



Getting Started Guide

1 Getting Started with WildFly 8

- 1.1 Download
- 1.2 Requirements
- 1.3 Installation
- 1.4 WildFly - A Quick Tour
 - 1.4.1 WildFly 8 Directory Structure
 - 1.4.2 WildFly 8 Configurations
 - 1.4.3 Starting WildFly 8
 - 1.4.4 Starting WildFly 8 with an Alternate Configuration
 - 1.4.5 Managing your WildFly 8
 - 1.4.6 Modifying the Example DataSource


2 All WildFly 8 documentation

Getting Started with WildFly 8

WildFly 8 is the latest release in a series of JBoss open-source application server offerings. WildFly 8 is an exceptionally fast, lightweight and powerful implementation of the Java Enterprise Edition 7 Platform specifications. The state-of-the-art architecture built on the Modular Service Container enables services on-demand when your application requires them. The table below lists the Java Enterprise Edition 7 technologies and the technologies available in Wildfly 8 server configuration profiles.

| Java EE 7 Platform Technology | Java EE 7 Full Profile | Java EE 7 Web Profile | WildFly 8 Full Profile | WildFly 8 Web Profile |
|--|--------------------------|-----------------------|-------------------------------|-----------------------|
| JSR-356: Java API for Web Socket | X | X | X | X |
| JSR-353: Java API for JSON Processing | X | X | X | X |
| JSR-340: Java Servlet 3.1 | X | X | X | X |
| JSR-344: JavaServer Faces 2.2 | X | X | X | X |
| JSR-341: Expression Language 3.0 | X | X | X | X |
| JSR-245: JavaServer Pages 2.3 | X | X | X | X |
| JSR-52: Standard Tag Library for JavaServer Pages (JSTL) 1.2 | X | X | X | X |
| JSR-352: Batch Applications for the Java Platform 1.0 | X | -- | X | -- |
| JSR-236: Concurrency Utilities for Java EE 1.0 | X | X | X | X |
| JSR-346: Contexts and Dependency Injection for Java 1.1 | X | X | X | X |
| JSR-330: Dependency Injection for Java 1.0 | X | X | X | X |
| JSR-349: Bean Validation 1.1 | X | X | X | X |
| JSR-345: Enterprise JavaBeans 3.2 | X CMP 2.0 Optional | X (Lite) | X CMP 2.0 Not Available | X (Lite) |
| JSR-318: Interceptors 1.2 | X | X | X | X |
| JSR-322: Java EE Connector Architecture 1.7 | X | -- | X | X |
| JSR-338: Java Persistence 2.1 | X | X | X | X |
| JSR-250: Common Annotations for the Java Platform 1.2 | X | X | X | X |
| JSR-343: Java Message Service API 2.0 | X | -- | X | -- |

| | | | | |
|--|----------|----|----|----|
| JSR-907: Java Transaction API 1.2 | X | X | X | X |
| JSR-919: JavaMail 1.5 | X | -- | X | X |
| JSR-339: Java API for RESTful Web Services 2.0 | X | X | X | X |
| JSR-109: Implementing Enterprise Web Services 1.3 | X | -- | X | -- |
| JSR-224: Java API for XML-Based Web Services 2.2 | X | X | X | X |
| JSR-181: Web Services Metadata for the Java Platform | X | -- | X | -- |
| JSR-101: Java API for XML-Based RPC 1.1 | Optional | -- | -- | -- |
| JSR-67: Java APIs for XML Messaging 1.3 | X | -- | X | -- |
| JSR-93: Java API for XML Registries | Optional | -- | -- | -- |
| JSR-196: Java Authentication Service Provider Interface for Containers 1.1 | X | -- | X | -- |
| JSR-115: Java Authorization Contract for Containers 1.5 | X | -- | X | -- |
| JSR-88: Java EE Application Deployment 1.2 | Optional | -- | -- | -- |
| JSR-77: J2EE Management 1.1 | X | | X | |
| JSR-45: Debugging Support for Other Languages 1.0 | X | X | X | X |

 **Missing HornetQ and JMS?**
The WildFly Web Profile doesn't include JMS (provided by HornetQ) by default. If you want to use messaging, make sure you start the server using the "Full Profile" configuration.

This document provides a quick overview on how to download and get started using WildFly 8 for your application development. For in-depth content on administrative features, refer to the WildFly 8 Admin Guide.

Download

Wildfly 8 distributions can be obtained from:


wildfly.org/downloads

WildFly 8 provides a single distribution available in zip or tar file formats.

- **wildfly-8.0.0.Final.zip**
- **wildfly-8.0.0.Final.tar.gz**

Requirements

- Java SE 7 or later (we recommend that you use the latest update available)

 Java SE 8 can be used with Wildfly 8

Installation

Simply extract your chosen download to the directory of your choice. You can install WildFly 8 on any operating system that supports the zip or tar formats. Refer to the Release Notes for additional information related to the release.

WildFly - A Quick Tour

Now that you've downloaded WildFly 8, the next thing to discuss is the layout of the distribution and explore the server directory structure, key configuration files, log files, user deployments and so on. It's worth familiarizing yourself with the layout so that you'll be able to find your way around when it comes to deploying your own applications.

WildFly 8 Directory Structure

| DIRECTORY | DESCRIPTION |
|-----------------------|---|
| appclient | Configuration files, deployment content, and writable areas used by the application client container run from this installation. |
| bin | Start up scripts, start up configuration files and various command line utilities like Vault, add-user and Java diagnostic report available for Unix and Windows environments |
| bin/client | Contains a client jar for use by non-maven based clients. |
| docs/schema | XML schema definition files |
| docs/examples/configs | Example configuration files representing specific use cases |
| domain | Configuration files, deployment content, and writable areas used by the domain mode processes run from this installation. |
| modules | WildFly 8 is based on a modular classloading architecture. The various modules used in the server are stored here. |
| standalone | Configuration files, deployment content, and writable areas used by the single standalone server run from this installation. |
| welcome-content | Default Welcome Page content |

Standalone Directory Structure

In "**standalone**" mode each WildFly 8 server instance is an independent process (similar to previous JBoss AS versions; e.g., 3, 4, 5, or 6). The configuration files, deployment content and writable areas used by the single standalone server run from a WildFly installation are found in the following subdirectories under the top level "standalone" directory:

| DIRECTORY | DESCRIPTION |
|---------------|---|
| configuration | Configuration files for the standalone server that runs off of this installation. All configuration information for the running server is located here and is the single place for configuration modifications for the standalone server. |
| data | Persistent information written by the server to survive a restart of the server |
| deployments | End user deployment content can be placed in this directory for automatic detection and deployment of that content into the server's runtime. NOTE: The server's management API is recommended for installing deployment content. File system based deployment scanning capabilities remain for developer convenience. |
| lib/ext | Location for installed library jars referenced by applications using the Extension-List mechanism |
| log | standalone server log files |
| tmp | location for temporary files written by the server |
| tmp/auth | Special location used to exchange authentication tokens with local clients so they can confirm that they are local to the running AS process. |

Domain Directory Structure

A key feature of WildFly 8 is the managing multiple servers from a single control point. A collection of multiple servers are referred to as a "**domain**". Domains can span multiple physical (or virtual) machines with all WildFly instances on a given host under the control of a Host Controller process. The Host Controllers interact with the Domain Controller to control the lifecycle of the WildFly instances running on that host and to assist the Domain Controller in managing them. The configuration files, deployment content and writeable

areas used by domain mode processes run from a WildFly installation are found in the following subdirectories under the top level "domain" directory:

| DIRECTORY | DESCRIPTION |
|---------------|--|
| configuration | Configuration files for the domain and for the Host Controller and any servers running off of this installation. All configuration information for the servers managed within the domain is located here and is the single place for configuration information. |
| content | an internal working area for the Host Controller that controls this installation. This is where it internally stores deployment content. This directory is not meant to be manipulated by end users. Note that "domain" mode does not support deploying content based on scanning a file system. |
| lib/ext | Location for installed library jars referenced by applications using the Extension-List mechanism |
| log | Location where the Host Controller process writes its logs. The Process Controller, a small lightweight process that actually spawns the other Host Controller process and any Application Server processes also writes a log here. |
| servers | Writable area used by each Application Server instance that runs from this installation. Each Application Server instance will have its own subdirectory, created when the server is first started. In each server's subdirectory there will be the following subdirectories: data -- information written by the server that needs to survive a restart of the server log -- the server's log files tmp -- location for temporary files written by the server |
| tmp | location for temporary files written by the server |
| tmp/auth | Special location used to exchange authentication tokens with local clients so they can confirm that they are local to the running AS process. |

WildFly 8 Configurations

Standalone Server Configurations

- standalone.xml (*default*)
 - Java Enterprise Edition 7 web profile certified configuration with the required technologies plus those noted in the table above.
- standalone-ha.xml
 - Java Enterprise Edition 7 web profile certified configuration with high availability
- standalone-full.xml
 - Java Enterprise Edition 7 full profile certified configuration including all the required EE 7 technologies
- standalone-full-ha.xml
 - Java Enterprise Edition 7 full profile certified configuration with high availability

Domain Server Configurations

- domain.xml
 - Java Enterprise Edition 7 full and web profiles available with or without high availability

Important to note is that the **domain** and **standalone** modes determine how the servers are managed not what capabilities they provide.

Starting WildFly 8

To start WildFly 8 using the default web profile configuration in "standalone" mode, change directory to \$JBOSS_HOME/bin.

```
./standalone.sh
```

To start the default web profile configuration using domain management capabilities,

```
./domain.sh
```

Starting WildFly 8 with an Alternate Configuration

If you choose to start your server with one of the other provided configurations, they can be accessed by passing the `--server-config` argument with the server-config file to be used.

To use the full profile with clustering capabilities, use the following syntax from `$JBOSS_HOME/bin`:

```
./standalone.sh --server-config=standalone-full-ha.xml
```

Similarly to start an alternate configuration in *domain* mode:

```
./domain.sh --domain-config=my-domain-configuration.xml
```

Alternatively, you can create your own selecting the additional subsystems you want to add, remove, or modify.

Test Your Installation

After executing one of the above commands, you should see output similar to what's shown below.

```
=====
JBoss Bootstrap Environment

JBOSS_HOME: /work/wildfly-8.0.0.Final

JAVA: /usr/jdk1.7/bin/java

JAVA_OPTS: -server -Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true
-Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -
Dsun.rmi.dgc.server.gcInterval=3600000
-Djboss.modules.system.pkgs=org.jboss.byteman

=====

22:58:56,741 INFO  [org.jboss.modules] (main) JBoss Modules version 1.3.0.Final
22:58:56,964 INFO  [org.jboss.msc] (main) JBoss MSC version 1.2.0.Final
22:58:57,046 INFO  [org.jboss.as] (MSC service thread 1-6) JBAS015899: WildFly 8.0.0.Final "WildFly"
starting
<snip>
23:04:27,150 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.0.0.Final "WildFly"
started in 1974ms - Started 183 of 232 services (80 services are lazy, passive or on-demand)
```

As with previous JBoss releases, you can point your browser to **<http://localhost:8080>** (if using the default configured http port) which brings you to the Welcome Screen:



From here you can access links to the WildFly community documentation set, stay up-to-date on the latest project information, have a discussion in the user forum and access the enhanced web-based Administration Console. Or, if you uncover a defect while using WildFly, report an issue to inform us (attached patches will be reviewed). This landing page is recommended for convenient access to information about WildFly 8 but can easily be replaced with your own if desired.

Managing your WildFly 8

WildFly 8 offers two administrative mechanisms for managing your running instance:

- web-based Administration Console
- command-line interface

Authentication

By default WildFly 8 is now distributed with security enabled for the management interfaces, this means that before you connect using the administration console or remotely using the CLI you will need to add a new user, this can be achieved simply by using the `add-user.sh` script in the bin folder.

After starting the script you will be guided through the process to add a new user: -

```
./add-user.sh
What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):
```

In this case a new user is being added for the purpose of managing the servers so select option a.

You will then be prompted to enter the details of the new user being added: -

```
Enter the details of the new user to add.
Realm (ManagementRealm) :
Username :
Password :
Re-enter Password :
```

It is important to leave the name of the realm as 'ManagementRealm' as this needs to match the name used in the server's configuration, for the remaining fields enter the new username, password and password confirmation.

Provided there are no errors in the values entered you will then be asked to confirm that you want to add the user, the user will be written to the properties files used for authentication and a confirmation message will be displayed.

The modified time of the properties files are inspected at the time of authentication and the files reloaded if they have changed, for this reason you do not need to re-start the server after adding a new user.

Administration Console

To access the web-based Administration Console, simply follow the link from the Welcome Screen. To directly access the Management Console, point your browser at:

http://localhost:9990/console

NOTE: port 9990 is the default port configured.

```
<management-interfaces>
  <native-interface security-realm="ManagementRealm">
    <socket-binding native="management-native"/>
  </native-interface>
  <http-interface security-realm="ManagementRealm">
    <socket-binding http="management-http"/>
  </http-interface>
</management-interfaces>
```

If you modify the *management-http* socket binding in your running configuration: adjust the above command accordingly. If such modifications are made, then the link from the Welcome Screen will also be inaccessible.

If you have not yet added at least one management user an error page will be displayed asking you to add a new user, after a user has been added you can click on the 'Try Again' link at the bottom of the error page to try connecting to the administration console again.

Command-Line Interface

If you prefer to manage your server from the command line (or batching), the *jboss-cli.sh* script provides the same capabilities available via the web-based UI. This script is accessed from \$JBOSS_HOME/bin directory; e.g.,

```
cd $JBOSS_HOME/bin
./jboss-cli.sh --connect
Connected to standalone controller at localhost:9999
```

Notice if no host or port information provided, it will default to localhost:9999.

When running locally to the WildFly process the CLI will silently authenticate against the server by exchanging tokens on the file system, the purpose of this exchange is to verify that the client does have access to the local file system. If the CLI is connecting to a remote WildFly installation then you will be prompted to enter the username and password of a user already added to the realm.

Once connected you can add, modify, remove resources and deploy or undeploy applications. For a complete list of commands and command syntax, type **help** once connected.



5 Ways to Deploy your Application to WildFly 8

Check out this video showing step-by-step options for deploying your application to WildFly 8

Modifying the Example DataSource

As with previous JBoss application server releases, a default data source, **ExampleDS**, is configured using the embedded H2 database for developer convenience. There are two ways to define datasource configurations:

1. as a module
2. as a deployment

In the provided configurations, H2 is configured as a module. The module is located in the `$JBOSS_HOME/modules/com/h2database/h2` directory. The H2 datasource configuration is shown below.

```
<subsystem xmlns="urn:jboss:domain:datasources:1.0">
  <datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS">
      <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1</connection-url>
      <driver>h2</driver>
      <pool>
        <min-pool-size>10</min-pool-size>
        <max-pool-size>20</max-pool-size>
        <prefill>true</prefill>
      </pool>
      <security>
        <user-name>sa</user-name>
        <password>sa</password>
      </security>
    </datasource>
    <xa-datasource jndi-name="java:jboss/datasources/ExampleXADS" pool-name="ExampleXADS">
      <driver>h2</driver>
      <xa-datasource-property name="URL">jdbc:h2:mem:test</xa-datasource-property>
      <xa-pool>
        <min-pool-size>10</min-pool-size>
        <max-pool-size>20</max-pool-size>
        <prefill>true</prefill>
      </xa-pool>
      <security>
        <user-name>sa</user-name>
        <password>sa</password>
      </security>
    </xa-datasource>
    <drivers>
      <driver name="h2" module="com.h2database.h2">
        <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
      </driver>
    </drivers>
  </datasources>
</subsystem>
```

The datasource subsystem is provided by the IronJacamar project. For a detailed description of the available configuration properties, please consult the project documentation.

- IronJacamar homepage: <http://www.jboss.org/ironjacamar>
- Project Documentation: <http://www.jboss.org/ironjacamar/docs>
- Schema description: http://docs.jboss.org/ironjacamar/userguide/1.0/en-US/html/deployment.html#deployingds_descriptor

Configure Logging in WildFly 8

WildFly 8 logging can be configured in the XML configuration files, the web console or the command line interface. You can get more detail on the Logging Configuration page.

Turn on debugging for a specific category:


```
<subsystem xmlns="urn:jboss:domain:logging:1.0">
  <console-handler name="CONSOLE" autoflush="true">
    <level name="DEBUG"/>
    <formatter>
      <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n"/>
    </formatter>
  </console-handler>
  [...]
  <logger category="org.jboss.as">
    <level name="DEBUG"/>
  </logger>
  [...]
</subsystem>
```

By default the `server.log` is configured to include all levels in its log output. In the above example we changed the console to also display debug messages.

All WildFly 8 documentation

There are several guides in the WildFly 8 documentation series. This list gives an overview of each of the guides:

- *Getting Started Guide - Explains how to download and start WildFly 8.
- *Getting Started Developing Applications Guide - Talks you through developing your first applications on WildFly 8, and introduces you to JBoss Tools and how to deploy your applications.
- *JavaEE 6 Tutorial - A Java EE 6 Tutorial.
- *Admin Guide - Tells you how to configure and manage your WildFly 8 instances.
- *Developer Guide - Contains concepts that you need to be aware of when developing applications for WildFly 8. Classloading is explained in depth.
- *High Availability Guide - Reference guide for how to set up clustered WildFly 8 instances.
- *Extending WildFly 8 - A guide to adding new functionality to WildFly 8.

1 Comment



Marc Walter

May 04, 2012

In my opinion there are missing some important parts about configuring Java.

First of all I found out that JBoss AS7 runs on Java 64-bit. So if there's a 64-bit JDK installed no need to download and install a 32-bit Java version.

You don't need to setup an environment variable `JAVA_HOME`. Use the config files to configure the Java installation directory.

If you run AS7 on Windows and the path to your Java installation contains spaces you'll run into issues if you put the path into quotes. So don't use quotes! For example write `set JAVA_HOME=C:/Program Files/Java/jdk 1.7.0_02` into your `standalone.conf.bat`.

Hope this helps!