



Migrating BBjspServlets to BBxServlets

By Alex Anastassatos

BASIS Advantage 2023

In BBj 21.00, BASIS introduced BBxServlets to succeed BBjspServlets. On the surface, BBxServlets were designed to perform the same functions as BBjspServlets; however, differences under the hood make BBxServlets quicker and more efficient at handling large numbers of requests and license check-outs. For this reason, and perhaps more importantly because BBjspServlets are no longer supported, you will want to consider migrating your BBjspServlet-based applications to BBxServlets. One of the core design goals of BBxServlets was to offer ease of migration, and with some enhancements made in BBj 22.01, this process is simpler than ever. This article covers the syntax changes that you can make to your BBjspServlets to convert them to BBxServlets, as well as how to easily re-configure your servlet through both the `jetty.xml` file and the Admin API.

Migrating the Service Method

Let's start with the migration of your servlet's heart and soul: the service method. The service method is located in the servlet's source file and is the

entry point for your servlet's execution. For BBjspServlets, the service method must be called `service()` for BBjServices to identify what method to call when a servlet is executed. This hard-coding restriction is lifted with the BBxServlet; the service method name for a BBxServlet is set at configuration time, allowing you to use any name for the service method. If no method name is given at configuration time, it will default to "service" in order to make migration simpler.

It is of chief importance that the functionality of the service method remains intact while migrating. Therefore, the goal is to change as little of the actual logic as possible when migrating a BBjspServlet to a BBxServlet. BBjspServlets and BBxServlets both only utilize four distinct objects for storing/retrieving request and response data. For a BBjspServlet, these objects are the [BBjspServletContext](#), [BBjspWebRequest](#), [BBjspWebResponse](#), and [BBjspWebSession](#). The [BBjspServletContext](#) is the sole argument of the service method and is used to retrieve the other servlet objects. Similarly, in a BBxServlet, the sole argument for the service method is the [BBxServletContext](#), which contains the other three equivalent objects: [BBxServletRequest](#), [BBxServletResponse](#), and [BBxWebSession](#).

For ease of migration, the various methods used in the BBjsp and BBxServlet service objects are identical, making migration as simple as searching and replacing the object declarations in your service method. This is illustrated in the figure below, where all that was changed were the BBjspServletContext, BBjspWebRequest, BBjspWebResponse, and BBjspWebSession.

```
1 2
3=CLASS PUBLIC BBjspToBBxServlet
4= METHOD PUBLIC VOID service(BBjspServletContext context!)
5
6     Declare BBjspWebRequest request!
7     Declare BBjspWebResponse response!
8     Declare BBjspWebSession session!
9
10    request! = context!.getRequest()
11    response! = context!.getResponse()
12    session! = context!.getSession()
13
14
15    out! = response!.getOutputStream()
16    out!.write("Hello World")
17 METHODEND
18 CLASSEND

1 2
3=CLASS PUBLIC BBjspToBBxServlet
4= METHOD PUBLIC VOID service(BBxServletContext context!)
5
6     Declare BBxServletRequest request!
7     Declare BBxServletResponse response!
8     Declare BBxWebSession session!
9
10    request! = context!.getRequest()
11    response! = context!.getResponse()
12    session! = context!.getSession()
13
14
15    out! = response!.getOutputStream()
16    out!.write("Hello World")
17 METHODEND
18 CLASSEND
```

Figure 1. A Simple BBjspServlet Converted to a BBxServlet

Re-Registering the Servlet

With the service method successfully migrated, the next step is to re-register the source file with BBjServices as a BBxServlet. If you are unfamiliar with how to configure a servlet, this [BBx Servlet Tutorial](#) goes over the process in detail. When it comes to a pre-existing BBjspServlet,

however, you can simply alter the servlet's configuration element in the `jetty.xml` and restart BBjServices to quickly register the servlet as a BBxServlet.

NOTE: In general, avoid editing the `jetty.xml` file directly, and make your servlet configuration changes through the Enterprise Manager. Migrating a BBjspServlet to a BBxServlet is a special case, and as of the writing of this article editing the `jetty.xml` file directly is the best way to efficiently perform the migration.

To edit the servlet's configuration element, open your `jetty.xml` file located in the `"cfg/"` folder of your BBj home directory. The `jetty.xml` file stores the configuration information for everything Jetty-related, including the configuration data for servlets. If you search for the path of the source file you just converted into a BBxServlet, you will find a configuration node that looks something like the code below.

```
<bbjsp-servlet class="ServletClass" mapping="/testMapping"
source="C:\Path\To\Source\File.bbj"/>
```

If you are running BBj 22.01 or higher, you can simply replace the `"bbjsp-servlet"` tag with `"bbx-servlet"` as shown in the code below and restart BBjServices. This will re-register the servlet and allow you to access it at the mapping that it was configured to run on.

```
<bbx-servlet class="ServletClass" mapping="/testMapping"
source="C:\Path\To\Source\File.bbj"/>
```

NOTE: If you are running a version of BBj before 22.01, you will also need to add an additional attribute and value for the method name to the configuration node. This will be `"methodName='service'"` if you didn't choose to also change the method name in the source file. If you did choose to change the servlet's method name during the migration process, you must use that name in place of `"service."`

```
<bbx-servlet class="ServletClass" mapping="/testMapping"
source="C:\Path\To\Source\File.bbj" methodName="service"/>
```

Migrating Admin API Configuration Code

If you are dealing with pre-configured BBjspServlets, the simplest way to convert the configuration is in the `jetty.xml` file. However, if you have a BBj program that configures the servlet dynamically through the Admin API, you will have to make a few changes to that code as well. Let us take a look at a snippet of simple code that configures a BBjspServlet with the Admin API.

```

Declare BBJettyContextConfiguration context!
context! = BBJAPI().getAdmin(ADMIN_USERNAME, ADMIN_PASSWORD).getJettyServerConfig().getCustomContext("root")

REM Remove servlets in case they already exist
IF (context!.getBBjspServlet("/bbjspServlet") <> NULL())
    context!.removeBBjspServlet("/bbjspServlet")
FI

Declare BBjspServletConfiguration bbjspServletConfig!
bbjspServletConfig! = context!.addBBjspServlet("BBjspServletClass", "/bbjspServlet", "/PATH/T0/BBJSP/SERVLET")

```

Figure 2. *BBjspServlet Configuration Code Through the Admin API*

Servlet configuration is done through the [BBJettyContextConfiguration](#) object, which has various methods that allow you to add, remove, and retrieve a servlet's configuration. For BBjspServlets, these methods are [addBBjspServlet](#), [getBBjspServlet](#), and [removeBBjspServlet](#). Likewise, the Admin API can interface with BBxServlet configurations with [addBBxServlet](#), [getBBxServlet](#), and [removeBBxServlet](#). BBjspServlets and BBxServlets use data structures to contain their configuration data called [BBjspServletConfiguration](#) and [BBxServlet](#), respectively. The first step for converting the above sample of code would be to search and replace any declarations of [BBjspServletConfiguration](#) with [BBxServlet](#) in order to ensure that you are working with the BBxServlet configuration object when you add your servlet.

Now that you have the configuration objects changed, you can simply do a case-sensitive search for the string "BBjspServlet" and replace it with the string "BBxServlet" to convert the rest of the methods that are adding, removing, or retrieving a BBjspServlet configuration. You may also want to change the variable names to reference BBxServlets instead of BBjspServlets, but that is not necessary to get the code to run. Once you have made these replacements, the configuration program will look like the figure below.

```

Declare BBJettyContextConfiguration context!
context! = BBJAPI().getAdmin(ADMIN_USERNAME, ADMIN_PASSWORD).getJettyServerConfig().getCustomContext("root")

REM Remove servlets in case they already exist
IF (context!.getBBxServlet("/bbxServlet") <> NULL())
    context!.removeBBxServlet("/bbxServlet")
FI

Declare BBxServlet bbxServletConfig!
bbxServletConfig! = context!.addBBxServlet("BBxServletClass", "/bbxServlet", "/PATH/T0/BBX/SERVLET")

```

Figure 3. *BBxServlet Configuration Code Converted from the BBjspServlet Code in Figure 5*

If you are running BBJ 22.01 or higher, this is all you need to do to migrate your BBjspServlet code to use BBxServlets instead!

Additional Migratory Steps for BBj < 22.01 only

If you are running a version of BBj prior to 22.01, then you will need to perform additional steps or you will run into some errors after following the simple steps above to convert your Admin API code. Here are the additional migratory steps:

addBBjspServlet arguments incompatible with addBBxServlet

One of the issues you will encounter in versions less than 22.01 is that the argument patterns for addBBjspServlet are in a different order than those in addBBxServlet, as illustrated below.

BBjspServlet

```
addBBjspServlet(classname, mapping,  
source)
```

```
addBBjspServlet(classname, mapping,  
source, config)
```

BBxServlet

```
addBBxServlet(mapping, source, classname,  
method)
```

```
addBBxServlet(mapping, source, classname,  
method, config)
```

```
addBBxServlet(mapping, source, classname,  
method, config, enabled)
```

When migrating the addBBjspServlet method to addBBxServlet the classname must be moved to be the third parameter. Additionally, BBxServlets allow you to use a service method with a name other than "service," unlike BBjspServlets. For this reason, you must add the string "service" as the fourth argument in addBBxServlet methods.

Differences in method names in BBjspServletConfiguration and BBxServlet

The methods used to interact with the [BBjspServletConfiguration](#) and [BBxServlet](#) are slightly different. Illustrated below are the method names in BBjspServletConfiguration and their counterparts in BBxServlet.

BBjspServletConfiguration

```
addParam(String name, String  
value)
```

```
clearParams()
```

BBxServlet

```
addParameter(String name, String  
value)
```

```
clearParameters()
```

<code>getConfig()</code>	<code>getConfigFile()</code>
<code>setConfig(String configFile)</code>	<code>setConfigFile(String configFile)</code>
<code>getSourceName()</code>	<code>getSourceFile()</code>
<code>setEnabled(boolean enabled)</code>	<code>enable()/disable()</code>

If you receive errors stating that a method does not exist in the BBxServlet configuration object, you will want to search and replace the configuration methods listed above.

All these differences were eliminated in BBj 22.01 and higher in order to make the migration process simpler.

Summary

In summary, migration of a BBjspServlet to a BBxServlet begins with the service method, which is the method executed when the servlet is called. Then you can simply change the [BBjspServletContext](#), [BBjspWebRequest](#), [BBjspWebResponse](#), and [BBjspWebSession](#) objects to their BBxServlet counterparts: [BBxServletContext](#), [BBxServletRequest](#), [BBxServletResponse](#), and [BBxWebSession](#).

The next step, as outlined above, is to re-register the BBjspServlet as a BBxServlet through the `jetty.xml` file. If you have a BBj program that dynamically creates BBjspServlets, it can be adapted to create BBxServlets instead by replacing the [BBjspServletConfiguration](#) object declarations with [BBxServlet](#) object declarations, then replacing all instances of the text "BBjspServlet" with "BBxServlet" to change BBjspServlet configuration method calls for BBxServlet method calls. Finally, note that there will be extra steps needed, as outlined above, if you are running a version of BBj earlier than 22.01.

BBxServlets provide an opportunity to easily implement new servlet-based applications, but you can also use them to improve your existing servlets. Compared to BBjspServlets, BBxServlets are easier to work with, are more performant and efficient, and will benefit from ongoing improvements and support from BASIS. On top of that, migrating your servlet functionality to BBxServlets has never been easier, thanks to recent updates that specifically focus on improving the migration process. With the instructions from this article, you have everything you need, so there's no need to wait — upgrade to BBxServlets today!