this deployment.

**To verify the results:**

The workflow will complete and report SUCCESS on the Console if it has run successfully. If an error occurs during workflow execution, the error is logged, and the workflow terminates in the FAILURE state.

See the Console page output for error messages that indicate whether problems occurred during the application of the patches. Specifically, look at the JBoss Patch Server step to see the results of applying each individual patch.

## Sample Scenario

It is very straightforward to run the JBoss - Rollback Patch Software v2 workflow. This topic shows you typical parameter values to use.

### Input Parameters for Gather Parameters for Rollback Patching JBoss Software

Parameter Name	Example Value	Description
JBoss Home	see description	Fully qualified path to the product installation directory from which to roll back JBoss patches. Example: /opt/jboss/jboss-eap-6.3
JBoss Patch IDs	see description	Fully qualified path to the comma-separated list of JBoss patch IDs. Example: jboss-eap-6.3.1.CP

### Input Parameters for Gather Advanced Parameters for Rollback Patching JBoss Software

Parameter Name	Example Value	Description
Reset Configuration	see description	Updates the installation configuration and resets the snapshots that were taken when the patch was applied. Valid values are True and False. The default value is True.

## Parameters for JBoss - Rollback Patch Software v2

The following tables describe the required and optional input parameters for this workflow.

### Parameters Defined in this Step: Gather Parameters for Rollback Patching JBoss Software

Parameter Name	Default Value	Required	Description
JBoss Home	no default	required	Fully qualified path to the product installation directory from which to roll back JBoss patches.  Example: /opt/jboss/jboss-as-7.0.1.Final/
JBoss Patch IDs	no default	required	Fully qualified path to the comma-separated list of JBoss patch IDs.

### Additional Parameters Defined in this Step: Gather Advanced Parameters for Rollback Patching JBoss Software

Parameter Name	Default Value	Required	Description
Override all	True	optional	If true, this bypasses any content verification on the miscellaneous items changed by the patch that is rolled back.
Reset Configuration	True	optional	If True, this updates the installation configuration and resets the snapshots that were taken when the patch was applied.

# Send documentation feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

**Feedback on Workflows for Red Hat JBoss (Database and Middleware Automation  
10.60.000.000)**

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to [hpe\\_dma\\_docs@hpe.com](mailto:hpe_dma_docs@hpe.com).

We appreciate your feedback!