

Configuring a remote Apache HTTP server

 [Configure log level](#)

 [Administering the Apache Geronimo Server](#)

[Configuring the JMS server](#) 

At the time of writing this article the **Apache mod_jk Configuration** portlet provided in the Geronimo Administration Console presented some inconsistencies recognizing the already installed web applications. As a workaround this article will cover two alternative ways to configure a remote Apache HTTP server.

Apache Geronimo comes in two flavors depending on the Web container you decide to use. You may choose between the Tomcat or Jetty distributions. If you want to place a HTTPd in front of Geronimo you will have to configure it in such way it can forward client requests to Geronimo.

As we already mentioned, this article will cover two different ways on how Apache HTTPd can be configured to forward client requests to Geronimo, this is configuring it as a reverse proxy using the "built-in" mod_proxy module or configuring the Jakarta Tomcat Connector mod_jk available with the Apache Tomcat source.

When configuring a reverse proxy in the Apache HTTPd, that configuration will work for any distribution of Geronimo (Tomcat or Jetty) you may have without the need for any additional configuration on the Geronimo side.

When configuring the Jakarta Tomcat Connector, additional configuration will be required. There will be a number of new terms that will be addressed as needed but keep in mind the scope of this article is to configure a two-tier scenario with a remote HTTPd.

Based on these two alternatives, this article is organized in the following sections:

- [Configure Apache HTTPd as a reverse proxy \(mod_proxy\)](#)
- [Configure Apache HTTPd with Jakarta Tomcat Connector \(mod_jk\)](#)

In order to set a common ground for this configuration example, here is a list with the versions tested:

- Apache Geronimo v1.2
- Apache HTTPd v2.2.3
- mod_jk-apache-2.2.3.so
- Windows XP SP2