

Using the Barista Installation Manager to Install and Maintain Multiple Copies of Barista

Introduction

The Barista® Installation Manager (BIM) acts as a gatekeeper to provide developers a mechanism for creating and maintaining multiple, distinct copies of Barista on a single machine. A developer or reseller may elect to run Barista in this fashion for several reasons:

- to keep demo copies of one or more Barista applications on the same machine as a development environment
- to facilitate multiple developers working on different Barista projects on the same server
- to allow Barista applications for multiple clients to reside on a single machine, while keeping them separate/isolated from one another

This document illustrates how to use the BIM to create multiple copies of Barista and synchronize them with your applications, using AddonSoftware™ from the BASIS svn repository and a hypothetical vertical called "myapp" as examples.

What You'll Need

Latest software

Before beginning, be sure to install the latest recommended versions of BBj® and Barista. BASIS recommends uninstalling any previous versions of BBj/Barista prior to a new install. If using the BASIS IDE to interface with SVN, be sure to select/install the IDE as well. After the install there will be a "clean" copy of Barista in the <basis install> directory, i.e., it contains no application-specific data or pathname references. When you create a new instance of Barista using the BIM, this unmodified version of Barista is copied to the location where your application files reside. As such, you will typically never run this originally installed Barista, but use it as the source for creating copies of Barista elsewhere.

SVN

In order to check out a copy of AddonSoftware from the BASIS svn repository, you must have svn client software and a BASIS-issued private key. This document assumes a functioning svn environment. Contact addon-developer@basis.com or BASIS if you need assistance with SVN.


Understanding the File Structures

Overview

It is important to understand the various configuration files in play when running multiple copies of Barista. Fortunately, the BIM does most of the heavy lifting for you.

Each copy of Barista uses its own barista.cfg file, located in <app_install>/barista/sys/config/<language>. The barista.cfg file contains numerous settings used by Barista, as well as stbl and pathname settings for any application(s) synchronized into Barista.

Barista makes use of the BASIS SQL engine when performing inquiries, so each copy of Barista contains an <app_install>/barista/bbdict folder as well, containing the BASIS dd files. Because Barista hands over control to the BASIS SQL engine for inquiries, the BASIS config.ini file must contain information about each database being queried.

 **Important:** Each copy of Barista must refer to a unique database name in the BASIS config.ini file. When you create a new copy of Barista, the BIM will copy the default "[Barista]" section of config.ini and paste it back into the config.ini with your new database name.

Last but not least, each application that you'll be synchronizing into a given copy of Barista needs a synchronization (.syn) file containing application-specific settings and/or pathnames. The information in your application's app_name.syn file, located in <app_install>/apps/<app_name>/config, is used to update the barista.cfg file and some Barista administrative tables, as well as the BASIS config.ini file.

Required File Structure

Prior to using the BIM to create a new copy of Barista for your application, you need to have your application stored in the format depicted below. Directory/file names in <angle brackets> are those you supply. The apps/ directory is required by the BIM and must be defined inside your <app_install> folder. Other directories are required and/or defined by Barista.

```
<app_install>/
  apps/
    <app_name>/
      callpoints/
      config/
      <app_name.syn>
      data/
      ...
```

Once this structure is in place, the BIM will copy Barista into <app_install>/barista/, and make the necessary configuration file additions/modifications. Below is file structure diagram showing how the BIM sets up two copies of Barista: one running a copy of AddonSoftware checked out from the svn repository, and another running an application called "myapp."

After the BIM makes a copy of the "clean" Barista in the target location, it updates the barista.cfg file with the new database name and barista directory pathnames, and also creates a section in the BASIS config.ini for the new database.

When you actually launch the application for the first time, the Auto-Synchronization process further updates `barista.cfg` and `config.ini` based on the entries in the application's `.syn` file ("`app_name.syn`" in the diagram above).

DRAFT

```

<BASIS_home>
  barista/ (contains "clean" Barista)
  bin/ (contains bbj.exe, IDE)
  cfg/
    config.ini:
      [Barista]
        DATABASE=Barista
        DICTIONARY=<BASIS_home>/barista/bbdict/
      [ADDON]
        +DATA=<addon_svn_sandbox>/apps/aon/data/
        DATABASE=ADDON
        DICTIONARY=<addon_svn_sandbox>/barista/bbdict/
      [MYAPP]
        +DATA=<myapp_install>/apps/myapp/data/
        DATABASE=MYAPP
        DICTIONARY=<myapp_install>/barista/bbdict/

```

```

<addon_svn_sandbox>/
  apps/
    aon/
      config/
        addon_BIM.syn:
          DIR=<addon_svn_sandbox>/apps/aon
          STBL=SET +DATA=<addon_svn_sandbox>/apps/aon/data/
          CFG=+DATA=<addon_svn_sandbox>/apps/aon/data/
      data/
        bar/
        cdf/
        data files...
    barista/
      sys/
        config/
          enu/
            barista.cfg:
              [BaristaSystemSettings ]
              SET +DBNAME=ADDON
              [UserDataDirectories]
              SET +DATA=<addon svn sandbox>/apps/aon/data/

```

```

<myapp_install>/
  apps/
    myapp/
      config/
        myapp.syn:
          DIR=<myapp_install>/apps/myapp
          STBL=SET +DATA=<myapp_install>/apps/myapp/data/
          CFG=+DATA=<myapp_install>/apps/myapp/data/
      data/
        bar/
        cdf/
        data files...
    barista/
      sys/
        config/
          enu/
            barista.cfg:
              [BaristaSystemSettings ]
              SET +DBNAME=MYAPP
              [UserDataDirectories]
              SET +DATA=<myapp_install>/apps/myapp/data/

```

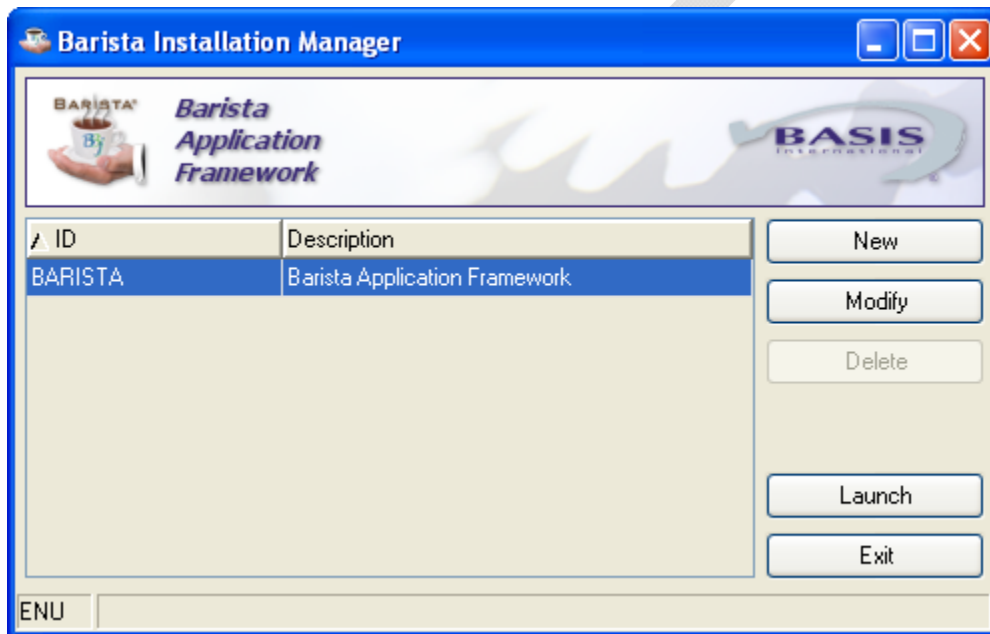
Launching and Running the BIM

Create a Modified Shortcut

The BIM is built into Barista, so you don't need to install any additional components in order to use it. Barista starts up with or without the BIM depending on how your launch command is configured. On a windows machine, create a copy of the Barista shortcut and modify the target line by appending " - -s" to the end (that's a space, dash, space, dash, "s").

For a Linux machine, open `/usr/local/basis/bin/barista` in an editor and add " - -s" to the end of the exec line.

Once you've altered the Barista shortcut, you can launch Barista. After logging in, you'll see the Barista Installation Manager form:



The BARISTA line above refers to the unused copy of Barista in the `<BASIS install>/` directory. From here, you can create a new copy of Barista, modify settings for an existing copy, delete a copy, or simply launch the desired copy.

You may, if desired, create a separate shortcut pointing directly to an installed copy of Barista if you want to bypass the BIM form once the copy is made.

Create a New Barista

The following shows what happens when you select the "New" button. The Create/Modify Barista Installation window appears, and you're prompted to fill in an Installation ID, Description and Location. The Installation ID is the name that Barista will use to set up the `+DBNAME stbl` and create the database section in the BASIS

config.ini file. Fill out (or browse to) the Location field using the name of the directory/folder in which your application is stored.

Create/Modify Barista Installation

Installation ID:

Description:

Location:

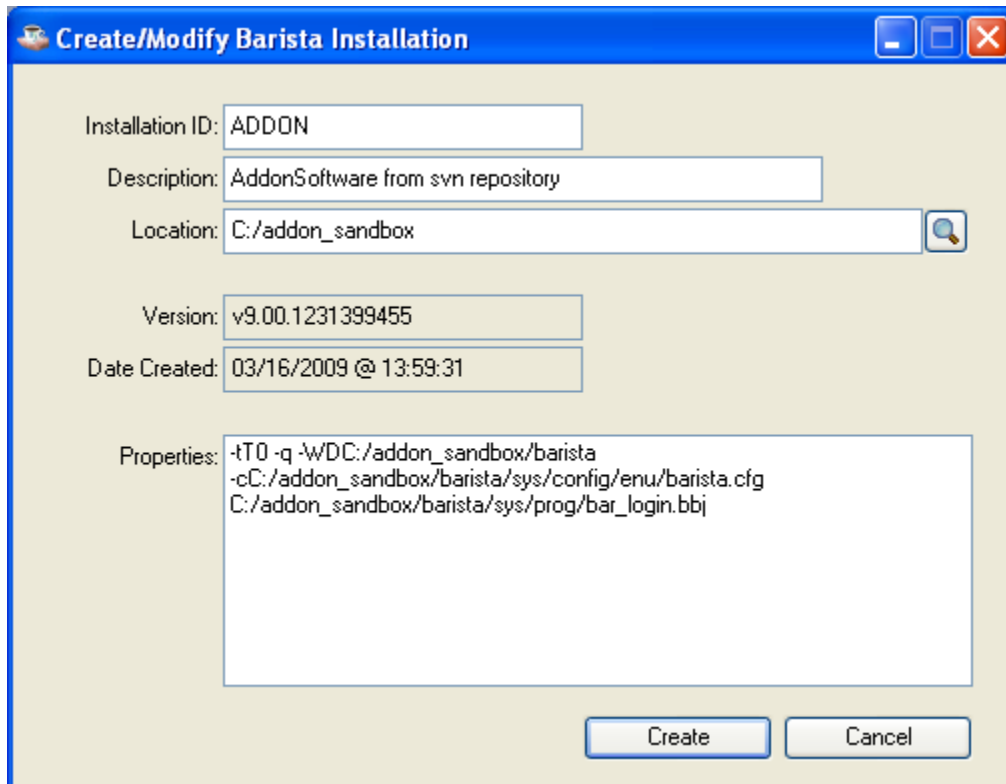
Version:

Date Created:

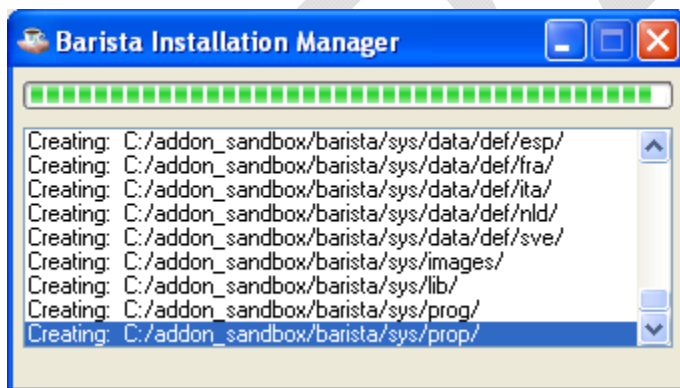
Properties:

```
tT0 -q -wD<<DIR>>/barista -c<<DIR>>/barista/sys/config/enu/barista.cfg  
<<DIR>>/barista/sys/prog/bar_login.bbj
```

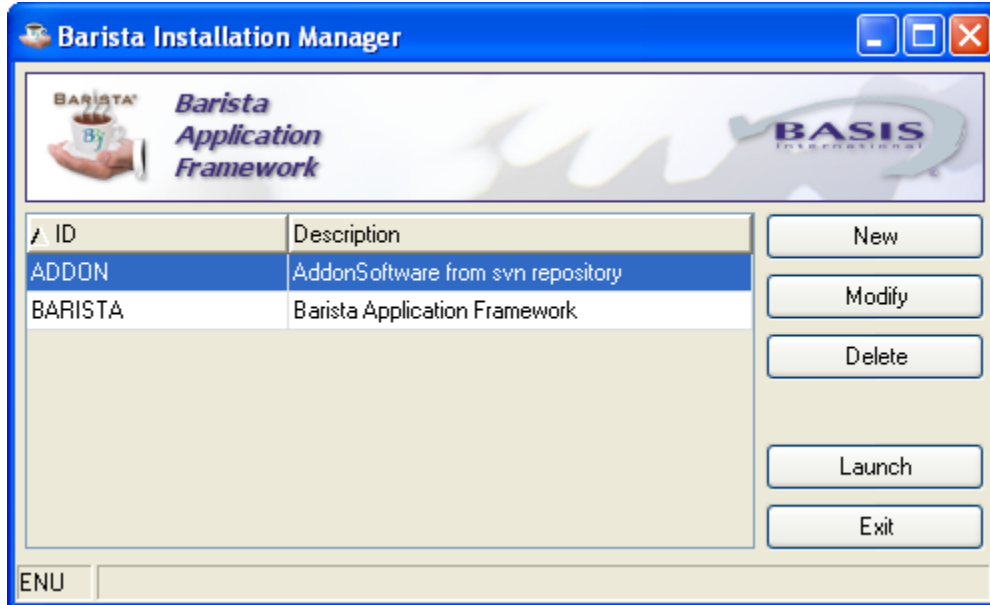
Once you tab out of the Location field, Barista fills in the Properties box using the new information. Below is another picture of the form, filled out using our AddonSoftware "<addon_svn_sandbox>" example:



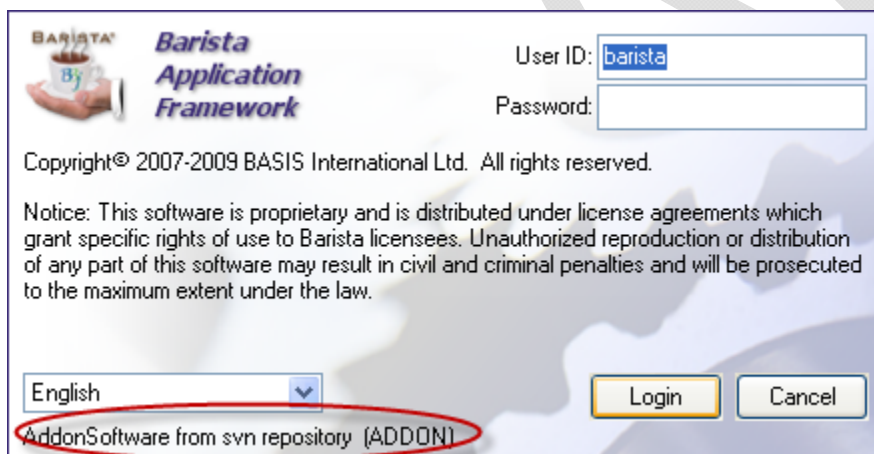
When you press the Create button, Barista installs a copy of itself in the location specified and makes the necessary configuration file entries.



You now have a fully functional copy of Barista ready to launch:



When launching from the BIM, you'll be presented with an additional login screen, showing you which application/copy of Barista you're launching:



Auto-Synchronize Your Application

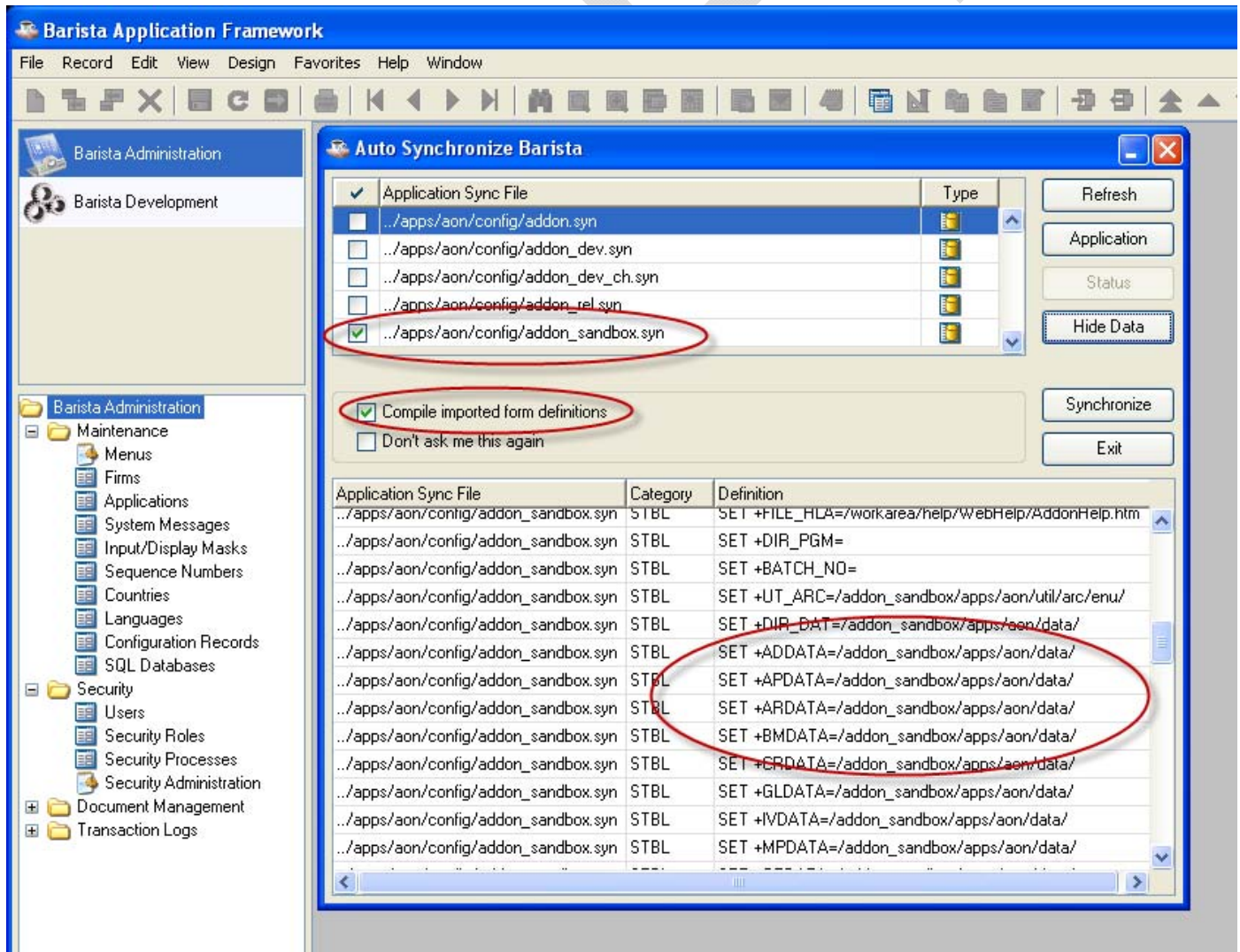
When you launch an application from the BIM the first time, the Auto-Synchronize process builds the Barista and BASIS dd files and compiles callpoint code for the application. The Auto-Sync looks in the directory structure you've established and lists any .syn file as candidates for synchronization. If you have multiple applications under the <app_install>/apps/ directory, such as AddonSoftware and your own vertical, you may want to synchronize them both into your new copy of Barista.

In the case of AddonSoftware as checked out from the svn repository, you'll see more than one .syn file listed. These .syn files differ internally in how the pathnames are defined. The version of Addon that is pre-built and available for download from the

website uses `addon.syn`, which contains pathnames of `"../apps/aon/data/"`. The `addon_dev.syn` file is used by Addon VAR's who checkout from the repository, and are running a single copy of Barista. Pathnames in this `.syn` file are the default `"/apps/aon/data."` You should not modify either of these files, but rather save a copy under a different name, and then replace the pathnames to reflect where you've saved your copy of Addon.

Important: The Auto-Synchronize process will look in the directory structure for your application for any `.syn` files. Make sure you have created a customized `.syn` file that contains stbl's, pathnames, etc. corresponding to where you've stored your application!

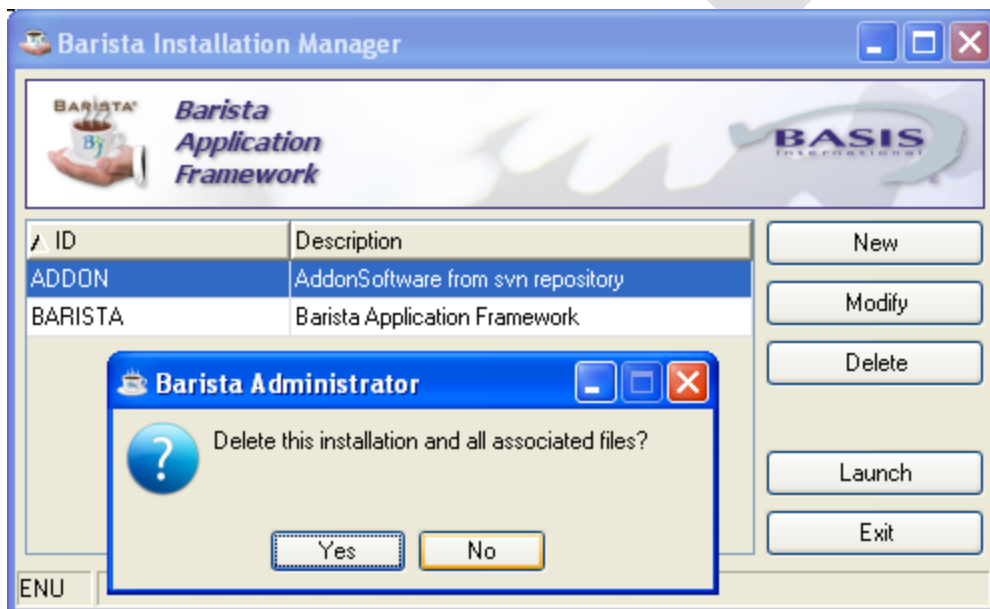
Using our previous diagram, if you've created `<addon_svn_sandbox>` as `C:/addon_sandbox` on your system, you will set pathnames inside your copy of `addon_<my_copy>.syn` to say `"/addon_sandbox/apps/aon/data/,"` as shown in the screenshot below.



Since Barista builds the arc and def files needed to run forms, the arcs and defs themselves are not stored in the repository. You have a few options for building forms. First, you can check the "Compile imported form definitions" box in the Auto-Sync, as pictured above. Alternatively, you can leave this box unchecked, and then rebuild all forms from the Form Manager after the Auto-Sync has completed and you launch Addon (or any other application). Both of these options will build all forms, which is nice, but will add to the time needed to get the application up and running. If you elect not to build all forms, Barista will build them the first time a given form is launched. This option makes for faster startup, but will impose a bit of overhead as you run any form that hasn't been built before.

Deleting a Copy of Barista

The BIM also lets you remove a copy of Barista that has been synchronized with one or more of your applications. The deletion removes the <app_install>/barista/ folder, and removes the corresponding database section from the BASIS config.ini file. The delete process does NOT delete any of your application files.



After deletion, your file structure will once again look like what we started with – the Barista folder has been removed:

```
<app_install>/
  apps/
    <app_name>/
      callpoints/
      config/
      <app_name.syn>
      data/
      ...
```

When you want to upgrade the copy of Barista for one of your applications, install a new BBJ/Barista, then use the Delete process to remove the old Barista, and the Create process to once again copy the new Barista into your <app_install> directory.

DRAFT