

Contexts and BBJSP

Using Contexts to Build a Better Web App

by Richard Stollar

If you want to set up a context for a new BBJSP application then you are in the right place. In this article, we'll take a quick look at creating and configuring a new context to hold your BBJSP project.

Overview

You can use contexts to isolate parts of your browser-based component. This also gives you control over who can access the different parts. BBJServices is capable of serving up content for many domains or subdomains and one way to manage this is to configure a context for each domain or subdomain. Each context can deploy applications and fixed content like HTML, CSS, and JavaScript files, as well as your BBJSP components.

If you are unfamiliar with configuring contexts in general, then you can take a look at an article from our 2014 issue of the Advantage magazine called "[Don't Put All of Your Jetty Eggs in One Context](#)".

BBj 16.00 introduced the ability to create contexts for holding BBJSP applications. Since then, we have improved and streamlined the context configuration in Enterprise Manager (EM), so if you have a version earlier than 16.20 we strongly suggest upgrading your BBj installation.

Activate the Built-in BBJSP Context

When you start using BBJSP you will need to configure a context to store the pages and tell the BBJSP framework about the components of your application; we'll get onto that soon enough.

Our first step will be to enable the default BBJSP context. To do this, open EM, select the Context Configuration node, then click on BBJSP list item as shown in **Figure 1**. After the configuration panel opens in the right-hand pane, check the Auto Start checkbox and Save the configuration.

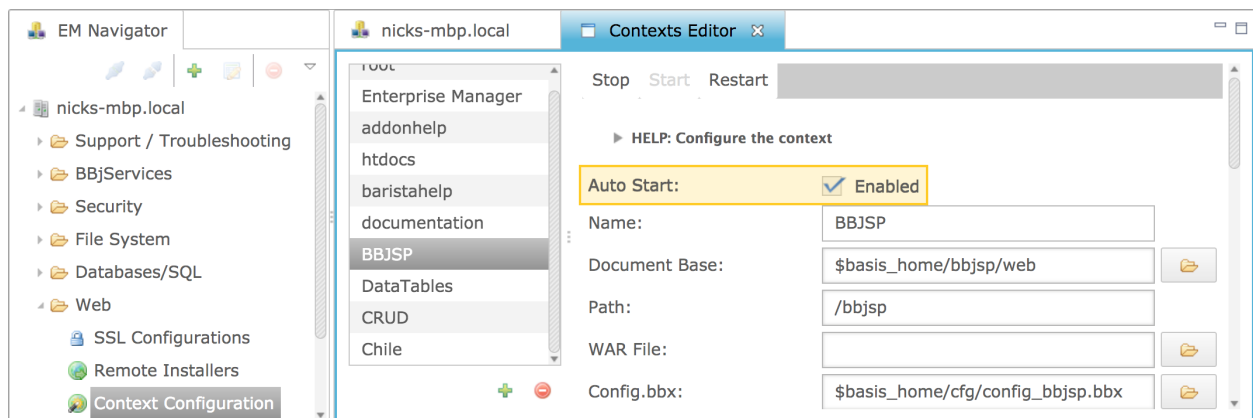


Figure 1: Enable the BBJSP context in Enterprise Manager

For those of you who prefer to edit the configuration files for yourself, you can manually edit the `<bbjhome>/cfg/jetty.xml` file that contains a configuration section similar to **Figure 2**. To enable the default BBJSP context, change the status setting from "disabled" to "enabled".

```
<context config="$basis_home/cfg/config_bbjsp.bbx
docbase="$basis_home/bbjsp/web" name="BBJSP"
path="/bbjsp"
status="disabled">
  <welcome-file>index.bbjsp</welcome-file>
  <bbjsp-servlet class="TestServlet"
mapping="/test"
source="TestServlet.bbj"/>
</context>
```

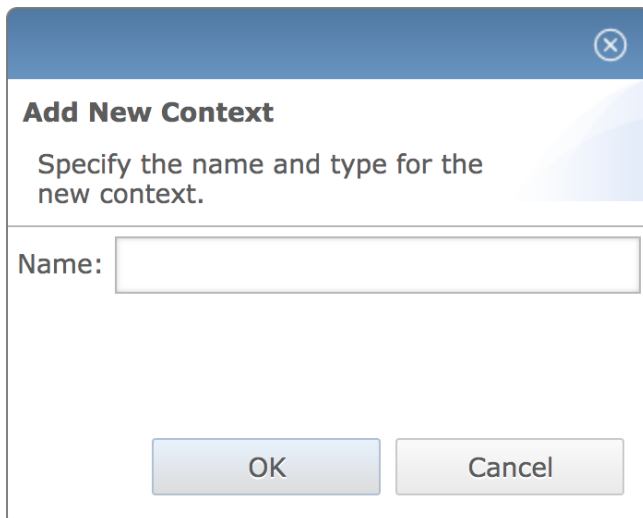
Figure 2: The preconfigured BBJSP context entry in the `jetty.xml` file

After enabling the context you will need to restart BBJ Services. Once that's done you can browse to <http://localhost:8888/bbjsp/index.bbjsp> and, assuming everything's working as it should, you'll see the BBJSP documentation and examples installed with BBJ.

Configuring a New Context for Your BBJSP Application

Let's look now at making a clean configuration that's needed for the upcoming [create, read, update and delete](#) (CRUD) application, covered in a future article, which uses the ChileCompany database.

Through Enterprise Manager, you need to select the Web folder's Context Configuration node and then click the **plus** icon to add a new context. EM displays a popup window where you can enter the name for the new context. We'll use **Chile** but you can choose whatever you want.



The image shows a dialog box titled "Add New Context" with a close button in the top right corner. Below the title, it says "Specify the name and type for the new context." There is a text input field labeled "Name:" which is currently empty. At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Figure 3. Add a new context for the BBJSP application

The next step will be to complete the remaining configuration data for the context. Selecting the newly-created **Chile** context in the list brings up its configuration panel in the right-hand pane shown in **Figure 4**.



The image shows the "Contexts Editor" panel. On the left, there is a list of contexts: root, baristahelp, CRUD, BBJSP, documentation, addonhelp, GoldMine, Chile (highlighted), and htdocs. Below the list are plus and minus icons. On the right, there are configuration fields for the selected context:

- Enabled:
- Name:
- Document Base: 
- Path:
- WAR File: 
- Config.bbx: 
- Encoding:
- SSCP:

Figure 4: The Contexts Editor panel

The Document Base is the location where you will store all the files for your context. This includes all the BBJSP pages plus some context-specific configuration files relating to the BBJSP framework. You should enter a folder or click the folder icon to choose an existing folder or create a new one as depicted in **Figure 5**.

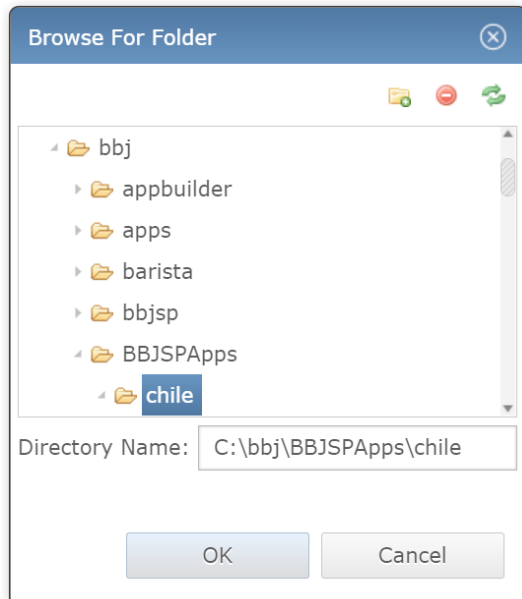


Figure 5: Choose the Document Base folder to store the web content

There are still some more fields that need to be completed before the context configuration is ready, such as specifying which config.bbx file to use for code execution. The *Path* part of the URL is also required, and we'll use `/chile` so that our final BBJSP URL will be `http://localhost:8888/chile`. That leaves us with one last task to complete the essential parts of the BBJSP configuration: clicking the *Update BBJSP Tag Libraries* button on the context configuration page. Doing so each time a new context is added provides essential functionality that you'll rely on in your pages.

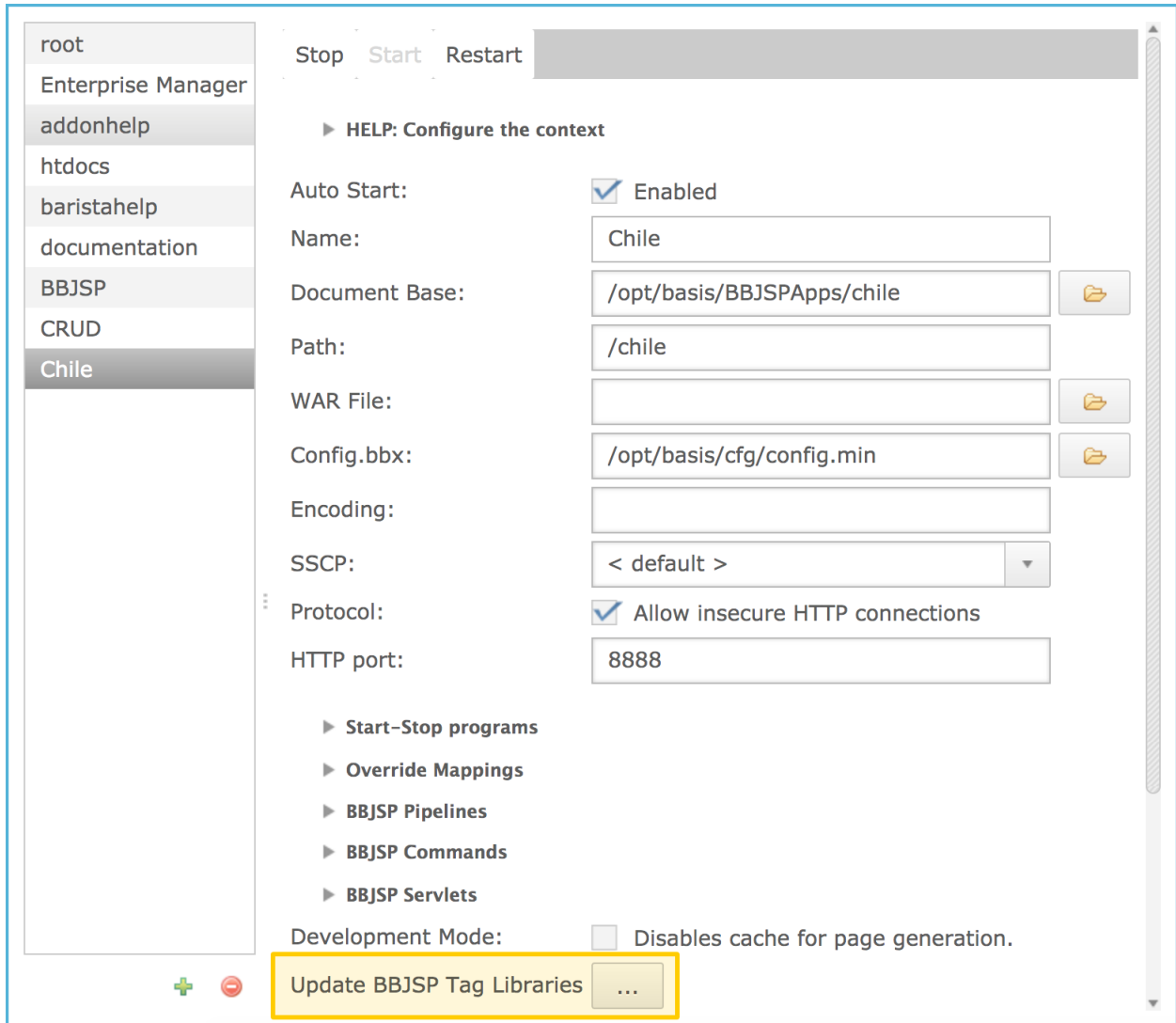


Figure 6: The completed Chile context configuration with option to Update the BBJSP Tag Libraries

Now you need to restart BBJ Services to complete the configuration of the new context. If *Auto Start* was enabled then the context will now be up and running.

Let's see it working then

So now we have a configured BBJSP context that's ready to receive your files. For a normal context, we would now create `index.html` and take it from there but for a BBJSP context we need to create the `index.bbjsp` file. **Figure 7** shows a short `index.bbjsp` file that we saved in the *Document Base* folder we specified in **Figure 5**.

```
<!DOCTYPE html>
<html>
  <body>
    <%
      #PageContext!.write("<h1>BBJSP is running</h1>")
    %>
  </body>
</html>
```

Figure 7: The `index.bbjsp` file contents

To see our BBJSP application in action, we point our browsers to <http://localhost:8888/chile/index.bbjsp>. This results in the success page shown in **Figure 8**.



Figure 8: The browser output from our test BBJSP page

Summary

As you have read, creating a context for building a BBJSP application is rather easy. Simply follow the steps we've outlined and you'll very quickly have the new context ready. Now you're ready to get on with deploying your application or work through building the CRUD application outlined in a forthcoming article. Thanks for reading.



Check out the example available for download at links.basis.com/16code