



FUEGO 5.5 WORK PORTAL SINGLE-SIGN-ON WITH A WINDOWS DOMAIN (Using Tomcat 5)

Fernando Dobladez
ferd@fuego.com

December 30, 2005

Abstract

This document describes a way of configuring the Fuego Work Portal running in Apache Tomcat integrated with Microsoft's IIS in order to achieve Single-Sign-On with a Windows Domain.

1 Introduction

The goal is to achieve Single-Sign-On functionality in the Fuego Work Portal.

When a user is working from a Windows workstation and is logged into the Windows Domain, he/she will be automatically recognized by the Fuego Work Portal application with no need to provide a user ID and password.

Each user must first be created in the Fuego Directory. A customizable error page will be presented to the users that have not been added to the Fuego Directory.

2 How it works

At a high level, the solution implemented works as follows:

- IIS (Microsoft's Web Server) will handle all HTTP traffic from the end users. All requests addressed to the Fuego Portal will be delegated to Tomcat (the Java webapp container hosting the Fuego Portal).
- IIS and IE (Internet Explorer) both support a protocol for identifying the user if already logged into the domain. Basically, IE passes an identification token to IIS, and IIS validates the token and obtains the user ID.
- IIS sits in the middle between the web browsers (IE) and Tomcat. When an authenticated request is accepted and is addressed to the Fuego Work Portal, IIS will inject the user ID into the request and delegate it to Tomcat.
- When Tomcat receives an HTTP request with a user ID, it will pass the ID to the Fuego Work Portal application which will in turn use it to identify the user.

3 IIS Configuration

The Tomcat Connector ISAPI filter needs to be installed in IIS. This filter will allow IIS to delegate the specified URL requests to Tomcat.

The Apache Jakarta website includes documentation on how to install and configure the filter in IIS:

<http://jakarta.apache.org/tomcat/connectors-doc/index.html>

Once installed and working properly, it is important to enable the *Windows Integrated Authentication* option on the IIS site where the connector is configured. This will make IIS do the authentication of the user against the Windows Domain and then inject the user id into the request before delegating it to Tomcat.

4 Tomcat configuration

Tomcat v5.0.x uses the "JK2" connector. Once installed, the following properties need to be added to the `jk2.properties` file:

.../jakarta-tomcat-5.0.28/conf/jk2.connector

```
...
request.tomcatAuthentication=false
request.registerRequests=false
```

Those properties tell the connector not to use Tomcat's authentication, but accept the authentication passed by IIS.

Note: *The JK2 connector is deprecated in the newer Tomcat version 5.5.x (although it is still supported). The JK connector is recommended instead.*

When using JK, the properties are not specified in a separate file as in JK2. Instead, they need to be set as an attribute of the <Connector> tag definition inside Tomcat's `server.xml` file.

5 Fuego Configuration

Since the Fuego Work Portal will not be doing the authentication itself, it needs to be configured for *container-based* authentication.

To achieve this, follow the next configuration steps:

1. Configure the `directory.properties` file of the Portal webapp so that it can create Fuego Directory sessions without asking the users for a password.
2. Switch the authentication servlets of the Fuego Work Portal (modifying the `web.xml` file).
3. Add *participant trust* entries into the Fuego Directory. This allows the Portal to automatically create Fuego Directory sessions for the users without requiring a password.

The following sections explain each step in more detail.

5.1 Configuring `directory.properties`

Three new properties need to be added to the `directory.properties` file of the Fuego Work Portal:

`directory.DIRECTORY_ID.preset.container-auth.jdbc-user` This is the JDBC user that will be used to connect to the Fuego Directory database when using Container-based authentication.

`directory.DIRECTORY_ID.preset.container-auth.jdbc-password` This is the JDBC password for the user specified in the previous property.

`directory.DIRECTORY_ID.preset.container-auth.skip-auth` This property should be set to `true` in order to automatically log the user in without asking for a password.

Example:

```
# Container Authentication Fuego Directory Service Configuration
directory.default.preset.container-auth.jdbc-user=FUEGOFDIADM
directory.default.preset.container-auth.jdbc-password=<encrypt>password
directory.default.preset.container-auth.skip-auth=true
```

Note the optional `<encrypt>` prefix added to the password. If this is specified the Portal application¹ will scramble the password in the file, so that it no longer shows in plain text. After the application starts, the prefix will change to `<crypted>` and the value of the password will be a random-looking string similar to the following:

```
directory.default.preset.container-auth.jdbc-password=<crypted>
UA7jo3Pvnul2sc/NMcxJ3ijnJHzgtte9YrbPr3NkA5wYNm1BbAHAOmLDvlq8vEpQ6fpD8g==
```

5.2 Switch Portal Authentication Servlets

The default servlets of the Fuego Work Portal always authenticate the users against the Fuego Directory using an HTML login form.

The Fuego Work Portal provides another set of authentication servlets that delegate the authentication to the servlet container. These provide the needed behavior for the Tomcat+IIS single-sign-on solution described in this document.

To enable the container-based authentication servlets, the `web.xml` file needs to be changed from the default values:

.../webapps/portal/WEB-INF/web.xml

```
<servlet>
  <servlet-name>
```

¹Or any Fuego application using this `directory.properties` file.

```

    startup
  </servlet-name>
  <servlet-class>fuego.portal.servlet.deploy.SimpleStartup</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
  <servlet-name>
    loginWam
  </servlet-name>
  <servlet-class>fuego.portal.servlet.deploy.SimpleLogin</servlet-class>
</servlet>
...
<servlet>
  <servlet-name>
    logoutWam
  </servlet-name>
  <servlet-class>fuego.portal.servlet.deploy.SimpleLogout</servlet-class>
</servlet>

```

to the following:

.../webapps/portal/WEB-INF/web.xml

```

<servlet>
  <servlet-name>
    startup
  </servlet-name>
  <servlet-class>fuego.portal.servlet.deploy.UserPrincipalStarup</servlet-
    class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
  <servlet-name>
    loginWam
  </servlet-name>
  <servlet-class>fuego.portal.servlet.deploy.UserPrincipalLogin</servlet-
    class>
</servlet>
...
<servlet>
  <servlet-name>
    logoutWam
  </servlet-name>
  <servlet-class>fuego.portal.servlet.deploy.UserPrincipalLogout</servlet-
    class>
</servlet>

```

5.3 Add participant *trusts* to Fuego Directory

The Fuego Directory *trusted* user structure needs to be configured in order to allow the Fuego Work Portal to establish Fuego Directory sessions for the end users without specifying a password.

When the Fuego Directory is implemented on top of a relational database, this structure is stored in the `FUEGO_PARTTRUST` table. The following rows need to be inserted into the table in order to *trust* the Portal jdbc user:

FUEGO_PARTTRUST		
	FUEGO_ID	FUEGO_TRUSTID
1	null	FUEGOFDIADM*
2	admin	FUEGOFDIADM

The first row means that a Fuego Directory Service JDBC User "FUEGOFDIADM" should trust *any* Fuego Participant already authenticated. The * suffix means no double authentication. The value specified for the `FUEGO_TRUSTID` is the one specified in the property `directory.DIRECTORY_ID.preset.container-auth.jdbc-user` on the Fuego Work Portal's `directory.properties` file.

The second row means that a Fuego Directory Service JDBC User "FUEGOFDIADM" should trust Fuego Participant *admin* doing authentication. This is the Fuego Administrator participant (also commonly configured as *root*). The value of `FUEGO_TRUSTID` does *not* have the * suffix as we *do* want to perform authentication for this Fuego participant.