

Jars, Jars, and More Jars

By Bruce Gardner

Along with the many new features and enhancements of BBj® 7.0 come a new set of jar files (Java ARchives) and a new method to streamline their usage. Several of these new jars correspond to exciting new controls such as bar, line, and pie charts. BBj 7.0 optimizes the search for classes and simplifies the class path by taking advantage of jar indexing.

Q What is jar indexing?

A Jar indexing optimizes the search process of class loaders in applets and Web start applications, reducing the number of jars downloaded to the client machine and greatly simplifying the class path specification. With applets and Web start, it is no longer necessary to list all of the jars needed for your application. On the server side, jar indexing makes it easier to run BBj on operating systems, such as SCO, where limits are placed on the length of the class path.

In BBj 7.0, the main jar **BBjIndex.jar** indexes all of the jar files in the `<bbjhome>lib` directory. As a result, the default class path contains only one jar file: **BBjIndex.jar**.

Q What jars are available?

A In addition to the main jar, **BBjIndex.jar**, 7.0 adds the new jars shown in the table on the next page and combines and/or renames existing jars to reflect their function more correctly.

Q Will the jar changes affect my Web start implementation?

A Yes. In previous revisions of BBj, the **ThirdParty.jar** was a key jar element listed in the JNLP file. BBj 7.0 deprecated the **ThirdParty.jar** so we recommend changing the entry in the JNLP to **BBjUtil.jar**.

Note: In order to minimize the initial impact of this change, the **ThirdParty.jar** is now a copy of **BBjUtil.jar**, which allows most Web start applications to continue to work without a change in BBj 7.0. However, note this change since BASIS will not ship **ThirdParty.jar** in future revisions of BBj.

Applications that take advantage of new BBj 7.0 controls will also need to list the associated jar files. For example, a Web start application that uses the **BBjBarChart** would require a **Charts.jar** element in the application's JNLP file. Similarly, to use the new spell checker feature available in all BBj 7.0 text-based controls (see Interface *TextControl* in online documentation at www.basis.com), a developer would include **SpellChecker.jar** and **SpellCheckerDictionary-xx.jar** as elements in the JNLP file. Where **xx** corresponds to the desired language.

Q Can I take advantage of the new jar indexing feature with my Web start implementation?

A Yes. Web start applications can list the **BBjIndex.jar** as an element to load classes from such unlisted jars as those mentioned on the next page. However, two conditions must be met:

1. The Web start JNLP file must contain a **BBjThinClient.jar** element and a **BBjUtil.jar** element. The **BBjThinClient.jar** must appear as the first element in the application JNLP.
2. Both the server and client systems must have JRE 1.6 or greater installed to take advantage of jar indexing.

continued...



Bruce Gardner
Quality Assurance
Engineer



NEW JAR SET	OLD JAR SET
BBJIndex.jar Charts.jar FontChooser.jar HelpAll.jar JDBCPooling.jar SpellChecker.jar SpellCheckerDictionary-xx.jar (10 jars*)	
ConfiguratorHelp.jar	config.jar
GuiBuilderHelp.jar	guibuild.jar
BBJEM.jar	b3odbc.jar BBJEM.jar BBjXML.jar
BBj.jar	BBj.jar CustomIDE.jar
BasisInstall.jar	BasisInstall.jar
BBjFileSystem.jar	BBjFileSystem.jar
BBjJDBC.jar	BBjJDBC.jar
BBjPlugin.jar	BBjPlugin.jar
BBjPortID.jar	BBjPortID.jar
BBjServer.jar	BBjServer.jar
BBjSQL.jar	BBjSQL.jar
BBjThinClient.jar	BBjBridge.jar BBjThinClient.jar
BBjUtil.jar	BBjUtil.jar
ThirdParty.jar (Duplicate of BBjUtil.jar**)	ThirdParty.jar
bcel.jar	bcel.jar
CustomLookAndFeel.jar	skinlf.jar
PDF.jar	itext.jar
RXTXcomm.jar	RXTXcomm.jar
<p>* Each of these jars provides a spellCheck dictionary for a different language or dialect: SpellCheckerDictionary-am.jar, SpellCheckerDictionary-br.jar, SpellCheckerDictionary-ca.jar, SpellCheckerDictionary-du.jar, SpellCheckerDictionary-en.jar, SpellCheckerDictionary-fr.jar, SpellCheckerDictionary-ge.jar, SpellCheckerDictionary-it.jar, SpellCheckerDictionary-sp.jar, SpellCheckerDictionary-sw.jar</p> <p>** The ThirdParty.jar is deprecated with BBj 7.0.</p>	

continued...



Q Are there any drawbacks to using jar indexing with Web start?

A Generally, no, however one possible drawback to jar indexing in a Web start environment is the lack of caching of the indexed jars on the client. If caching is needed for certain jar files, use the 'lazy' attribute. When used in conjunction with the **BBjIndex.jar**, the 'lazy' attribute will download the jar on demand, and keep the jar in the client cache. See the sample JNLP file below.

```
<?xml version="1.0" encoding="UTF-8"?>
<jnlp spec="1.0+"
  codebase="http://www.test.com/jnlp"
  href="indexed.jnlp">
  <information>
    <title>Sample Index JNLP</title>
    <vendor>BASIS International Ltd.</vendor>
    <icon href="myicon.gif"/>
    <description>Sample JNLP With Index and Lazy Elements</description>
    <offline-allowed/>
  </information>
  <security>
    <all-permissions/>
  </security>
  <resources>
    <j2se version="1.6+" initial-heap-size="48m" max-heap-size="48m"/>
    <jar href="BBjThinClient.jar"/>
    <jar href="BBjIndex.jar"/>
    <jar href="BBjUtil.jar"/>
    <jar href="Charts.jar" download="lazy"/>
    <jar href="SpellChecker.jar" download="lazy"/>
    <jar href="SpellCheckerDictionary-en.jar" download="lazy"/>
    ...
  </resources>
</jnlp>
```

Q Will the JAR changes also affect applets?

A Yes. In previous revisions of BBj, the **ThirdParty.jar** was a key jar element loaded in the applet HTML file or Java Script. BBj 7.0 deprecated the **ThirdParty.jar**, so we recommend changing the **ThirdParty.jar** element to list **BBjUtil.jar**.

Applications that take advantage of new BBj 7.0 controls will also need to list the associated jar files or list the **BBjIndex.jar** in the HTML or Java script file. Here is a sample Java script with **BBjIndex.jar**:

```
string = '';
string += '<object classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93" codebase="http://
java.sun.com/products/plugin/autodl/jinstall-1_4_0-win.cab#Version=1,4,0,mn" height="400"
width="500">\n';
string += '  \n';
string += '    <param name="type" value="application/x-java-applet;version=1.6">\n';
string += '    <param name="scriptable" value="false">\n';
string += '    <param name="CODE" value="com.basis.bbj.client.bbjapplet.BBjApplet">\n';
string += '    <param name="cache_archive" value="../../lib/BBjThinClient.jar,../../lib/
BBjUtil.jar,../../lib/BBjIndex.jar">\n';
string += '    <param name="cache_option" value="Plugin">\n';
string += '    <param name="cache_version" value="1.0.0.0,1.0.0.0,1.0.0.0">\n';
...

```

Q How will the JAR changes affect installation?

A The BBj 7.0 installation will check for and remove any pre-existing BBj jars in the **bbjhome/lib** directory of the target installation directory. It is important to note that BBj 7.0 *must be uninstalled* prior to installing an older version of BBj to the same directory. Jar and configuration files new to BBj 7.0 will cause unpredictable behavior in older versions of BBj. 