

# Developing

Permalink to this page: <https://wiki.apache.org/confluence/x/8CklBg>

## About

This section of the FAQ discusses common questions related to Tomcat development.

## Questions

1. [Hacking](#)
  - a. [How do I start hacking Tomcat in Eclipse?](#)
2. [Debugging](#)
  - a. [How do I configure Tomcat to support remote debugging?](#)
  - b. [How do I remotely debug Tomcat using Eclipse?](#)
  - c. [How do I remotely debug Tomcat using NetBeans?](#)
3. Other
  - a. [How do I change the monitoring interval for modified resources and application reloading?](#)
  - b. [Official Eclipse IDE Web Tools FAQ for Tomcat](#)

## Answers

### Hacking

#### How do I start hacking Tomcat in Eclipse?

Briefly:

```
$ git clone https://github.com/apache/tomcat.git
(or whatever branch you want: clearly, this would be better
to do directly from within Eclipse but it's easier to describe
as a command)

$ cd tomcat

$ echo "base.path=/path/to/where/tomcat/can/put/its/3rd-party/libs" > build.properties

$ ant ide-eclipse
```

Then, in Eclipse, go to Preferences | Java/Build Path/Classpath Variables and set the following variables:

```
ANT_HOME=path to your Ant install (where lib/ant.jar can be found)
TOMCAT_LIBS_BASE=[whatever you set base.path to above]
```

If you look in (project root)/res/ide-support/eclipse/java-compiler-errors-warnings.txt, you'll see a set of compiler warnings and import organization rules that you will have to set up manually in your project. If you set those up properly, you are more likely to submit cleaner patches.

A full explanation for Tomcat in Eclipse can be found here:

- Tomcat 9.0.x: [https://tomcat.apache.org/tomcat-9.0-doc/building.html#Building\\_with\\_Eclipse](https://tomcat.apache.org/tomcat-9.0-doc/building.html#Building_with_Eclipse)
- Tomcat 8.5.x: [https://tomcat.apache.org/tomcat-8.5-doc/building.html#Building\\_with\\_Eclipse](https://tomcat.apache.org/tomcat-8.5-doc/building.html#Building_with_Eclipse)
- Tomcat 7.0.x: [https://tomcat.apache.org/tomcat-7.0-doc/building.html#Building\\_with\\_Eclipse](https://tomcat.apache.org/tomcat-7.0-doc/building.html#Building_with_Eclipse)

### Debugging

#### How do I configure Tomcat to support remote debugging?

The short answer is to add the following options when the JVM is started:

```
-Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n
```

There are a number of ways you can do this depending on how you normally start Tomcat:

- If you are using **shell scripts** to start Tomcat, start it with the following command:

```
catalina jpda start
```

It will start Tomcat so that a remote debugger can be connected to port 8000.

The above mentioned options can be provided by setting certain environment variables. See the comments at the top of `catalina.sh` or `.bat` file for details.

For example, the port number and JPDA transport implementation can be set with `JPDA_ADDRESS=8000` and `JPDA_TRANSPORT=dt_socket`.

- If you run Tomcat using a **service wrapper** (e.g. [Apache Commons Daemon](#)), add the above JVM options before any other JVM options. Check the documentation for the service wrapper to determine how to set JVM options.
- If you start Tomcat from within an IDE, check the documentation for the IDE to determine how to set the required JVM options.

The port does not need to be set to 8000, it may be any value appropriate for your system.

If you need to debug Tomcat startup or the auto-deployment of an application (where a breakpoint set in a debugger would be reached before you have time to connect with the debugger), then set the parameter `suspend=y` instead of `suspend=n`. Doing this will cause the JVM to pause execution very early in the launch process and wait until a debugger is attached before proceeding.

Whilst this is very useful in development it should not be used in production because of both security and performance implications.

## How do I remotely debug Tomcat using Eclipse?

This answer assumes that you have a project set up and have some idea of what you are doing in this respect. If not then that is really outside the scope of this topic and you need to go to [eclipse.org](#) and read up on how to use your IDE, and maybe practice a little bit before you come back to this. We are also going to assume that you have some idea of what a debugger is and how to use one.

- Make sure that Tomcat is started in remote debugging mode as described above and that your app is deployed.
- Make sure that you have the sources for the code that you are trying to debug in your IDE. For the libraries and for Tomcat itself you can "attach" the sources to the jar files: open a class file and then click "Attach Source..." button.
- If you have a servlet or something, set a breakpoint where it is sure to hit on the next request.
- Go to "Run->Debug Configurations...". Click on "Remote Java Applications", then click "New". Type in the title. Note that port 8000 from the Tomcat instructions. Save and run.
- Eclipse will connect to the JVM that Tomcat is running under. Wow, that was easy!

Now go type the url to submit to your servlet or whatever in your browser. Boom you hit the breakpoint right? Have fun!

## How do I remotely debug Tomcat using NetBeans?

This answer assumes that you know how to work with a NetBeans Project, and also how to use the NetBeans debugger. If not, please go to <http://www.netbeans.org/kb/using-netbeans/40/debug.html> and read up on how to use NetBeans and its debugger.

Starting with Tomcat trunk revision 1484409, the Tomcat source includes Ant tasks to configure your source directory as a NetBeans Free-Form Project. After you have successfully run Ant with the default build target (*deploy*), you can then run the *ide-netbeans* target. This task will configure your sandbox copy of Tomcat so the source can be inspected, maintained and debugged under the NetBeans IDE. You should follow the detailed instructions in the README.txt file that will have been installed in your new nbproject directory.

Make sure that Tomcat is started in debug mode as described above, that your application is deployed, and that the sources are all defined as resources in your application. If you have a servlet or JSP file, set a breakpoint where you think a problem might be occurring. Go to "Run->Attach Debugger". A dialog pops up to let you specify the following options:

- Debugger: JPDA Debugger
- Connector: SocketAttach
- Host: The IP address of the host your Tomcat installation is running on (127.0.0.1 if it is your local machine).
- Port: The port of your Tomcat debugging interface, which should be 8000 if you've followed the instructions above.

When you press OK, you have a debugging connection very similar to local debugging.

Note that NetBeans has a second option – you can debug JSP files and servlets locally using a Tomcat server that is bundled with the IDE. When you debug a JSP file or servlet in the IDE, the bundled Tomcat server automatically starts in debug mode, and the debugger connects to it.

The Tomcat NetBeans targets have not yet been back-ported to Tomcat 7. You can copy the files from your copy of the trunk and they should only require minimal editing of paths to work with older versions of Tomcat.

## Other

### How do I change the monitoring interval for modified resources and application reloading?

Monitoring interval for application reloading is controlled by the `backgroundProcessorDelay` property on `Context` element or on its parent containers: `Host` and `Engine`. See [Tomcat Configuration Reference](#) for details. By default there is a single background processing thread that is run by `Engine`. See its [configuration](#) for the default delay value.

Interval that controls reloading of the changed JSP pages is set in the [Jasper configuration](#) in `web.xml`.

## Official Eclipse IDE Web Tools FAQ for Tomcat

Eclipse IDE has support for development of Web applications and running them on Apache Tomcat. This support is provided by [Eclipse Web Tools Platform Project](#). An easy way to get Web Tools is to download "for Java EE Developers" edition of [Eclipse IDE](#).

The Web Tools project has a FAQ page.

- [WTP Tomcat FAQ](#)
- In [Eclipse Help](#) see "Web Tools Platform User Guide" > "Using the server tools" > "Testing and publishing on your server"