



Apache Tomcat 6.0

Version 6.0.44,
May 8 2015



The **Apache**
Software Foundation
<http://www.apache.org/>

Apache Tomcat 6.0

Tomcat Web Application Deployment

Table of Contents

- [Introduction](#)
- [Installation](#)
- [A word on Contexts](#)
- [Deployment on Tomcat startup](#)
- [Deploying on a running Tomcat server](#)
- [Deploying using the Tomcat Manager](#)
- [Deploying using the Client Deployer Package](#)

Introduction

Deployment is the term used for the process of installing a web application (either a 3rd party WAR or your own custom web application) into the Tomcat server.

Web application deployment may be accomplished in a number of ways within the Tomcat server.

- Statically; the web application is setup before Tomcat is started
- Dynamically; by directly manipulating already deployed web applications (relying on *auto-deployment* feature) or remotely by using the Tomcat Manager web application

The [Tomcat Manager](#) is a web application that can be used interactively (via HTML GUI) or programmatically (via URL-based API) to deploy and manage web applications.

There are a number of ways to perform deployment that rely on the Manager web application. Apache Tomcat provides tasks for Apache Ant build tool. [Apache Tomcat Maven Plugin](#) project provides integration with Apache Maven. There is also a tool called the Client Deployer, which can be used from a command line and provides additional functionality such as compiling and validating web applications as well as packaging web application into web application resource (WAR) files.

Installation

There is no installation required for static deployment of web applications as this is provided out of the box by Tomcat. Nor is any installation required for deployment functions with the Tomcat Manager, although some configuration is required as

detailed in the [Tomcat Manager manual](#). An installation is however required if you wish to use the Tomcat Client Deployer (TCD).

The TCD is not packaged with the Tomcat core distribution, and must therefore be downloaded separately from the Downloads area. The download is usually labelled *apache-tomcat-6.0.x-deployer*.

TCD has prerequisites of Apache Ant 1.6.2+ and a Java installation. Your environment should define an ANT_HOME environment value pointing to the root of your Ant installation, and a JAVA_HOME value pointing to your Java installation. Additionally, you should ensure Ant's ant command, and the Java javac compiler command run from the command shell that your operating system provides.

1. Download the TCD distribution
2. The TCD package need not be extracted into any existing Tomcat installation, it can be extracted to any location.
3. Read Using the [Tomcat Client Deployer](#)

A word on Contexts

In talking about deployment of web applications, the concept of a *Context* is required to be understood. A Context is what Tomcat calls a web application.

In order to configure a Context within Tomcat a *Context Descriptor* is required. A Context Descriptor is simply an XML file that contains Tomcat related configuration for a Context, e.g naming resources or session manager configuration. In earlier versions of Tomcat the content of a Context Descriptor configuration was often stored within Tomcat's primary configuration file *server.xml* but this is now discouraged (although it currently still works).

Context Descriptors not only help Tomcat to know how to configure Contexts but other tools such as the Tomcat Manager and TCD often use these Context Descriptors to perform their roles properly.

The locations for Context Descriptors are:

1. \$CATALINA_BASE/conf/[enginename]/[hostname]/[webappname].xml
2. \$CATALINA_BASE/webapps/[webappname]/META-INF/context.xml

Files in (1) are named [webappname].xml but files in (2) are named context.xml. If a Context Descriptor is not provided for a Context, Tomcat configures the Context using default values.

Deployment on Tomcat startup

If you are not interested in using the Tomcat Manager, or TCD, then you'll need to deploy your web applications statically to Tomcat, followed by a Tomcat startup. The location you deploy web applications to for this type of deployment is called the appBase which is specified per Host. You either copy a so-called *exploded web application*, i.e non-compressed, to this location, or a compressed web application resource .WAR file.

The web applications present in the location specified by the Host's (default Host is "localhost") appBase attribute (default appBase is "\$CATALINA_BASE/webapps") will be deployed on Tomcat startup only if the Host's deployOnStartup attribute is "true".

The following deployment sequence will occur on Tomcat startup in that case:

1. Any Context Descriptors will be deployed first.
2. Exploded web applications not referenced by any Context Descriptor will then be deployed. Note that if an exploded web application has an associated .WAR file in the appBase, Tomcat will not detect if the associated .WAR has been updated while Tomcat was stopped and will deploy the exploded web application as is. The exploded web application will not be removed and replaced with the contents of the updated .WAR file.
3. .WAR files will be deployed

Note again that for each deployed web application, a Context Descriptor will be created *unless one exists already*.

Deploying on a running Tomcat server

It is possible to deploy web applications to a running Tomcat server.

If the Host `autoDeploy` attribute is "true", the Host will attempt to deploy and update web applications dynamically, as needed, for example if a new .WAR is dropped into the appBase. For this to work, the Host needs to have background processing enabled which is the default configuration.

`autoDeploy` set to "true" and a running Tomcat allows for:

- Deployment of .WAR files copied into the Host appBase.
- Deployment of exploded web applications which are copied into the Host appBase.
- Re-deployment of a web application which has already been deployed from a .WAR when the new .WAR is provided. In this case the exploded web application is removed, and the .WAR is expanded again. Note that the explosion will not occur if the Host is configured so that .WARs are not exploded with a `unpackWARs` attribute set to "false", in which case the web application will be simply redeployed as a compressed archive.
- Re-deployment of a web application if the `/WEB-INF/web.xml` file (or any other resource defined as a `WatchedResource`) is updated.
- Re-deployment of a web application if the Context Descriptor file from which the web application has been deployed is updated.
- Re-deployment of a web application if a Context Descriptor file (with a filename corresponding to the Context path of the previously deployed web application) is added to the `$CATALINA_BASE/conf/[engineName]/[hostname]/` directory.
- Undeployment of a web application if its document base (`docBase`) is deleted. Note that on Windows, this assumes that anti-locking features (see Context configuration) are enabled, otherwise it is not possible to delete the resources of a running web application.

Note that web application reloading can also be configured in the loader, in which case loaded classes will be tracked for changes.

Deploying using the Tomcat Manager

The Tomcat Manager is covered in its [own manual page](#).

Deploying using the Client Deployer Package

Finally, deployment of web application may be achieved using the Tomcat Client Deployer. This is a package which can be used to validate, compile, compress to

.WAR, and deploy web applications to production or development Tomcat servers. It should be noted that this feature uses the Tomcat Manager and as such the target Tomcat server should be running.

It is assumed the user will be familiar with Apache Ant for using the TCD. Apache Ant is a scripted build tool. The TCD comes pre-packaged with a build script to use. Only a modest understanding of Apache Ant is required (installation as listed earlier in this page, and familiarity with using the operating system command shell and configuring environment variables).

The TCD includes Ant tasks, the Jasper page compiler for JSP compilation before deployment, as well as a task which validates the web application Context Descriptor. The validator task (class `org.apache.catalina.ant.ValidatorTask`) allows only one parameter: the base path of an exploded web application.

The TCD uses an exploded web application as input (see the list of the properties used below). A web application that is programmatically deployed with the deployer may include a Context Descriptor in `/META-INF/context.xml`.

The TCD includes a ready-to-use Ant script, with the following targets:

- `compile` (default): Compile and validate the web application. This can be used standalone, and does not need a running Tomcat server. The compiled application will only run on the associated Tomcat 6.0.x server release, and is not guaranteed to work on another Tomcat release, as the code generated by Jasper depends on its runtime component. It should also be noted that this target will also compile automatically any Java source file located in the `/WEB-INF/classes` folder of the web application.
- `deploy`: Deploy a web application (compiled or not) to a Tomcat server.
- `undeploy`: Undeploy a web application
- `start`: Start web application
- `reload`: Reload web application
- `stop`: Stop web application

In order for the deployment to be configured, create a file called `deployer.properties` in the TCD installation directory root. In this file, add the following name=value pairs per line:

Additionally, you will need to ensure that a user has been setup for the target Tomcat Manager (which TCD uses) otherwise the TCD will not authenticate with the Tomcat Manager and the deployment will fail. To do this, see the Tomcat Manager page.

- `build`: The build folder used will be, by default, `${build}/webapp/${path}`. After the end of the execution of the `compile` target, the web application .WAR will be located at `${build}/webapp/${path}.war`.
- `webapp`: The directory containing the exploded web application which will be compiled and validated. By default, the folder is `myapp`.
- `path`: Deployed context path of the web application, by default `/myapp`.
- `url`: Absolute URL to the Tomcat Manager web application of a running Tomcat server, which will be used to deploy and undeploy the web application. By default, the deployer will attempt to access a Tomcat instance running on localhost, at `http://localhost:8080/manager`.
- `username`: Tomcat Manager username (user should have a role of manager-script)
- `password`: Tomcat Manager password.

