

=====  
**THIS TOPIC IS DEPRECATED** - Refer to [JNLP Applications](#)  
=====

## JNLP Basics for AddonSoftware

### Introduction

The BBJ Enterprise Manager for BBJ version 10.00 and higher contains a JNLP Configuration tool. JNLP files are required to run BBJ via Java Web Start. Several pre-built JNLP configurations are available in the Demos area, including one for AddonSoftware®.

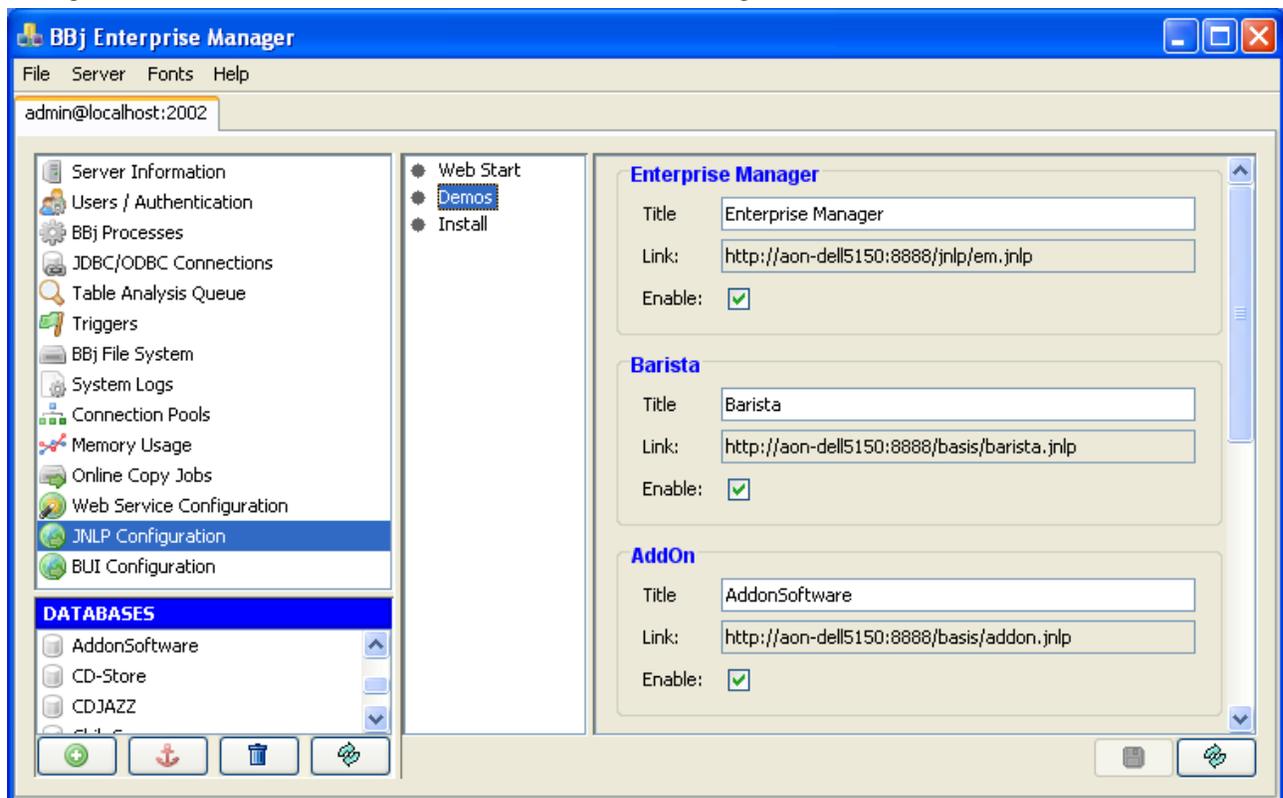


Figure 1: Enterprise Manager JNLP Configuration, showing pre-defined Demos

You can enter the link for the AddonSoftware JNLP in a client browser, or make it a desktop shortcut, and quickly be up and running with Web Start. Note that this default JNLP assumes that clients already have permission to connect to the server.

While the pre-built JNLP works well for initial testing purposes, you may want to create your own JNLP for production. This document provides the basic information required for creating an Addon JNLP.

## JNLP Configuration

Launch Enterprise Manager, and select JNLP Configuration in the left panel. Note that you must be running a Java JDK in order to use JNLP. Select the Web Start node, and click the New JNLP Application button.

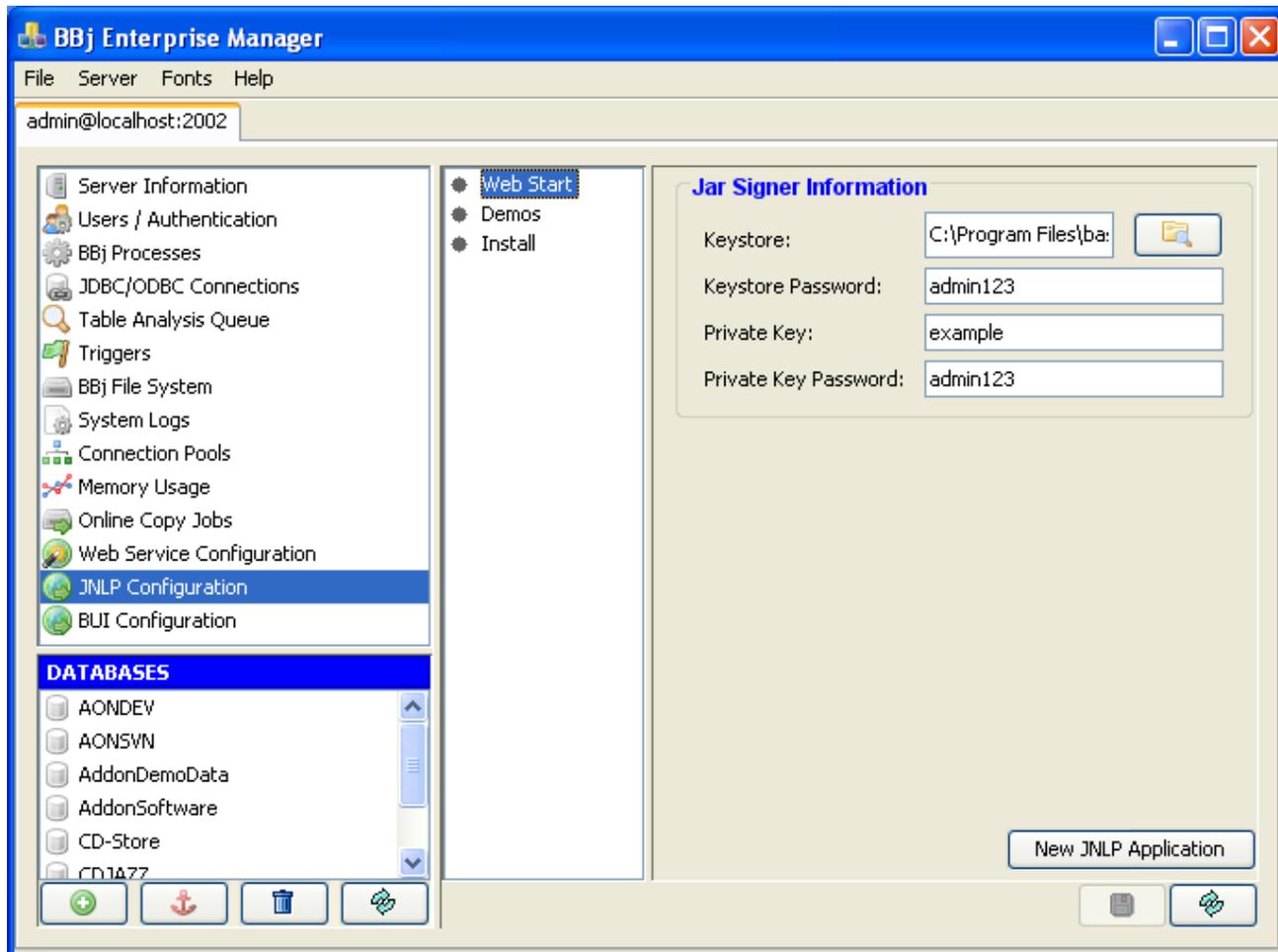


Figure 2: Create your own JNLP using the “New JNLP Application” button.

A new folder tree is created called “newapp.” Click the newapp folder and provide the desired Application Name, Title, Vendor and Short Description, and then click the Save icon. The “newapp” placeholder is replaced with your Application Name, and the Application Link is updated automatically:

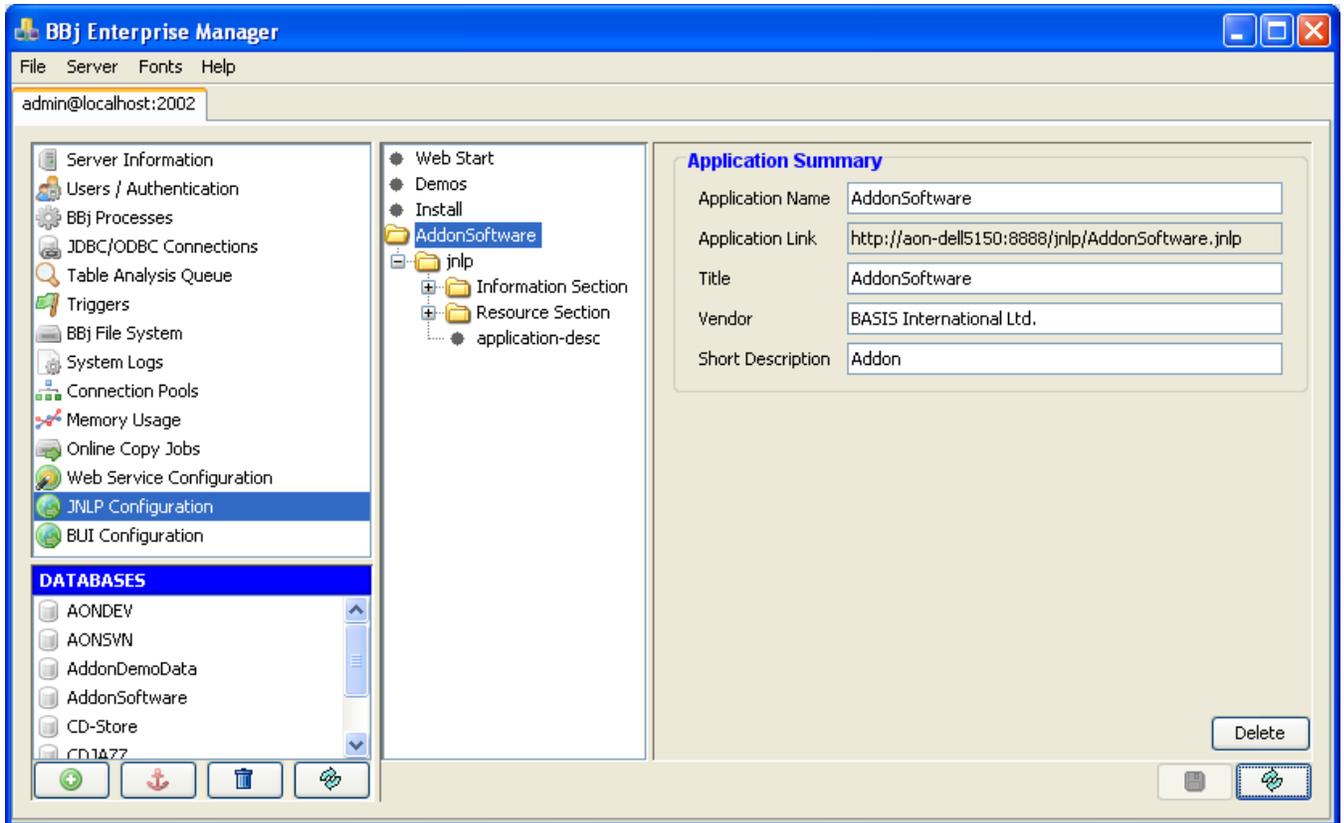


Figure 3: Application Summary information for a new AddonSoftware JNLP Configuration.

You can provide other information in the jnlp, Information Section, and Resource Section nodes as desired, or as your environment dictates. A few configuration settings in this area are discussed that the end of this document, under JNLP Maintenance. For a basic Addon Web Start jnlp, the defaults are typically adequate. See the BASIS JNLP Configuration help page for more information

([http://www.basis.com/onlinedocs/documentation/b3odbc/enterprise\\_manager\\_-\\_jnlp\\_configuration.htm](http://www.basis.com/onlinedocs/documentation/b3odbc/enterprise_manager_-_jnlp_configuration.htm)).

At a minimum, you will need to fill in the application-desc fields as shown in Figure 4.

**BBj Arguments**

Configuration file:	"C:/Program Files/basis/barista/sys/config/enu/barista.cfg"
ClassPath:	
Debug:	<input type="checkbox"/>
Terminal:	T0
User:	
Look and feel:	
INF (JVM Sharing):	aon-dell5150
Load Only:	<input type="checkbox"/>
Local Port:	
Remote Host:	aon-dell5150
Remote Port:	
Secure Client:	<input checked="" type="checkbox"/>
Suppress Splash (quiet):	<input type="checkbox"/>
Working directory:	"C:/Program Files/basis/barista/"
Application (-a):	
Name (-n):	
Program and Arguments:	"C:/Program Files/basis/barista/sys/prog/bar_login.bbj" - aon -IENU;ESP

Figure 4: Minimum required fields in the application-desc window

- Provide the full path to the barista.cfg file in the Configuration file field. If the pathname includes spaces, enclose it in quotation marks.
- Enter the host name in the INF and Remote Host fields.
- Specify the Barista working directory, using quotation marks if needed.
- The Program and Arguments line should contain the full path to the bar\_login.bbj start program (quotations required if there are spaces in the path). If you want to provide your own MDI Title when running Addon, add `- aon` after the start program name. Finally, to limit the login form's language choices, add the lowercase "l" plus the desired language abbreviations, separated by semicolons.
- It is also possible to launch an individual Addon form, rather than the Barista MDI, via Web Start. Simply replace the Program and Arguments line to run the `bax_launch_task.bbj` program rather than `bar_login.bbj`. More information about using `bax_launch_task.bbj` is available [here](#), but a typical setting might look like the one below to launch the Customer Master form. The brackets on the user password tell Barista to look up the password (encrypted on disk) based on the `user_password_id`, so passwords need not be placed "in the clear" on the command line.
  - "C:/Program Files/basis/barista/sys/prog/bax\_launch\_task.bbj" - -username

-p[user\_password\_id] -tARM\_CUSTMAST -mMNT

### Accessing AddonHelp

In order to use the F1 help in Addon, you'll need to add a jar file to your JNLP configuration. Expand the Resource Section node, and under it, expand the default node as well. Click the Add Jar button. A new node called "jar" is placed in the tree. Select it, and enter "AddonHelp.jar" in the Relative URL (href) field. Figure 5 shows this configuration.

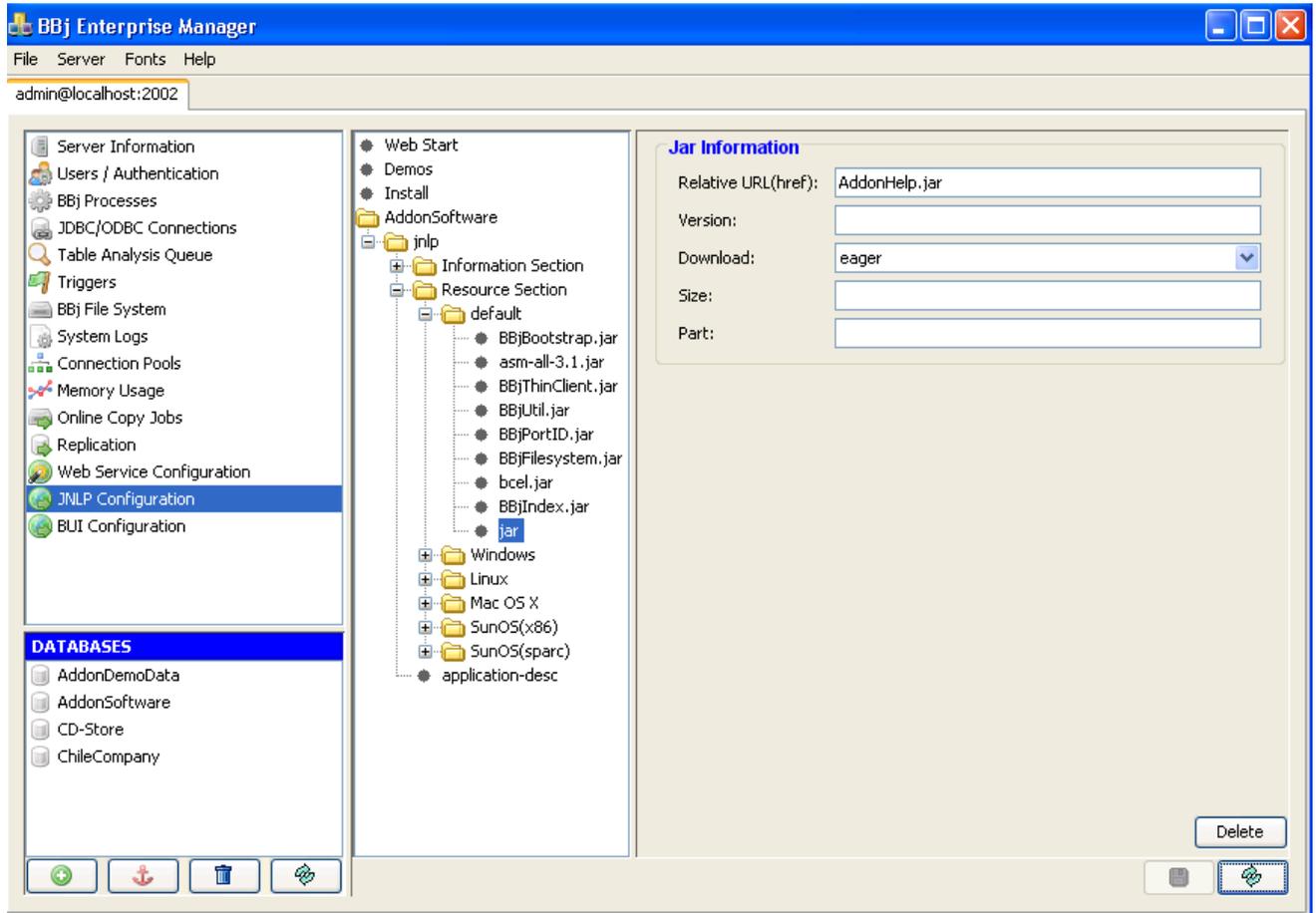


Figure 5: adding the AddonHelp.jar to your jnlp

Note that in this case the AddonHelp.jar being accessed is the one located in the <bbj\_home>/apps/aon/help directory. If you have used the AddonSoftware Installation/Upgrade Wizard utilities (AIUW) to copy Addon to a different location, and you elected to copy the AddonHelp.jar file, then after adding the jar as above, add a reference to that copy of the jar in your BBJ Classpath, as shown here in figure 6:

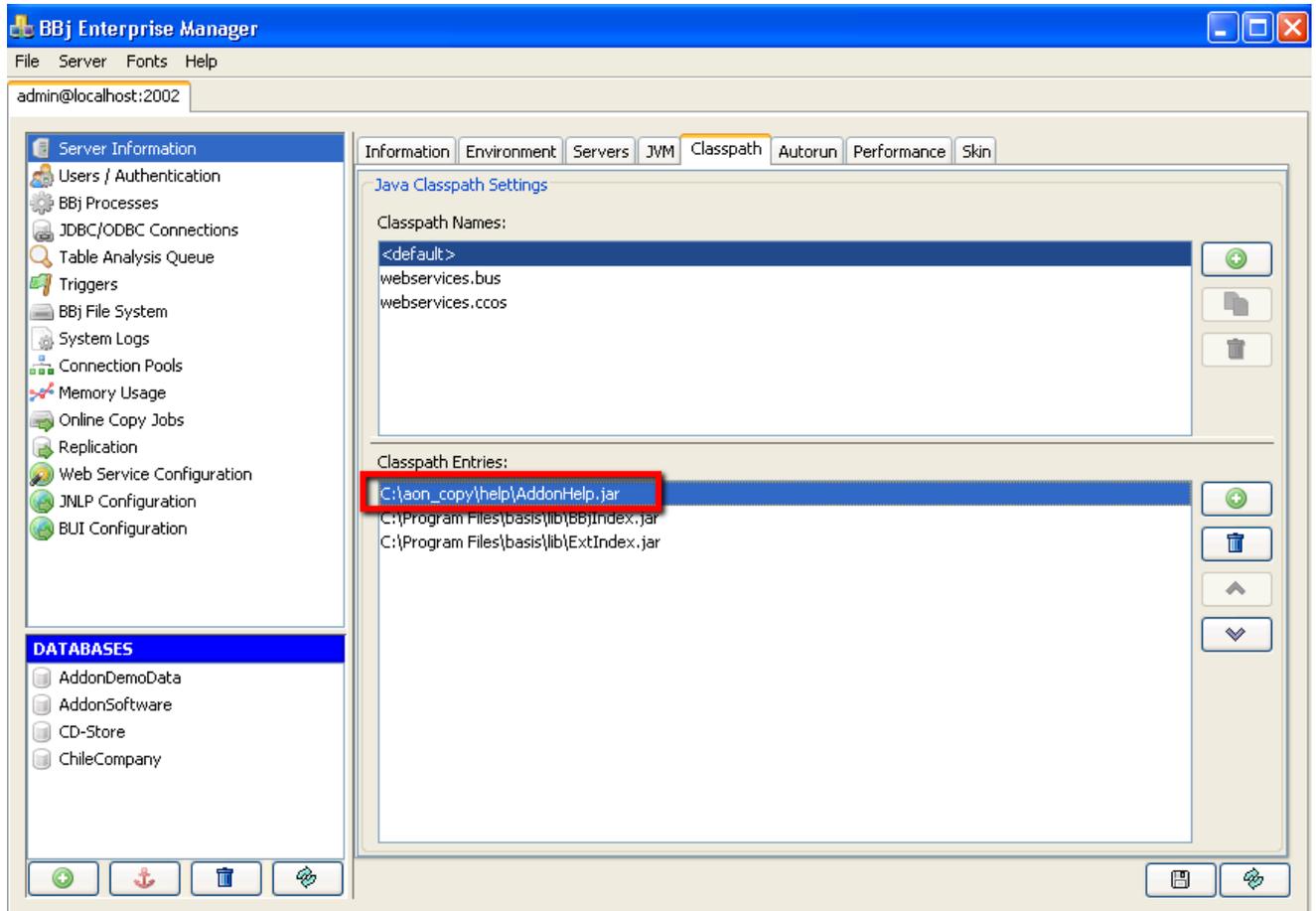


Figure 6: Adding the AddonHelp.jar to the BBJ Classpath for an instance of Addon running in C:/aon\_copy.

### JNLP Maintenance

If changes are made to the JNLP after the initial deployment, it's a good practice to assign or increment the Application Version, under the jnlp node in the JNLP configuration module. This assures that the new version of the JNLP file will be sent to web start clients when they are next invoked.

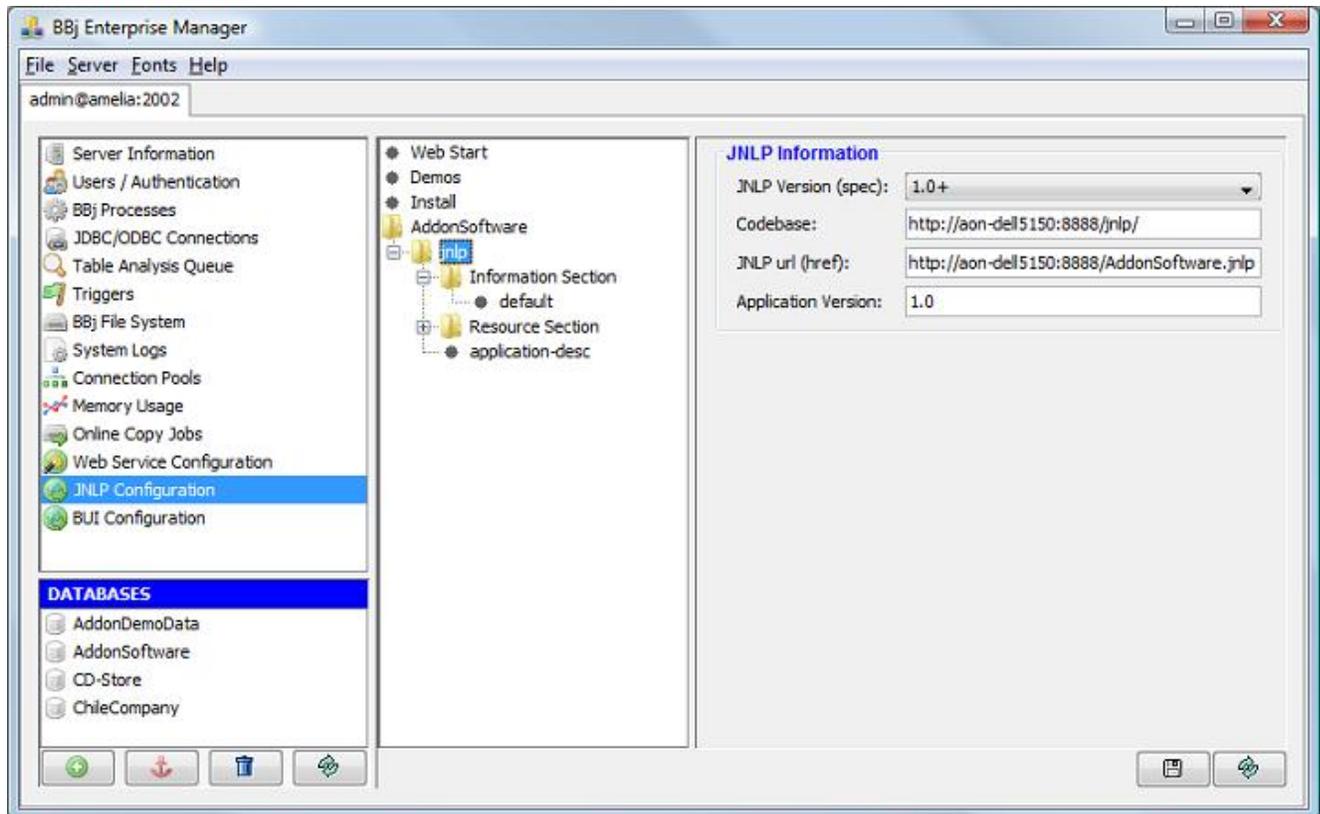


Figure 7: Assigning Application Version in the jnlp node window

If your Addon Web Start application is slow to respond or sluggish, you may need to allocate more memory to the web start client. This is done in the JNLP Configuration module, under the jnlp/Resource Section/default node (see Figure 8. below). Enter the desired value for:

- Initial Heap size (Xms) - the amount of memory in megabytes used by the JVM to launch the application. Example: 64m
- Maximum Heap size (Xmx) - the maximum amount of memory in megabytes allocated to the JVM to run your application. Example: 256m

**Important Note:** Beginning with 1.6.0\_18, Java includes a 'zero setting' option which dynamically allocates memory to the web start application, based on the hardware profile of the client and the memory needs of the application. BASIS highly recommends taking advantage of this setting - this is the default JNLP setting in BBJ 10.03 and above. To configure your JNLP this way, leave both the Xms and Xms fields empty. To ensure that that your web start clients run in a JVM that includes this algorithm (1.6u18 JVM or newer), do the following:

1. Restrict the Java Version to 1.6u18 or above by entering a value in the Java Version field in the Java Information panel (figure 8.) Example: 1.6.0\_18+
2. Enable to the web start client to automatically download a newer JVM by adding the following to the Java URL field: <http://java.sun.com/products/autodl/j2se>

If the web start client is running with a JVM older than 1.6u18, web start will automatically download and install a newer JVM. (Note: It doesn't matter that 1.6u18 is not the latest JVM. The latest JVM available will be downloaded from the URL).

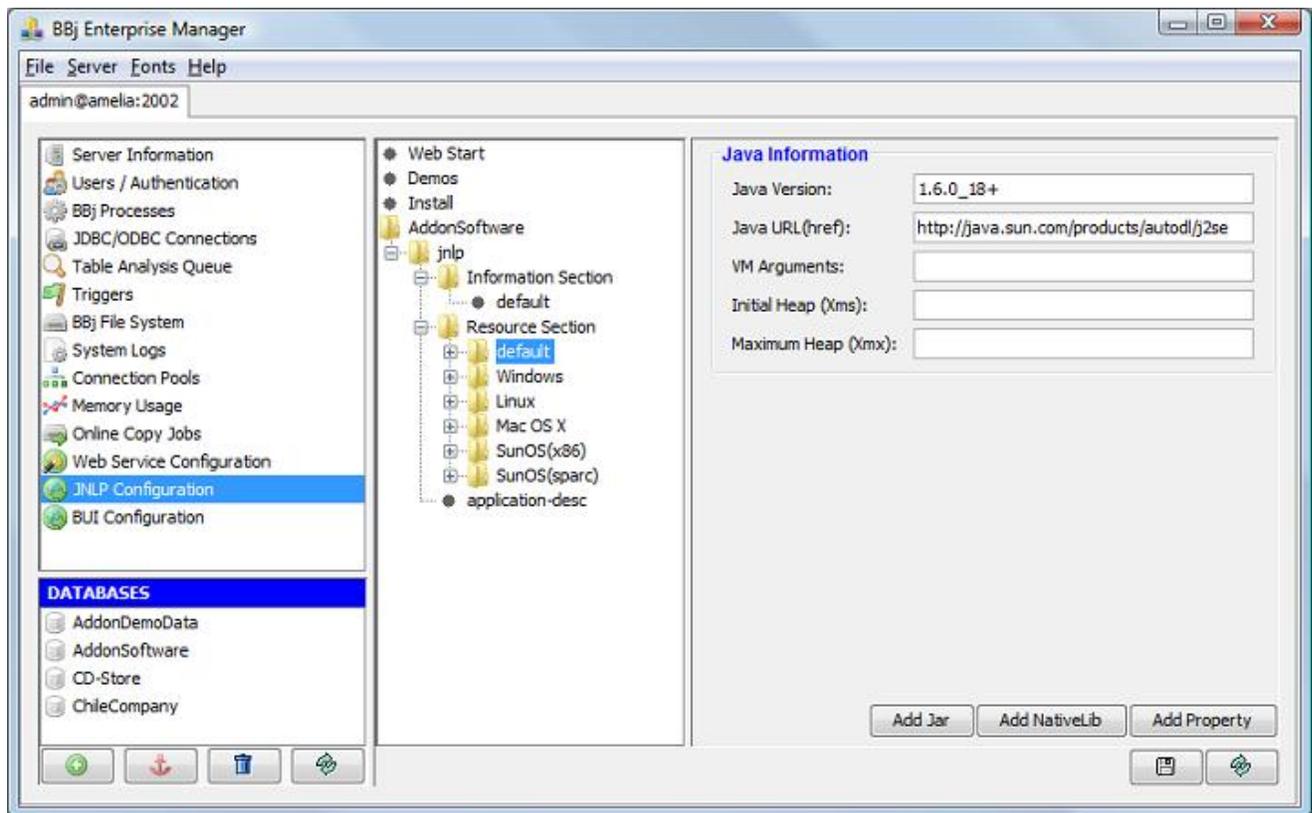


Figure 8: The Java Information window with Java Version Restriction and auto download configured