

Server scripts

by SuperBonBon

1. Scripts

Beanshell scripts can be used for some messages send by the server to the client :

- [Login message](#)
- [Logout message](#)
- [Disclamer message](#)
- [Server closed message](#)

JAFS ships a sample scripts file in the *config* directory named *scripts.xml*. Each scripts file can contain multiple scripts that can be used for the server output messages described above.

You will need some Java programmer skills to create new scripts, however no skills are need to use the sample scripts provided with JAFS.

2. Use an existing script

To use a script edit the server configuration file and add a *script identifier string* into the desired message :

2.1. Choose the script you want to use

I.E : I want to output the system date script provided in the scripts.xml file :

```
<script name="sysTime" desc="the current system date">
<![CDATA[
    Date dte = new Date();
    String frmt = "EEE d MMM yyyy HH:mm:ss Z";
    java.text.SimpleDateFormat format = new java.text.SimpleDateFormat( frmt );
    format.format( dte );
    ]]>
</script>
```

Here **sysTime** is the *script identifier string*

2.2. Add the script identifier string in the server configuration file

Edit your server configuration file, and add in the server output message tag your *script identifier string* between brackets with a leading \$ I.E : `${sysTime}`

```
<server xmlns="http://www.sbbi.net/jafs/1.0/jafs-server"
  ...
  code="sample_server" ... >
  ...
  <loginMessage>Hello new client the system time is : ${sysTime}</loginMessage>
  ...
</server>
```

In this example the login message has been choosed to output the result of the `${sysTime}` script

Remember that only the 4 messages mentionned in the [Scripts](#) section of this document are able to use such feature.

2.3. The final result

Here is the final output when the client logs in :

```
Resolving host name localhost...
Connecting to (localhost) -> IP: 127.0.0.1 PORT: 21
Connected to (localhost) -> Time = 0ms
Socket connected waiting for login sequence.
220 Hello new client the system time is : Sun 5 Oct 2003 14:06:40 +0200
USER username
...
```

3. Creating your own scripts

First you'll need to edit the script file and add an new xml `<script>` tag that match the grammar used by the other existing scripts tags.

You can also create your own script file and instruct JAFS to load this file, look at [the service configuration](#) file [scriptload](#) and [scripts](#) for more informations.

Once done, chose an unique *script identifier string* and add it to the *name* script attribute tag

Note:

The *script identifier string* **must** be unique

Server scripts

In the *script* tag body add your code I.E :

```
<script name="helloworld" desc="my first script">
  <![CDATA[
    return "hello world";
  ]]>
</script>
```

You are done, follow the instructions above to output your sample script

This example is very basic, but remember : you can use the entire JAVA API with these scripts, such feature open huge horizon : you can connect to databases using JDBC, connect to remote servers to retrieve data and much more..

Warning:

You can create your own scripts, and uses other people scripts ONLY if they can be considered as "trusted", since a malicious script could **severly damage** your system. SBBI declines all responsibility for such problems and certify that the sample scripts provided are **totally safe to use**.

4. Available JAFS variables in scripts

Some variable that represents JAFS objects are provided within scripts execution scope.

Variable Name	Object Class	Description
server	net.sbbi.jafs.FtpServer	The actual FTP server
upnp-service	net.sbbi.jafs.services.UPNPService	The JAFS UPNP service
stats-service	net.sbbi.jafs.services.StatsService	The JAFS statistics service
sound-notifications-service	net.sbbi.jafs.services.SoundNotificationsService	The JAFS sounds notifications playback service
site-list-service	net.sbbi.jafs.services.SiteListService	The JAFS site list generation service
servers-deployments-service	net.sbbi.jafs.services.ServersDeploymentsService	The JAFS FTP servers deployment service
scheduler-service	net.sbbi.jafs.services.SchedulerService	Service for tasks scheduling
resource-bundles-service	net.sbbi.jafs.services.ResourceBundlesService	Service for internationalization
ratio-service	net.sbbi.jafs.services.RatioService	Service for clients ratio
resource-bundles-service	net.sbbi.jafs.services.ResourceBundlesService	Service for internationalization

newsservice	net.sbbi.jafs.services.NewsService	News list generation service
jmxservice	net.sbbi.jafs.services.JMXService	JMX management service
jmxadapterservice	net.sbbi.jafs.services.JMXAdaptersService	JMX adapters management service
logservice	net.sbbi.jafs.services.LogService	JAFS logging service
hammeringservice	net.sbbi.jafs.services.HammeringService	Service for client that hammer the server
dyndnsservice	net.sbbi.jafs.services.DynDNSService	Service for DYNDns support
dynamicipupdaterservice	net.sbbi.jafs.services.DynamicIPUpdateService	Service for most dynamic IP's update
beanshellservice	net.sbbi.jafs.services.BeanShellService	Service service you're currently using for this scripting capability
banservice	net.sbbi.jafs.services.BanService	Service for banning clients
authrealmservice	net.sbbi.jafs.services.AuthRealmService	Service for authentication realms support in JAFS
ftpserverservice	net.sbbi.jafs.services.FtpServerService	The main JAFS service

Example of the *server* variable use :

```
<script name="maxConn" desc="the maximum number of connections">
  <[CDATA[
    server.getFtpServerConfig().getMaximumSessionsNumber();
  ]]>
</script>
```

Take a look at the JAFS API documentation for more informations about those objects capabilities.

5. Scripts Programming errors

During each server startup JAFS executes all the scripts and immediatly stops if a programming error is found. This prevents to start the server with invalid scripts and allows you to do some debug.