

Custom Apps Install – Easy as 1- 2- 3!

In the last issue of the *BASIS International Advantage*, [Introducing the BASIS Custom Installer](#) explained how BASIS had grown tired of using third party installers that were forever being discontinued or whose functionality and support did not meet BASIS' exacting standards. That article unveiled the exciting new smaller, faster installer, and hinted that it will facilitate the installation of non-BASIS applications in the future. Well, the future has arrived with the release of BBJ® version 11.0!

In these three easy steps, the [BASIS Customer Installer](#) (BCI) lets you install your application software in tandem with the BASIS product suite:

1. Create a .jar file that contains all files necessary to run your software.
2. Create an instruction file.
3. Add the files created in Steps 1 and 2 to a BASIS installable jar.

Let's look more closely as just how easy this is.

Step 1 - Create a Jar

The Jar Utility installs with Java and is much like the familiar .zip or .tar archival tools. In this case it saves multiple files in compressed format into one file with a .jar extension. The jar file must contain all the files needed to make your software run such as BBJ programs and resources, data files, configuration files, images, etc. These files are typically located in one directory so the jar command would be something like this:

```
jar cvf package_demos.jar demos
```

The syntax is **jar**, followed by "options," the name of the destination jar file, and then the source directory in which all the software resources exists. The **cvf** options in this example are "c"reate, "v"erbose, and "f"ile.

Step 2 - Create an Instruction File

The instruction file named **custominstall.xml** is simply an XML file that can be easily created, viewed, and modified using the BASIS IDE. It provides the installer with the detailed instructions for installing the software such as the directories in which to install specified files, on which ports (operating systems) to install particular files, and under what conditions to proceed with the installation. Other options include using installation variables like the **\$InstallDir** variable that contains the installation directory selected by the user. See the example in [Figure 1](#).

For a full-featured custom install, make use of some of the BCI's advanced features

- Create shortcuts on any OS to run your software
- Update files using Search/Keep and Replace nodes in the instruction file
- Create, modify, and remove registry entries on Windows
- Specify process execution, killing of processes, and invoking of custom Java code located in the instruction file to invoke at pre-install, post-install, pre-uninstall, or post-uninstall

Step 3 - Add Files to the Installable Jar

Select the appropriate BBJ installation bundle that fits your needs and simply add the jar that contains your software created in Step 1 and then add the instruction file that was created in Step 2. Once again, use the Jar Utility to accomplish this step as shown here:

```
jar uvf BBJBaristaIDE1110_08-02-2011_1413.jar package_demos.jar custominstall.xml
```

>>



By Brian Hipple
Quality Assurance
Supervisor

```

<?xml version="1.0" encoding="UTF-8"?>
<Suites>
  <Suite name="AdminAPI Demo Suite">
    <!-- AdminAPI Feature -->
    <Feature name="AdminAPI">
      <Jar name="package_adminapi.jar">
        <!-- readme -->
        <File name="AdminAPI/documentation/readme.txt" installdir="demos/AdminAPI"/>
        <!-- images -->
        <File name="AdminAPI/images/unix/AdminAPI.xpm" installdir="demos/images" portid="2145,6146,2169,2172,6173,2179,2184"/>
        <File name="AdminAPI/images/mac/AdminAPI.icns" installdir="demos/images" portid="2120"/>
        <File name="AdminAPI/images/win/AdminAPI.ico" installdir="demos/images" portid="2166"/>

        <!-- Unix script -->
        <File name="AdminAPI/script/adminapi" installdir="bin" portid="2120,2145,6146,2169,2172,6173,2179,2184" permission="rwx+ ">
          <Keep variable="$CustomDefs" begin="$BeginCustomDefs" end="$EndCustomDefs"/>
          <Replace search="$customdefs$" replace="$CustomDefs"/>
        </File>

        <!-- Mac Shortcut -->
        <Directory name="AdminAPI/mac/AdminAPI.app" installdir="/Applications/BASIS/AdminAPI.app" portid="2120"/>
        <File name="AdminAPI/mac/AdminAPI.app/Contents/Info.plist" installdir="/Applications/BASIS/AdminAPI.app/Contents" portid="2120">
          <Replace search="$version$" replace="$Version"/>
        </File>
        <File name="AdminAPI/mac/AdminAPI.app/Contents/Resources/English.lproj/InfoPlist.strings"
          installdir="/Applications/BASIS/AdminAPI.app/Contents/Resources/English.lproj" portid="2120">
          <Replace search="$version$" replace="1.0"/>
        </File>
        <File name="AdminAPI/mac/AdminAPI.app/Contents/Resources/script"
          installdir="/Applications/BASIS/AdminAPI.app/Contents/Resources" portid="2120" permission="rwx+ ">
          <Replace search="$installDir$" replace="$InstallDir"/>
        </File>
        <File name="AdminAPI/mac/AdminAPI.app/Contents/MacOS/AdminAPI"
          installdir="/Applications/BASIS/AdminAPI.app/Contents/MacOS" portid="2120" permission="rwx+ ">
        <!-- rest of files in the main directory -->
        <Directory name="AdminAPI" installdir="demos/AdminAPI"/>
      </Jar>

      <!-- Shortcuts -->
      <Shortcut name="BASIS/BBj/Demos/AdminAPI" target="bin/bbj.exe" parameters="-tT0 -g -WDSInstallDir/demos/AdminAPI AdminAPI"
        workingdir="demos/AdminAPI" image="demos/images/AdmnAPI.ico" description="Runs AdminAPI Demo" portid="2166"/>
      <Shortcut name="AdminAPI" target="bin/adminapi" image="demos/images/AdminAPI.xpm" description="Runs AdminAPI demo"
        portid="2145,2169,2172,6146,6173"/>
      <Shortcut name="BASIS/BBj/Demos/AdminAPIReadMe" target="demos/AdminAPI/readme.htm" parameters=""
        workingdir="demos/AdminAPI" image="images/Documentation.ico" description="AdminAPI Demo ReadMe" portid="2166"/>
    </Feature>
    <!-- End of AdminAPI Feature -->
  </Suite>
</Suites>
23:19 INS

```

Figure 1. Sample instruction file

The syntax is similar to creating the jar; start with `jar`, followed by “options,” the name of the installation bundle jar file, and then the jar and instruction file. The `uvf` options in this example are “u”pdate, “v”erbose, and “f”ile. The installation is now ready to run. Selecting a custom installation type shows your newly added suite as **Figure 2** illustrates.

Summary

Instead of your end users installing the BASIS products and then installing your software, wouldn't it be better to combine it all in one installation process? Using the powerful new BCI, bundling your software along with the BASIS products installation is as easy as 1-2-3! Remember too that you can create a response file and install silently, only prompting the user for user-specific information. Now you can benefit from BASIS' engineering efforts and reap the rewards both monetarily and functionally by discarding your third party installer and using the BCI. This is just another example of BASIS' building block strategy at work, saving you money and making your business more efficient. ■

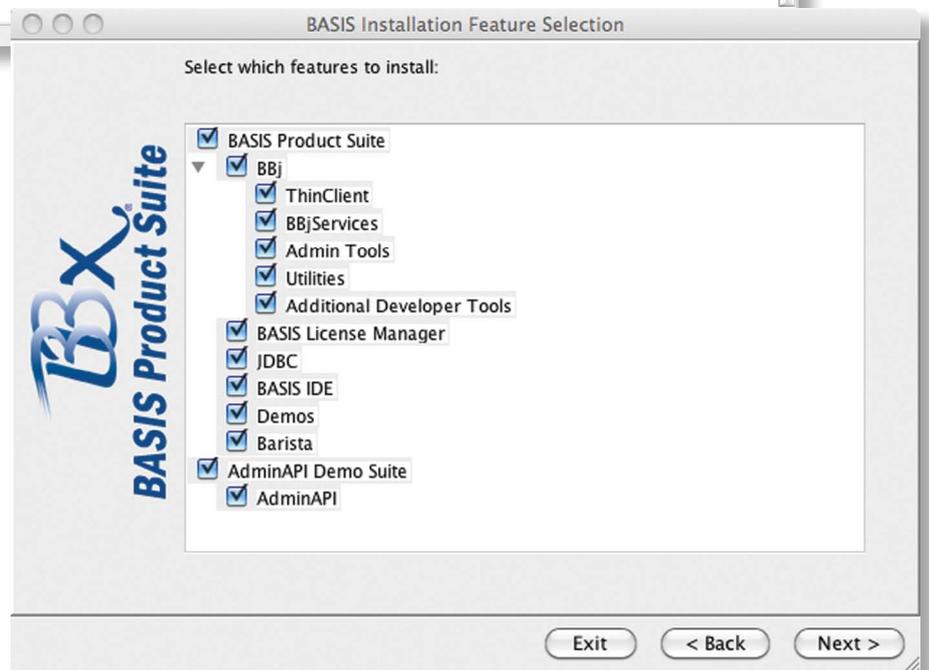


Figure 2. BASIS Installation Feature Selection



Review the full description of a BCI custom app installation and all that is available in the instruction file in the online help topic [BCI Overview](#)