

Create Vertical and Customize Applications with Barista

Part 1: Create and Synchronize Barista Applications

As a developer, you may be using Barista® Application Framework to create your own vertical applications, modify existing applications, or both. This tutorial takes you through the process of creating an application/project file structure to store your customizations or your new application, and also demonstrates installing the app/project into Barista at a user's site.

This three part document assumes a working knowledge of Barista development. Visit <http://www.basis.com/products/devtools/barista/documentation/> for additional Barista documentation.

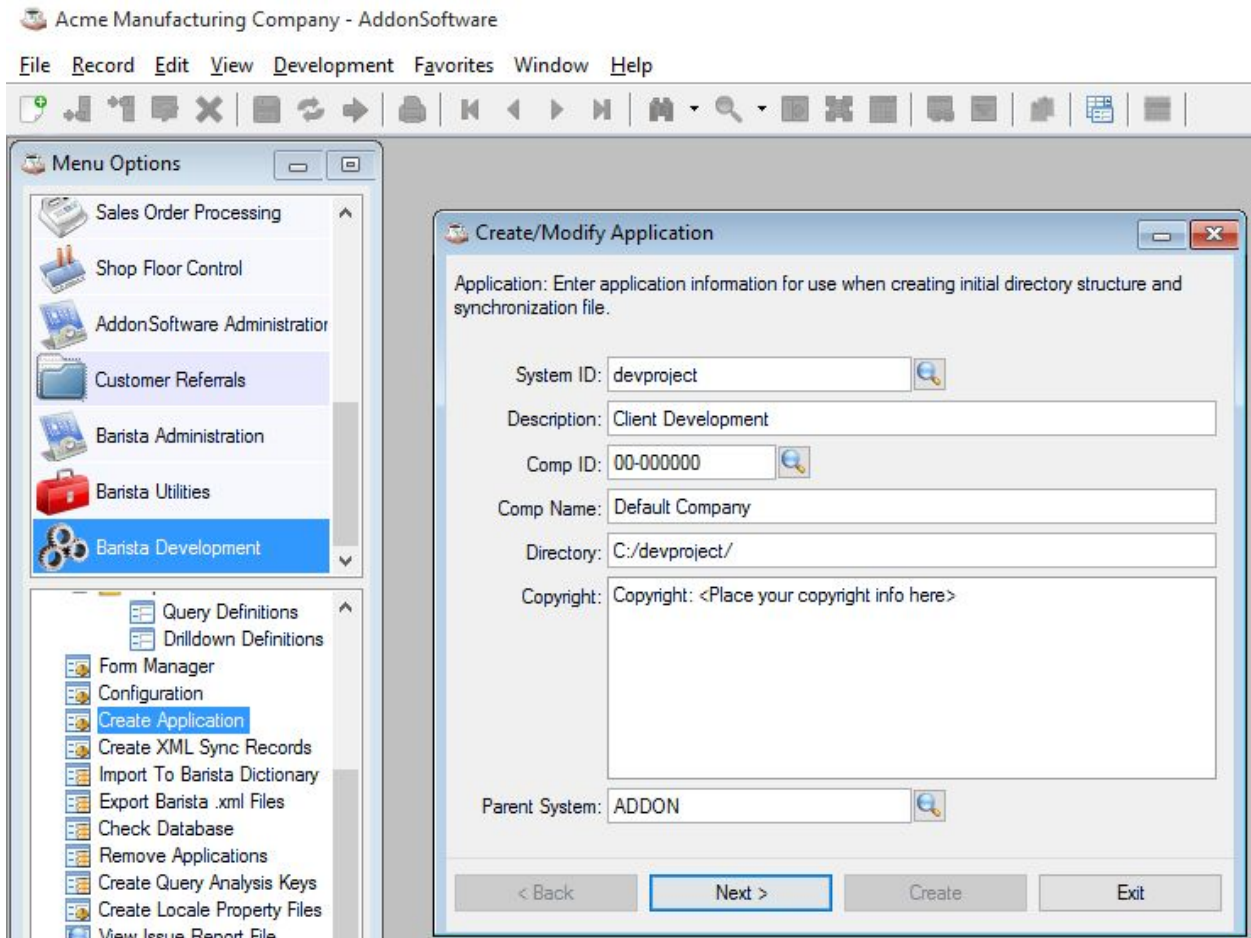
Barista's 'Create Application Wizard'

The Create Application Wizard is an important tool for any developer who wants to either customize existing Barista applications, or create their own verticals. The wizard leads you through several forms in which you specify the name of the project/application, the pathname Barista should use in creating the project directory structure, values for several STBL's, etc. Much of the information in the wizard is defaulted based on the initial project and pathnames supplied, and in most instances, these defaults can be accepted as is. The wizard creates the necessary directory structure and synchronization (.syn) file for your project.

If you are relatively new to Barista development, it may not be obvious why you need create an application/project for customizations to an existing application. After all, that application already resides within its own directory structure. However, if the application is one that you will want to re-install or update periodically in order to obtain bug fixes and/or enhancements, you do not want to save your customizations into that directory structure, or you risk losing them after an update.

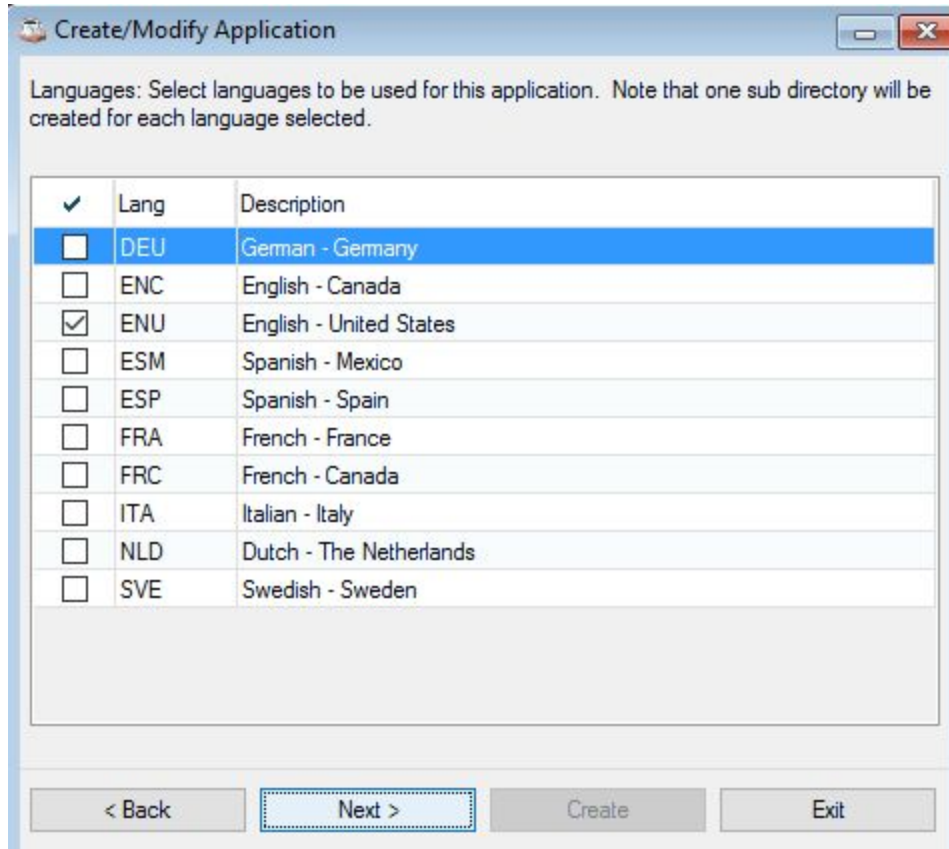
In Part 1, we will create a new project called "devproject." In Part 2, we'll create a new Customer Referrals form and report that interfaces with AddonSoftware. In Part 3, we'll use Application Replication to make a modification to Addon's Customer Master form, tying it to our Customer Referrals table.

To begin, select Barista Development in the top menu panel, and then click in the lower section of the menu to launch the Create Application Wizard:

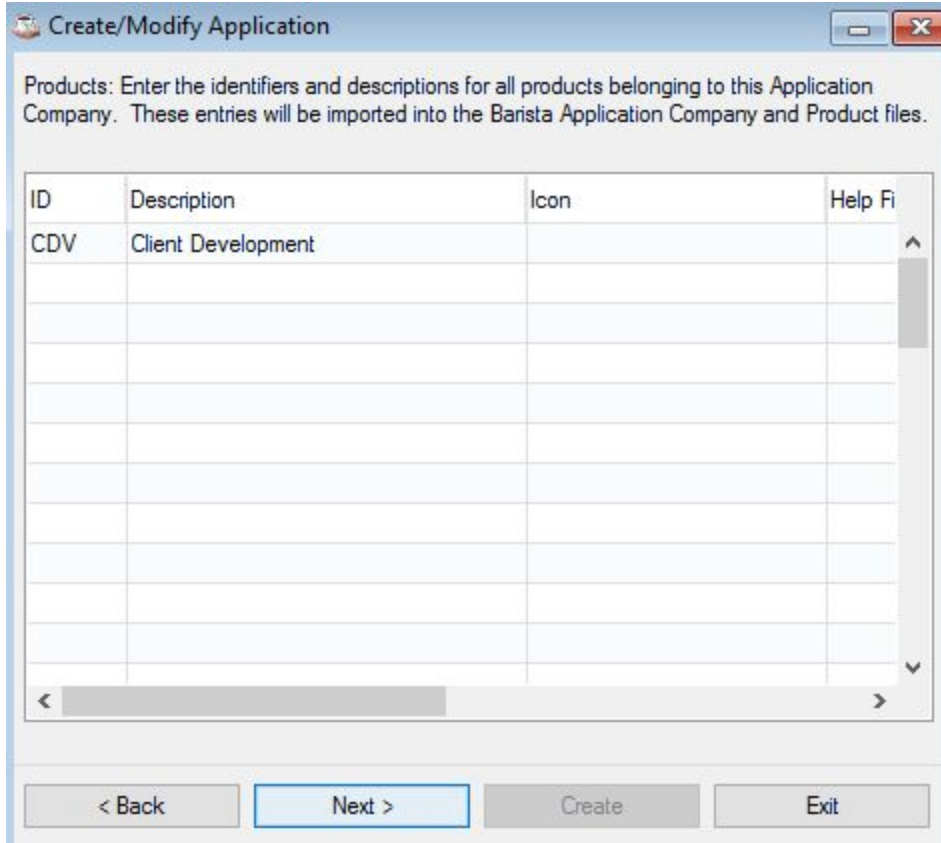


Enter an ID for the project name in the System ID field, and provide a Description. Next, provide a valid BASIS Company ID. The 00-000000 Comp ID is for development purposes only and requires a developer license. We will use 00-000000 for this tutorial but you should use your BASIS Comp ID if you will license and install this project elsewhere. If you supply your own Company ID, then also enter a descriptive Comp Name for the project. Barista defaults the directory location based on the System ID. You can change the path, but the final/root directory name should match the System ID. Since we will be customizing AddonSoftware, we will want to associate our new project with Addon by selecting ADDON for the Parent System (if you are creating a standalone project, leave the Parent System blank).

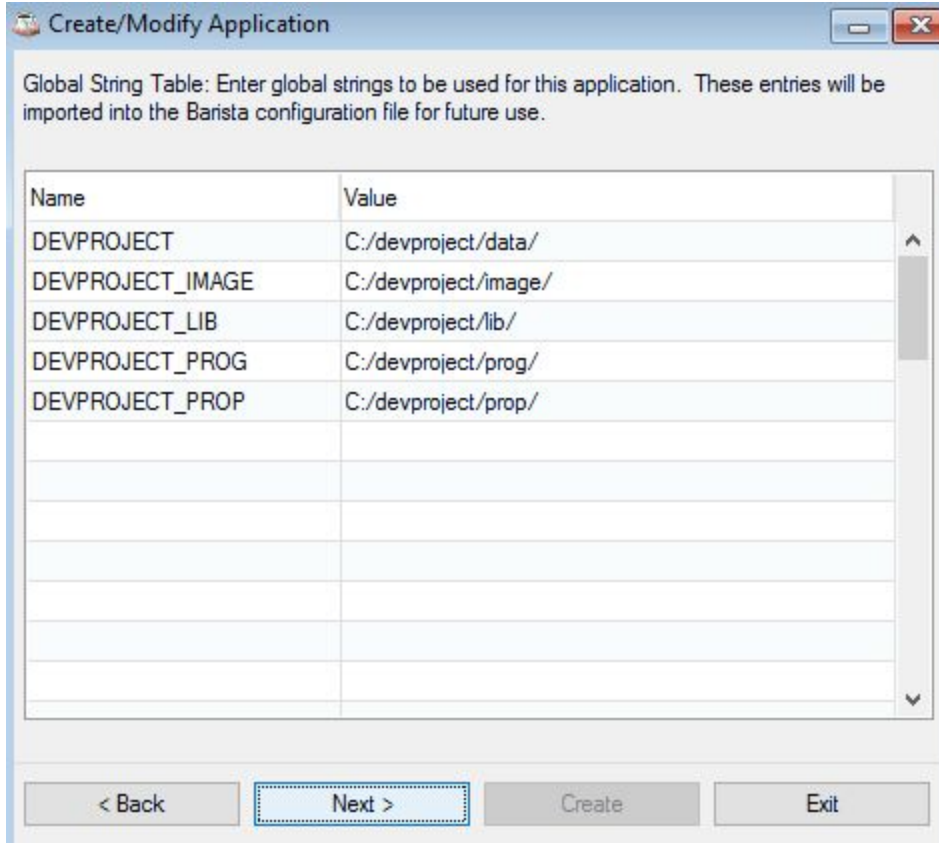
Click **Next** to move to the next panel and select the desired languages:



The next panel contains Product ID's and Descriptions. As a developer, you will register your product ID's with BASIS for licensing purposes. We will create one product as shown below (*Note: The "DEV" Product ID that ships with Barista is intended for training or demonstration purposes only, and should not be used when creating a new project.*)

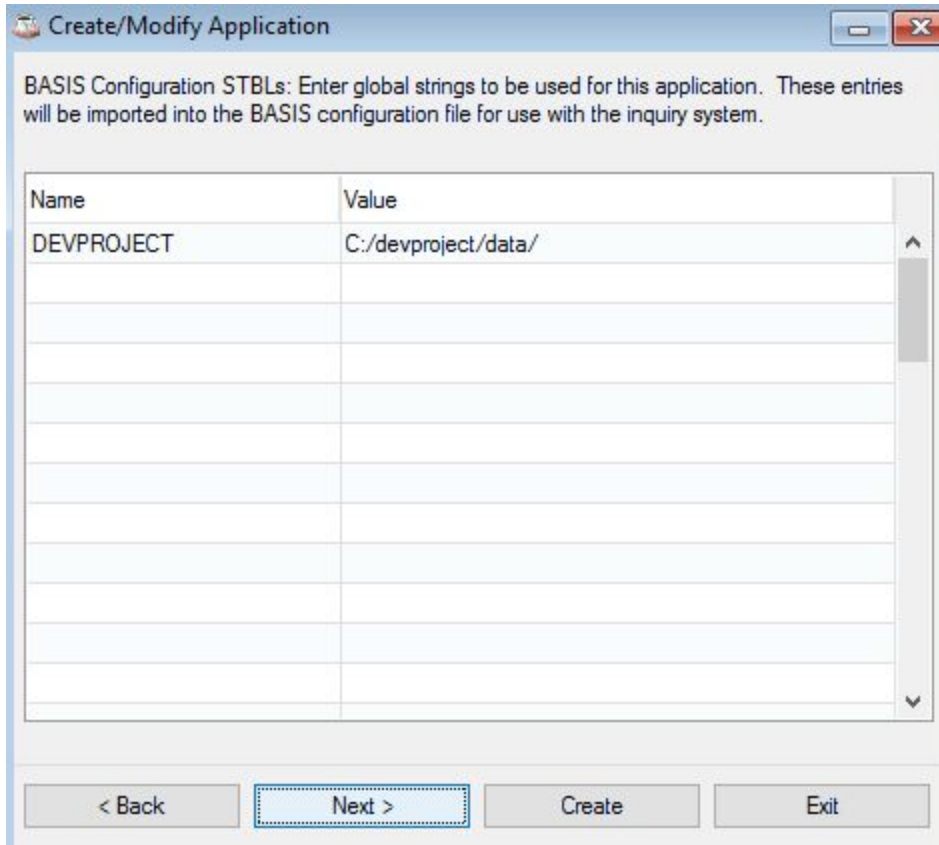


Click **Next** to move to the Global String Table panel. On this panel you'll see several default entries that Barista creates and will incorporate into the barista.cfg file.

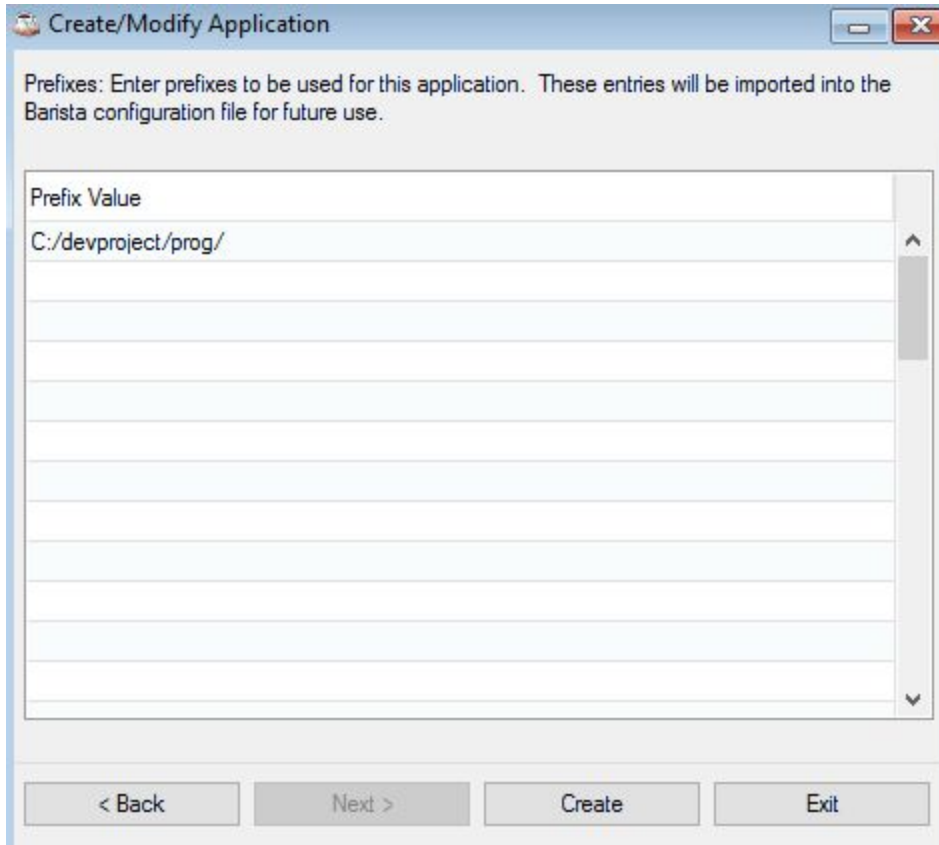


- DEVPROJECT references the project's data directory. This STBL is typically used whenever creating a data file rather than supplying a hard-coded pathname. In addition, Barista's alias and callpoint definition files are stored in subdirectories here.
- DEVPROJECT_IMAGE is the location of the directory you can use to store images, logos, etc.
- DEVPROJECT_LIB provides a reference to a code library directory, if desired.
- DEVPROJECT_PROG refers to the directory containing "back-end" programs such as reports, publics, and updates.
- DEVPROJECT_PROP tells Barista to store Java resource bundle files for language translation.

Clicking **Next** takes you to the BASIS Configuration STBLs panel, which again defaults the STBL for the data directory. This value integrates into the BASIS config.ini file for use by the SQL engine during Barista table inquiries.

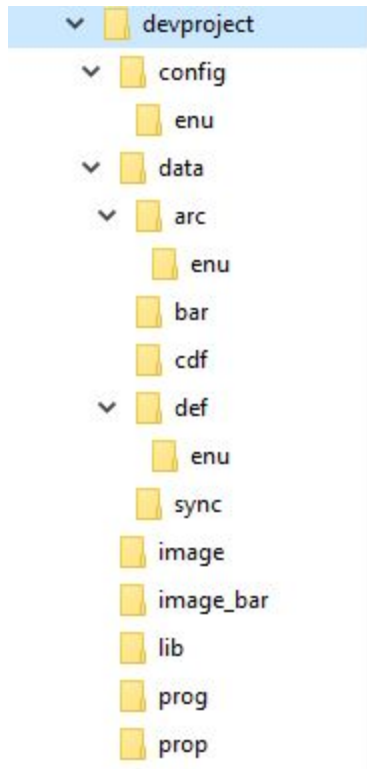


Click **Next** to move to the last panel, where a prefix entry has been created that references the new application's "prog" folder.



The **Create** button presents a confirmation dialog; click **No** to exit the utility without creating the new application/project or **Yes** to create the directory structure and .syn file. Barista also adds a new entry to the barista.syn file, or creates it, if necessary.

Based on our devproject example, Barista creates the following directory structure:



Directory

config/

config/enu/

data/

data/arc/enu/

data/bar/

data/cdf

data/def/enu/

data/sync/

image

image_bar

lib/

prog/

prop/

Use

devproject.syn file

application menu file (e.g., devproject.men)

data tables

arc files used to run forms

xml files in which Barista stores alias definitions

callpoint definition files containing code entered via the Barista callpoint editor

definition files used to run forms

xml representations of application-specific data stored in Barista tables, such as system masks and messages

images for forms, documents, etc.

icons to use for customizing the application menu

optional directory for shared code

"back-end" code such as reports, updates, and publics

Java resource bundle files, used in language translation

The contents of the devproject.syn file are shown below.

```
// Application Synchronization File
// Barista Application Framework. Copyright BASIS International Ltd.
SYSID=DEVPROJECT
SYDESC=Client Development
SYSCOMP=00000000;Default Company
```



```
SYSCOPY=Copyright: <Place your copyright info here>^
SYSPAR=ADDON
SYSDIR=C:/devproject/
SYSAPP=CDV;Client Development;;;devproject;C:/devproject/prop/;ENU
SYSSTBL=SET DEVPROJECT=C:/devproject/data/
SYSSTBL=SET DEVPROJECT_IMAGE=C:/devproject/image/
SYSSTBL=SET DEVPROJECT_LIB=C:/devproject/lib/
SYSSTBL=SET DEVPROJECT_PROG=C:/devproject/prog/
SYSSTBL=SET DEVPROJECT_PROP=C:/devproject/prop/
SYSPFX=C:/devproject/prog/
SYSCFG=DEVPROJECT=C:/devproject/data/
SYSLANG=ENU
```

If there is not already a `barista.syn` file in the `<barista install>/sys/config/` directory, the wizard creates one. Otherwise, it adds a new line to `barista.syn`. These entries "point" Barista to the locations of other projects or applications that should be included in the synchronization process. For example, after creating the `devproject` application, `barista.syn` includes this line:

```
SYN=C:/devproject/config/devproject.syn
```

If your application is in the same parent directory as Barista itself, then the entry in the `barista.syn` file is redundant, and can be removed (or the file deleted if your application is the *only* entry). Otherwise, the `barista.syn` file is unique to your environment, so make a backup copy of it. If you install a new Barista, you can paste the backup copy into `<barista install>/sys/config/` so Barista can once again incorporate external applications during the synchronization process.

After creating the application file structure, click **Yes** to launch Barista's Auto-Synchronize utility. Click the **Synchronize** button, then click **Yes** to confirm and start the process.

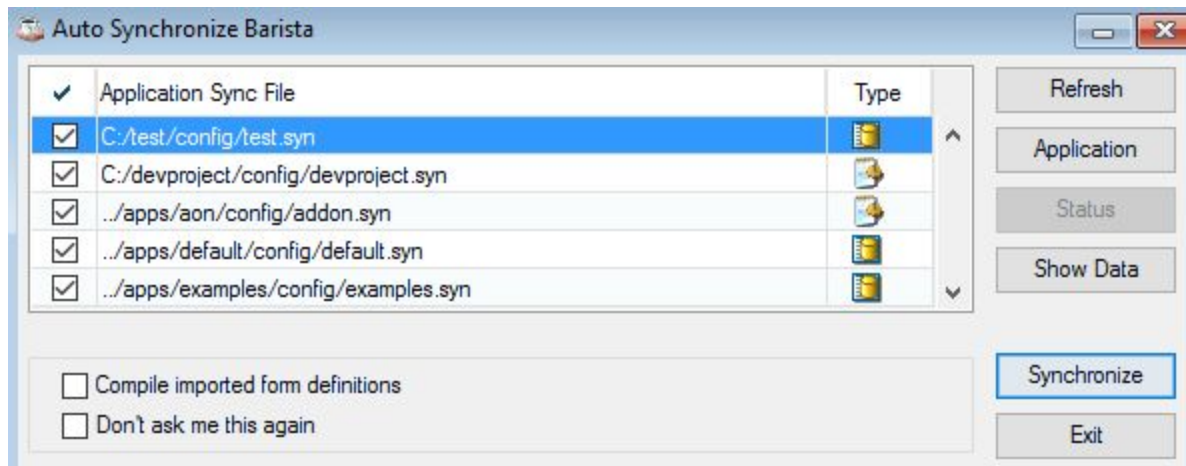
Finally, Barista asks if you want to auto-synchronize your new application. Click **Yes** to launch the Auto-Synchronize Barista utility, otherwise click **No** to exit the Create Application wizard.

Auto-Synchronize Barista

The Auto-Synchronize Barista utility (`auto-sync`) launches automatically after a fresh Barista install. In addition, it can be run immediately after the Create Application wizard or at any other time from the Barista Utilities menu.

The `auto-sync` builds and updates Barista dictionary files, compiles callpoint code, incorporates data from the `<application>/sync/` directory into Barista Administration tables, and updates `barista.cfg` and the `BASIS config.ini` files with settings defined in the `<application>.syn` file. The `auto-sync` can optionally build the form definition files (`<application>/data/arc/<enu>` and `<application>/data/def/<enu>`).

Prior to launching Barista for the first time after install, copy/paste your barista.syn file into the <barista install>/sys/config/ directory. The auto-sync process will look both in the barista.syn file as well as in the apps/ directory hierarchy installed with Barista for <application>.syn files to process during the sync. All .syn files discovered are displayed in the grid at the top of the Auto-Synchronize Barista form:



Applications not yet synchronized into the Barista data dictionary appear with a disk icon, while applications previously synchronized (if this is not a new install) have a gear or "config only" icon. In the latter case, Barista will not process all of the <application>/data/bar/*.xml and <application>/data/cdf/*.cdf files, but only forward data from the <application>.syn file to barista.cfg and the BASIS config.ini files.

Note that, even though marked for a "full" synchronization, the auto-sync for the devproject application will behave more like a "config only" process and take very little time because it is a brand new application. Synchronizing this application will update the barista.cfg and BASIS config.ini files, but no other data has been created in the application, i.e., there is no data in the /bar, /cdf or /sync directories to process.

Once the Auto-Synchronization process is complete, you can begin or resume development for the application within Barista.

See also:

[Part 2: Creating Vertical Applications](#)

[Part 3: Customizing and Installing Applications](#)