

AddonSoftware Installation and Upgrade Processes

Table of Contents

[Introduction](#)

[Installation/Upgrade Process Flow](#)

[One: Preparation](#)

[Two: Installation and Upgrade](#)

[Three: Testing](#)

[Four: Go Live - Refresh Data](#)

[Upgrading from previous version of Barista Addon](#)

[Upgrading from legacy Addon or non-Addon application](#)

[New Installation](#)

[APPENDIX A - Understanding the Java Classpath](#)

[Specify a session-specific classpath in Enterprise Manager.](#)

[Add the session-specific classpath to Barista Installation Manager \(BIM\).](#)

[APPENDIX B - Understanding How Upgrades Handle Menus](#)

[APPENDIX C - Installation and Upgrade \(Manual Method\)](#)

[APPENDIX D - Understanding the Utilities](#)

[Barista Utilities](#)

[Create Sync File Backup](#)

[Auto-Synchronize Barista](#)

[Install Application Wizard](#)

[Create Application Wizard](#)

[Addon Utilities](#)

[Install Wizard](#)

[Upgrade Wizard](#)

[Copy aon Directory](#)

[Update addon.syn File](#)

[Update sync XML Files](#)

[Set Up New Firm](#)

[Copy Application](#)

[Distribute EM Update Wizard Files](#)

[File Utility](#)

[Copy Directory Recursively](#)

[DataPort](#)

[APPENDIX E - Using Git in the Upgrade Process](#)

[\(back to contents\)](#)

Installing and Upgrading Barista Addon Using the AddonSoftware Install/Upgrade Utilities and Wizards

Introduction

AddonSoftware® by Barista® was developed and runs within the Barista Application Framework. One installation of BBj can support many different copies or instances of Barista, each with its own BBj® database and Barista dictionary. A given copy of Barista can, in turn, be synchronized with any number of applications. In terms of AddonSoftware (Addon), this architecture makes it possible to install and run several versions of Barista Addon on a single machine (each with its own instance of Barista), and also allows a single instance of Barista to support Barista Addon along with any number of other applications containing Addon customizations and/or verticals.

A complete demo/evaluation copy of Barista with Addon is available from the BASIS download page (www.basis.com/products/bbj/download.html). This version of Barista Addon is “pre-built,” that is, the forms and callpoints have all been compiled, the Barista dictionary tables are populated, and demo data is included. As such, you can download and install it very quickly. By default, the pre-built version is installed in the same directory as BBj.

WARNING

If you want to run Barista Addon in a production environment, you must move Addon to a different location and synchronize it with its own copy of Barista,

This not only ensures that you won't overlay production files should you decide to download/install another demo copy, but also facilitates running more than one copy of Addon, such as running one version in production while you simultaneously install and test an upgrade version.

[\(back to contents\)](#)

Installation/Upgrade Process Flow

The following outline describes the order in which the installation- and upgrade-related utilities are used when installing or upgrading Barista Addon.

IMPORTANT

It is vital to your success that you work through each section completely and in order:

- [One: Preparation](#)
- [Two: Installation and Upgrade](#)
- [Three: Testing](#)
- [Four: Go Live - Refresh Data](#)

One: Preparation

1. **Backup.** Perform a complete system backup. In addition, it may be convenient to do a backup of just the barista and aon file systems currently in use.
2. **Java JDK.** Ensure that you have the latest Java JDK installed.
3. **Available disk space.** Ensure that you have sufficient free disk space available for copying Addon, Barista, and any Addon customization projects (including data).
4. **Assumptions.** Ensure that the following installation/upgrade processing assumptions are true:
 - a. For upgrades, the old/source version of Barista Addon **must be a functional version**, i.e., you should be able to launch/run it, and the database must be known to EM and contain all of the necessary User Props (+APDATA, +ARDATA, etc.). to facilitate attaching to the database and doing a dictionary-based data conversion.

- b. Any vertical applications and/or custom modifications to Addon were made using the [Barista Create Application Wizard](#).

This ensures that any forms, code, or data that are not part of the base product are preserved in a separate file structure.

WARNING Prior to version 12.10, changes made under Barista's Default Company (App Company ID = 00000000) and Standard Development (App Product ID = DEV) that are not replicated to a separate application/project are **NOT** upgraded by this process.

- c. **No production applications or data reside in the download location <basis_home>.**

WARNING Production data or programs in the download location may be lost or overwritten during an upgrade.

If you are running Barista or Addon from the download location, you must **first move the existing Barista Addon to an external location prior to installing a new version**. Follow the steps below for [Installation and Upgrade](#) (using the Upgrade option) in order to relocate the existing production Barista Addon out of the download location <basis_home>.

5. **Directory structure.** By default, pre-built Addon stores application images and saved documents under the aon directory (see red text in **Figure 2**). This is not optimal for a production environment.

For production (or development) Addon, a modified structure such as that shown in **Figure 1** makes images and documents version-neutral, so they're accessible via the same pathname across multiple versions of Addon. This structure will require minor changes to the path STBLs defined in your addon.syn file. The process for making these changes is described in the "Run the Update addon.syn File utility" step described in [APPENDIX C - Installation and Upgrade \(Manual Method\)](#).

```
    /<aon_prod>/
      /images/
      /cust_images/
      /documents/
      /<vNN>/ (where NN is the major version number)
        /<vnnnn>/ (where nnnn is the minor version)
          /aon/
          /barista/
          /<cust_mod>/
          /<vert_app>/
```

Figure 1. A file structure such as this is strongly recommended for a production environment.

```
    /<bbj_home>/
      /barista/
        /apps/
          /aon/
            /images/
            /cust_images/prod
            /documents/
```

Figure 2. Pre-built default structure

[\(back to contents\)](#)

Two: Installation and Upgrade

The method described utilizes the automating features of Addon's [Install Wizard](#) and [Upgrade Wizard](#). These two tools combine a series of steps into one easy process for either new installations (Install Wizard) or upgrades (Upgrade Wizard).

Note: Though it is highly recommended that you use Addon's [Install Wizard](#) and [Upgrade Wizard](#) for your process, it is not required. If you choose the manual route, follow the steps listed in this section (Two: Installation and Upgrade) but replace the instructions for the appropriate Wizard with the corresponding manual steps. For information on running the individual steps in a manual fashion, please refer to [APPENDIX C - Installation and Upgrade \(Manual Method\)](#).

1. Lock users out of the existing (old) system by doing the following:

Note: This step is only necessary if upgrading from a previous version of Barista Addon and if users shouldn't be on the existing (old) system during the upgrade process (e.g. it's running in the default download location).

- a. Create a text file named app_maint_msg.txt.

Note: The first line of text is displayed in the system lockout message dialog title bar, and all other text is displayed in the message body. This will prevent new users from logging into the system until the file is removed.

- b. Place it in the barista directory.

2. For upgrades from previous Barista Addon, **STOP** if the old Addon is currently running in the production area (<basis_home>/apps/aon)! Production Addon **must** be relocated out of the BASIS product download location. See the [Assumptions](#) listed in section One: Preparation.

WARNING

Addon in the BASIS product download location **must not** be used for production! Data and modifications in the BASIS product download location are at high risk.

3. Download and install the new BBj/Barista/Addon release.

WARNING

This process will overwrite files in the download location. See the [Assumptions](#) listed in section One: Preparation and **ensure that no production programs/data exist in this directory structure.**

4. For upgrades from a previous Barista Addon, run Barista's [Create Sync File Backup](#) utility for the **current "old"** Addon application.
 - o Skip this step if the old Barista Addon system being upgraded is v12 or newer, since the Create Sync File Backup utility is automatically run by the Upgrade Wizard.
 - o For pre-version 12.00, you must clear the admin_backup directories if they aren't already empty.

Note: In addition to creating the sync backup files for Barista and Addon, this utility also creates them for any vertical and/or custom modification projects.

The sync backup files are created in an admin-backup directory under each application, e.g.:

- o /<old>/barista/admin_backup
- o /<old>/aon/data/admin_backup
- o /<old_cust_mod>/data/admin_backup
- o /<old_vert_app>/data/admin_backup

5. Enable the Barista Installation Manager (BIM).
 - a. Launch the new Barista Addon from the download location.
 - b. At the initial Barista login, click the Settings button.

- c. Select the “Barista Installation Manager” button for the “Launch” option.
6. Launch the new pre-built Addon using Barista Installation Manager (BIM)
 - a. Launch Barista Installation Manager (BIM) for the new Barista Addon in the download location.
 - b. Log in to the BIM.
 - c. Select the pre-built version you just installed.

Note: At this point, the BIM grid will most likely show only a single entry for the pre-built version just installed.
 - d. Press the Run Process button (green arrow) to launch Barista Addon.
 - e. Log in to Barista Addon.
 - f. Since this is a fresh install, you’ll see a welcome message, followed by the Auto-Synchronize Barista utility.
 - g. Click the Synchronize button and answer “Yes” to run the utility to synchronize Addon, the default.syn (and optionally the Examples application) into Barista.

*Note: Because this version of Barista Addon is pre-built, and the forms have already been compiled, do **not** check the box labeled “Compile imported form definitions” so that the synchronization process will run very quickly.*
 - h. Click the Okay button to complete the process.
7. For a new Addon installation (i.e. no existing data) or upgrade from legacy Addon (i.e. existing v6/v7 data),
 - a. Run Addon’s [Install Wizard](#)

Note: Though there is little advantage in doing so, you may manually run the individual utilities instead of using the wizard. The list of utilities automatically processed by the Install Wizard is listed in [APPENDIX D - Understanding the Utilities](#) section. The steps for manually running these utilities are in the [APPENDIX C - Installation and Upgrade \(Manual Method\)](#).

*If you manually run the utilities for a new install (**NOT recommended**), skip this step (“Run Addon’s Install Wizard”), work through the manual steps, and then return to the steps beginning below at “Close out of pre-built Addon”.*
 - i. Enter the New aon Location (i.e. the new install location <new>) that you want for the new copy of Barista/Addon.
 - ii. In the New Firm ID field enter the firm number that will be used by the new Addon installation.
 - iii. Check the “Install demo data” checkbox if you want Addon’s demo data copied to the <new> location.
 - iv. Check the “Install Addon help file” checkbox if you want the <basis_home>/apps/aon/help/addonhelp.jar file copied to the <new> location.
- b. Close out of pre-built Addon.
- c. **If installing version 13.00 or later**, close the Barista Installation Manager. After re-launching the Barista Installation Manager it will list the new Addon installation. **Skip ahead to step 7.g.**

- d. Create a new instance of Barista.
 - i. Launch the Barista Installation Manager if it is closed.
 - ii. Press the New button to create a new instance of Barista.
 - iii. In the Installation ID field, add a unique and descriptive ID.

Note: The Installation ID you supply will become the database name used by this copy of Barista when interfacing with the SQL engine. As such, you should make each Installation ID unique and descriptive (e.g., AON_Vnnnn where nnnn is the version number).
 - iv. In the Location field, supply the name of the parent directory for the new copy of Addon. In other words, if Addon has been copied to /var/aon_prod/vnnnn/aon, the new copy of Barista should be located in /var/aon_prod/vnnnn.
 - v. Click the Create button.
- e. [Specify a session-specific classpath in Enterprise Manager](#) as described in Appendix A. (The Install Wizard does this automatically when installing version 13.00 and later.)
- f. Add the session-specific classpath to the BIM entry for the new (upgraded) install. (The Install Wizard does this automatically when installing version 13.00 and later.)
 - i. Close the Barista Installation Manager, if open.
 - ii. Follow the steps listed in the section entitled [Add the session-specific classpath to Barista Installation Manager \(BIM\)](#).
- g. Manually copy the /<new>/aon/cust_images and /<new>/aon/images directories to a version-neutral location as recommended (illustrated in [Figure 1](#)).
- h. Verify your install by confirming that the following are true.
 - i. Your <new>/aon/config/addon.syn file is correct.
 1. Pathnames are correct for your <new>/aon/ location.

Note: Refer to [Figure 1](#) for the recommended version-neutral file structure.
 2. Your STBL entries for images and saved documents point to a version-neutral location as illustrated in [Figure 1](#).

Note: Use a text editor to make changes to addon.syn.

 - +AON_IMAGES
 - +CUST_IMAGES
 - +DOC_DIR_PDF
 - +DOC_DIR_HTM
 - +DOC_DIR_XML
 - +DOC_DIR_CSV
 - +DOC_DIR_TXT
 - +DOC_DIR_XLS
 - +DOC_DIR_DEFAULT
 - +DOC_DIR_ARCHIVE
 - ii. Pathnames in the various adm_modules*.xml files in the <new>/aon/data/sync/ directory are correct.
 - iii. Your newly-created instance of Barista is working as expected.
 1. Select/Launch the newly-created instance of Barista.

Note: If installing a version prior to 13.00, you will once again be presented with the Welcome screen, followed by the Barista Auto-Sync utility.

2. Run the auto-sync.

- The Auto-Sync will show only one application -- the new copy of Addon -- as a candidate for synchronization unless Addon was installed into an instance of Barista that's already sync'd with other applications.
- If installing a version prior to 13.00, be sure to **check** the box labeled "Compile imported form definitions" so Barista can build the required arc and def files for running in the <new> location. For version 13.00 and later, this has already been done.

Tip: Unless multiple languages are needed, synchronization times can be reduced by unchecking "Build All Active Locales" under the Barista Framework's Development drop-down menu.

- Click the Synchronize button followed by the Yes button to start synchronization.

Note: This copy of Barista is "clean," meaning that both the Barista and corresponding BASIS data dictionary files are empty. Therefore, this Auto-Sync will take longer than the sync performed earlier on the pre-built copy of Addon.

- i. If this is an upgrade from legacy Addon, run DataPort.

8. **Upgrading from a previous version of Barista Addon,**

- a. Run Addon's [Upgrade Wizard](#)

Note: Though there is little advantage in doing so, you may manually run the individual utilities instead of using the wizard. The list of utilities automatically processed by the Upgrade Wizard is listed in [APPENDIX D - Understanding the Utilities](#) section. The steps for manually running these utilities are in the [APPENDIX C - Installation and Upgrade \(Manual Method\)](#).

*If you manually run the utilities for upgrading (**NOT recommended**), skip all of this step for the Upgrade Wizard (i.e. all of "Run Addon's Upgrade Wizard" steps). work through the manual steps. and then continue starting at "[Upgrading on a different server](#)".*

- i. For the "New Database Name" field, add an identifier for the new database (e.g., AON_Vnnnn, where nnnn is the version number). This name will be appear in the list of databases in Enterprise Manager.
- ii. For the "New Description" field, add a description of the new database.
- iii. For the "New aon Location" field, key in or browse to the directory where you want the new copy of Barista Addon (i.e. the new install location <new>).

- iv. For the “Old barista Location” field, browse to or key in the parent directory where the **currently running “old”** Barista/Addon resides (i.e. the old install location <old>).
- v. If the “Old barista location” field and “Old sync Backup Dir” are not filled in automatically, manually enter them.

Note: The “Old barista location” and “Old sync Backup Dir” should be discoverable by the utility so long as the previous Addon was running in the same directory structure as Barista.
- vi. Check the “Use old modified menu” box if you have customized the old Addon menu and wish to continue using that menu rather than the new one.
- vii. Check the “Install” box for existing applications you want the Upgrade Wizard to install in the “new” Barista installation the Wizard will create. (AddonSoftware will automatically be installed, so is not listed in the grid.)

Note: Though the Upgrade Wizard can copy and re-install an existing version of non-Addon Barista applications, upgrading non-Addon applications is beyond the scope of a regular Addon upgrade. The application developer will need to decide if non-Addon Barista applications can be installed into the new instance of Barista from their current install location, or if a new copy/clone of the applications should be created and installed.

“Install” and “Copy”: **How to decide which boxes to check**
Installing an application is essentially creating menu pointers to the code and data. Here are guidelines for deciding whether to install and/or copy an application.

If you want different versions (instances) of Addon to use the same code and data, check “Install” and don’t check “Copy”.

If you want different versions (instances) of Addon to use their own copies of code and data, check “Install” and check “Copy”.

If you want an application to be completely unrelated to the new instance of Addon, do not check either “Install” or “Copy”.

If you want to create copies of code and data in the structure of the new install but do not want to access it until a later date, do not check “Install” and do check “Copy”. (Use case: Wanting a copy available in order to use Git Archive and then bring everything up to date before installing.)

Note: Vertical application <vert_app> and custom modification <cust_mod> projects installed in the “old” Barista are listed in the top grid in the Upgrade Wizard. Paths to the current install

location are shown in the Source column.

viii. Check the “Copy” box for applications you want to make a copy of rather than run from their current install location. You should do this if the new and old installations cannot share the same data files on the file system.

ix. Ensure that the “Target” locations in the top grid are the correct locations for where you want application code copied to (when the “Copy” box is checked) and installed from (when the “Install” box is also checked).

Tip: When the “Copy” box is checked, the Upgrade Wizard copies a selected application from its Source path to its Target path before installing it from the Target path.

IMPORTANT

If Addon callpoint code was changed in a custom modification <cust_mod> project, you should copy the modification project without installing it. This allows you to use the Git archiving process to update the modified callpoint code with changes in the new version of the standard Addon callpoint code before the modification project is installed (see [Using the BASIS Git Archive](#) for information about the archive, and [Upgrading Mods with Git](#) for information on using it). Updated callpoint code must be re-compiled. Barista’s [Install Application Wizard](#), which you will need to run in a later step, does this automatically, therefore avoiding having to manually re-compile each updated callpoint individually.

x. Carefully review the “Target” values in the bottom grid to ensure they are correct for the new installation.

Note: The utility merges together the addon.syn file from <basis_home> with the old addon.syn file. When the same STBL exists in both syn files, the value from the old addon.syn file is initially used for the Target value. If the Target value points to an existing directory or existing file under the “Old aon Location”, the path in the Target value is modified by replacing the “Old aon Location” with the “New aon Location”.

1. Modify the following STBL entries, if applicable, in order to keep images and saved documents in a version-neutral location as illustrated in

[Figure 1](#) above:

- +AON_IMAGES
- +CUST_IMAGES
- +DOC_DIR_PDF
- +DOC_DIR_HTM
- +DOC_DIR_XML
- +DOC_DIR_CSV
- +DOC_DIR_TXT
- +DOC_DIR_XLS

- +DOC_DIR_DEFAULT
 - +DOC_DIR_ARCHIVE
2. Modify the +MDI_TITLE STBL to provide unique MDI title.
- xi. Press the Run Process (green arrow) button.

*Note: The utility creates and syncs new copies of Addon and Barista. This process may take 45 - 60 minutes depending on the speed of your system. **Be sure the process is complete before moving to the next steps.***
- b. **If upgrading on a different server** than the production server, and the database server or port was changed, then update the SET_DBSERVER and SET_DBPORT parameters accordingly in each settings file (e.g. <new>/barista/settings/<language>/*.usr).

Note: This will only need to be addressed if upgrade and testing were done on a separate server than the production server and if the settings files were copied from the test server to the production server.
 - c. Close out of pre-built Addon.
 - d. If you are upgrading manually (NOT recommended), create a new instance of Barista.
 - i. Launch the Barista Installation Manager if it is closed.
 - ii. Press the New button to create a new instance of Barista.
 - iii. In the Installation ID field, add a unique and descriptive ID.

Note: The Installation ID you supply will become the database name used by this copy of Barista when interfacing with the SQL engine. As such, you should make each Installation ID unique and descriptive (e.g., AON_Vnnnn where nnnn is the version number).
 - v. In the Location field, supply the name of the parent directory for the new copy of Addon. In other words, if Addon has been copied to /var/aon_prod/vnnnn/aon, the new copy of Barista should be located in /var/aon_prod/vnnnn.
 - vi. Click the Create button.
 - e. [Specify a session-specific classpath in Enterprise Manager](#) as described in Appendix A. (The Upgrade Wizard does this automatically when upgrading to version 13.00 and later.)
 - f. Add the session-specific classpath to the BIM entry for the new (upgraded) install. (The Upgrade Wizard does this automatically when upgrading to version 13.00 and later.)
 - i. Close the Barista Installation Manager, if open.
 - ii. Follow the steps listed in the section entitled [Add the session-specific classpath to Barista Installation Manager \(BIM\)](#).
 - g. Launch the new instance of Barista Addon from the BIM.
 - h. If you want to re-install one or more Barista vertical applications <vert_app> and/or custom modifications <cust_mod> projects, you'll need to make copies/clones of those projects (particularly if you need to keep running your current application while performing and testing the upgrade).
 - i. Use Addon's [Copy Application](#) utility to copy <vert_app> and <cust_mod> projects from their <old> install location to the <new> install location (e.g., /var/aon_prod/vnnnn/).

Note: This is only necessary for applications or customizations that were not [selected to be copied](#) when filling out the upper grid of the Addon Upgrade Wizard earlier.)

3. In the “Source syn File” field select the old .syn file using the find button or type it in (e.g. /<old_cust_mod>/config/mymods.syn).
 4. In the “New config Location” field, use the browse button to locate the root directory for the new app (e.g., /var/aon_prod/vnnnn/).
 5. Create a new folder for the project (e.g., <cust_mod>) in the new app’s root directory.
 6. Now select the newly created folder (e.g., /var/aon_prod/vnnnn/<cust_mod>).
 7. Modify the target STBL values as needed.
- ii. If Addon callpoint code was changed in a custom modification <cust_mod> project, you should update the modified callpoint code with changes in the new version of the standard Addon callpoint code.

BASIS provides a Git archive to assist with, and automate, much of this code merging task (including backend code). Refer to [Using the BASIS Git Archive](#) for information about the archive, and to [Upgrading Mods with Git](#) for information on using it.

Note: Updated callpoint code must be re-compiled. Running Barista’s Install Application Wizard in the following step does this automatically.

- iii. Sync the new copies of <vert_app> and/or <cust_mod> projects into Barista
1. Launch the new instance of Barista Addon from the BIM.
 2. Run Barista’s [Install Application Wizard](#).

Note: This step is necessary to sync the new copies of <vert_app> and/or <cust_mod> projects that were [not selected to be installed](#) via the Upgrade Wizard when filling out the upper grid.

Caution

If multiple applications are being installed, the order in which they are installed may be important. For example, if a <cust_mod> project uses data elements defined in a <vert_app>, then the <vert_app> needs to be installed before the <cust_mod> project.

3. In the “Sync File ID” field, be sure to select the .syn file from the **<new>** install location (e.g., /var/aon_prod/vnnnn/<cust_mod>/config/mymods.syn).

- iv. Use a text editor to update the <new> barista.cfg (e.g., /usr/aon_prod/vnnnnn/barista/sys/config/enu/barista.cfg) as needed so that the <vert_app> and <cust_mod> project PREFIXs come **before** the Addon PREFIXs.
- i. Copy data into the new install for testing.

WARNING

Skipping this part of the process may compromise your ability to adequately test your upgrade.

Note: It is recommended that you copy the Addon data files from the <old> install location to the <new> install location (particularly if you need to keep running your old Addon while performing and testing the new Addon installation).

There are many strategies and ways for doing this. A common approach is described below. The key thing is to use the BASIS Enterprise Manager - Database Update Wizard to “copy and upgrade” files that were structurally changed between the old version and the new version.

- i. Run the EM Update Wizard utility from the old/source database.
 1. Add com.basis.ocj.debug=true to BBJ.properties, and restart BBJServices.
 2. Launch Enterprise Manager.
 3. Click on the <old> database listed in the “Databases” pane.
 4. Click on the “Update Wizard” button on the “Information” tab for the <old> database.
 5. Click the “Next” button on the “Welcome” panel.
 6. On the “Specify New Dictionary and Destination” panel:
 - For the “New Database” field, leave it blank.
 - For the “New Data Dictionary” field, key in or browse to the location of the new dictionary (/<new>/barista/bbdict).
 - For the “Updated Files Destination” field, key in or browse to the location of the new data folder (/<new>/aon/data).
 - **Do NOT check the box labeled “Copy Only Changed Table Data Files”.**
 - Click the Next button.
 7. On the “Specify User Property Destinations” panel:
 - Change all Destination Locations that defaulted to “./data”, to “.” (just a dot, i.e. use the Updated Files Destination) instead.
 - Click the Next button.

Note: A few minutes will elapse while the Wizard processes.
 8. Click “Next” on the following series of 7 panels.

*Note: You’ll be presented a series of panels asking if you want to copy things in the old dictionary that aren’t present in the new dictionary. Do **not** select anything on any of*

these panels.

9. Click “Finish” on the final panel.

Note: The wizard will begin a dictionary-based version of the EM Online Copy Job.

10. Select the “Online Copy Jobs” task from the EM task pane.

11. Monitor the progress of the Online Copy Job from Enterprise Manager-- using the Refresh button to view the progress.

Note: It may take a few moments before the job appears in the list.

12. When the job reaches 100%,

- Click the Refresh button.
- **Expand the list of files and look for any errors, as files with errors are not copied.**
- Resolve any errors.

13. Click “Finish”.

Note: Rather than a simple file copy, the wizard-based Online Copy Job compares the old and new data dictionaries, and copies the old data to the new location while making any structural modifications (different/additional keys, new fields, etc.).

The Online Copy Job processes the entire Barista database (unless the “Copy Only Changed Table Data Files” box is checked), so it will copy/convert not only Addon files, but also any files from <vert_app> and <cust_mod> projects that were sync'd into the old Barista.

Because the Update Wizard uses the Online Copy Job, it can copy data while the source files are still in use, and keep the source and destination files in sync until you indicate that the job is finished. This makes it possible to convert large databases without having to shut users out of the system.

- ii. Run Addon's [Distribute EM Update Wizard Files](#) utility.

Note: This step will remove redundant Barista files from the /<new>/aon/data/ directory, and distribute files for <vert_app> and/or <cust_mod> projects to their data directories.

The EM Online Copy Job puts files in the Destination Locations entered in the “Specify User Property Destinations” panel, which should resolve to /<new>/aon/data This is the wrong location for <vert_app> and <cust_mod> projects. The Distribute EM Update Wizard Files utility moves all the data files to their correct data directories.

1. Launch this wizard from the new instance of Barista Addon.
 2. In the “Updated Files Location” field, key in or browse to the location where the EM Online Copy Job put the updated files (/<new>/aon/data).
 3. In the “New Database Name” field, key in name of the new database (the application ID listed in Barista Installation Manager).
9. If triggers are used with code being upgraded, then the triggers will likely need to be redefined in BASIS’s Enterprise Manager (EM) with updated paths. And, references to old paths in trigger source code also need to be updated. If trigger definition or source files are modified, ensure the changes aren’t overwritten by subsequent file copies and/or rsyncs.
- a. On the EM Triggers tab, add a new File Triggers Directory that points to the new data file location where the *.trigger files are located.
 - b. For each trigger file listed in the new File Triggers Directory, it may be necessary to update the location of the *.src file using the Trigger Editor.
 - c. On the EM Trigger Configuration tab, update the Config File and Working Directory for the new installation. For example:
 - i. Config File: <new>/barista/sys/config/enu/barista.cfg
 - ii. Working Directory: <new>/barista
10. Check the AddonSoftware Administration -> Utilities -> Install/Upgrade Utilities -> Version Conversion menu for any other version-specific utilities.
11. If users were locked out of the old system for the upgrade, remove the system lockout message file app_maint_msg.txt in the barista directory.

[\(back to contents\)](#)

Three: Testing

Use the following list to test that the install/upgrade was successful. Augment the list with tests specific to your particular install.

1. Verify that all .syn files (for addon, <vert_app>, and/or <cust_mod>) contain the correct pathnames for STBL, CFG and PREFIX entries.
2. Verify PREFIX in new barista.cfg file. Adjust if/as needed to ensure that <vert_app> and <cust_mod> prefix pathnames come before Addon prefixes, if required.
3. ***Recommended but optional***
Create a separate document location for testing.
 - a. Create a directory for test documents.
 - b. Edit the +DOC_DIR* STBLs in the addon.syn file to point to the test document dir.
 - +DOC_DIR_PDF
 - +DOC_DIR_HTM
 - +DOC_DIR_XML
 - +DOC_DIR_CSV
 - +DOC_DIR_TXT
 - +DOC_DIR_XLS
 - +DOC_DIR_DEFAULT
 - +DOC_DIR_ARCHIVE
 - c. Run auto-sync.
4. Prevent unwanted fax/emails being sent while testing by not starting the [Document Processing Queue](#) for the /<new>/aon instance.

- a. If testing fax/emails, care must be taken to not send test documents to customers and vendors.
 - b. Before starting the [Document Processing Queue](#) for the /<new>/aon instance, either cancel documents or change recipient fax/email information as needed to not send to customers and vendors.
 - c. Stop the [Document Processing Queue](#) for the /<new>/aon instance when current batch of test have been sent.
5. Verify that the SQL engine is finding the new data by using the binoculars and/or magnifying glass in a form.
 6. Verify that forms and callpoints are built in the correct locations. The simplest way to do this is by making a benign callpoint change, such as adding a remark, and then building the form/callpoint.
 7. Create user shortcuts to launch the new Barista Addon.

Note: For shortcuts that launch Barista Addon directly, bypassing BIM, create the shortcuts using the information listed in the Properties field of BIM's Create/Modify Barista Installation form.
 8. Test Addon and any <vert_app> and/or <cust_mods> programs to verify that all is working correctly.

Note: If you have <cust_mods> that involved changes to Addon forms, make sure to run them and check formatting. Some forms may need to be rebuilt.
 9. If info(3,2) is the desired default when logging into Barista (rather than "admin"),
 - a. Edit addon.syn to include the following line:
SYSSTBL=SET +MASTER_USER=
 - b. Re-run Auto-Synchronize Barista for addon.syn only.

Note: This is a one-time change, as the value will come forward from the "old" addon.syn on subsequent upgrades.
 10. Review BUI Application information in Enterprise Manager, if applicable, to ensure that BUI launch configurations for the MDI/Application as a whole, or for any individual tasks, are present and correct for both old and new instances.

[\(back to contents\)](#)

Four: Go Live - Refresh Data

Work through the section appropriate to the type of upgrade/installation you are doing:

- [Upgrading from a previous version of Barista Addon](#)
- [Upgrading from legacy Addon or non-Addon application](#)
- [New Installation](#)

Upgrading from previous version of Barista Addon

1. Delete **test** data from all application data directories (addon, <vert_app>, <cust_mods>).

WARNING Be careful to **NOT** delete live production data. This step is for **test data only**.

2. Use the Enterprise Manager [Database Update Wizard](#) to once again copy <old> production data (including <vert_app> and <cust_mod>) to <new> location.

WARNING Do **NOT** check the “Copy Only Changed Table Data Files” checkbox.

When the Update Wizard’s copy job reaches 100%:

- a. Make sure everyone is off <old> system.
 - b. Click the Refresh button.
 - c. **Expand the list of files and look for any errors/failures, as files with errors are not copied.**
 - d. Finish the copy job and resolve any errors.
3. Re-run Addon utility to [Distribute EM Update Wizard Files](#) on <new> system.
 4. (Optional) In Enterprise Manager, do the following (in order) to preserve the name of the current production database:

Note: One use case for this is preserving a static database name through upgrades for the benefit of users querying the system via SQL. Using the methodology outlined below, you create a version-neutral database name referencing the current database, and then “roll” that database after an upgrade.

 - a. (Optional) If a backup database was created for a previous upgrade, drop it, deleting its dictionary and data files.
 - b. (Optional) Create (attach to existing dictionary) a backup database of the current production database
 - i. Add necessary User Props to point it to the dictionary and data files currently used by the production database.
 - ii. Set Advisory Locking.
 - c. Drop the current production database. **Do NOT delete its dictionary and data files.**
 - d. Create new database (attach to existing dictionary)
 - i. Use the same name as the previous production database (i.e. the database dropped in previous step)
 - ii. Add necessary User Props to point it to the dictionary and data files used by the new database created during this upgrade (i.e the “New Database Name” used when you ran the Addon Upgrade Wizard, or the “Installation ID” used when you [created the new instance of Barista](#) using the Barista Install Manager).
 - iii. Set Advisory Locking.
 - e. Drop the new database created during the upgrade (i.e the “New Database Name” used when you ran the Addon Upgrade Wizard, or the “Installation ID” used when you [created the new instance of Barista](#) using the Barista Install Manager). **Do NOT delete its dictionary and data files.**
 - f. Use a text editor to update +DBNAME and +DBNAME_API in the <new> barista.cfg (e.g., /usr/aon_prod/vnnnnn/barista/sys/config/enu/barista.cfg) to point to the production database (i.e. the database created in the previous “[create new database](#)” step).
 5. Refresh Barista Administrative data
 - a. Re-run Barista’s [Create Sync File Backup](#) utility on <old> system.

- b. Rename <old> data directory as a safeguard against accidental access, or use app_maint_msg.txt [lockout file](#) as described at the beginning of section Two: Install and Upgrade.
 - c. Remove shortcuts to the <old> application(s).
 - d. Re-run Addon's utility to [Update sync XML Files](#) to <new> install location. In addition to running this utility for Addon, it also needs to be run separately for each installed vertical application <vert_app> and custom modification <cust_mod> project.
 - e. Re-run [Auto-Synchronize Barista](#) utility on <new> system.
6. Check the AddonSoftware Administration -> Utilities -> Install/Upgrade Utilities -> Version Conversion menu for any other version-specific utilities.
 7. If you chose to [create a separate document location](#) for testing, re-initialize the document location to your production documents directory per [Figure 1](#).

WARNING

Do NOT delete <old> IMAGES or <old> DOCUMENTS if used by <new> system.

- c. Edit the +DOC_DIR* STBLs in the addon.syn file to point to the production document dir.
 - o +DOC_DIR_PDF
 - o +DOC_DIR_HTM
 - o +DOC_DIR_XML
 - o +DOC_DIR_CSV
 - o +DOC_DIR_TXT
 - o +DOC_DIR_XLS
 - o +DOC_DIR_DEFAULT
 - o +DOC_DIR_ARCHIVE
- d. Run auto-sync.
7. If sending fax or [emails](#), (1) cancel all test documents in the [Document Processing Queue](#), (2) stop the [Document Processing Queue](#) for the /<old>/aon instance, then (3) start the [Document Processing Queue](#) for the /<new>/aon instance.

[\(back to contents\)](#)

Upgrading from legacy Addon or non-Addon application

1. Delete test data from all application data directories (addon, <vert_app>, <cust_mods>).
2. Ensure that everyone is off the <old> system.
3. Re-run [DataPort](#) or other conversion utility to create fresh copy of Addon data.
4. Copy/Paste data from <vert_app> and/or <cust_mods> from <old> back into <new> location.

Note: If the volume of data is large, you may elect to use [Enterprise Manager - Online Copy Job](#).
5. When new copy of data is in place:
 - a. If EM copy job was used, when the job reaches 100%
 - i. Click the Refresh button.
 - ii. Expand the list of files and look for any errors, as files with errors are not copied.
 - iii. Resolve any errors.

- iv. Click "Finish".
 - b. Rename <old> data directory as a safeguard against accidental access.
 - c. Remove shortcuts to the <old> application(s).
6. Refresh Barista Administrative data
 - a. Re-run [Auto-Synchronize Barista](#) utility on <new> system to restore administrative records to pre-test values.
7. Check the AddonSoftware Administration -> Utilities -> Install/Upgrade Utilities -> Version Conversion menu for any other version-specific utilities.
8. If you chose to [create a separate document location](#) for testing, re-initialize document location to your production documents directory per [Figure 1](#).

WARNING

Do NOT delete <old> IMAGES or <old> DOCUMENTS if used by <new> system.

- a. Edit the +DOC_DIR* STBLs in the addon.syn file to point to the test document dir.
 - +DOC_DIR_PDF
 - +DOC_DIR_HTM
 - +DOC_DIR_XML
 - +DOC_DIR_CSV
 - +DOC_DIR_TXT
 - +DOC_DIR_XLS
 - +DOC_DIR_DEFAULT
 - +DOC_DIR_ARCHIVE
 - a. Run auto-sync.
9. If sending fax or [emails](#), start the Document Processing Queue Monitor either from within Barista or as an AutoRun task in Enterprise Manager.

[\(back to contents\)](#)

New Installation

1. Delete test data.
2. Refresh Barista Administrative data
 - a. Re-run [Auto-Synchronize Barista](#) utility to restore administrative records to pre-test values.
3. If you chose to [create a separate document location](#) for testing, re-initialize document location to your production documents directory per [Figure 1](#).

WARNING

Do NOT delete <old> IMAGES or <old> DOCUMENTS if used by <new> system.

- a. Edit the +DOC_DIR* STBLs in the addon.syn file to point to the test document dir.
 - +DOC_DIR_PDF
 - +DOC_DIR_HTM
 - +DOC_DIR_XML
 - +DOC_DIR_CSV

- +DOC_DIR_TXT
 - +DOC_DIR_XLS
 - +DOC_DIR_DEFAULT
 - +DOC_DIR_ARCHIVE
- b. Run auto-sync.
4. If sending fax or [emails](#), start the Document Processing Queue Monitor either from within Barista or as an AutoRun task in Enterprise Manager.

[\(back to contents\)](#)

APPENDIX A - Understanding the Java Classpath

Java Classpath Settings are defined in the BASIS BBJ Enterprise Manager. Each classpath includes one or more directory paths that allow BBJ to find Java jars (“Java ARchive”) and classes. These jars contain language resource bundles and are required for localization in Jasper-based reports. They are necessary for proper functioning of Barista Addon regardless of language/locale used. For Addon, these include Barista jars such as baristaprop.jar, and Addon jars such as addonprop.jar.

Each installation of Barista Addon requires a Java Classpath definition in Enterprise Manager, and a property definition in the Barista Install Manager pointing to that classpath. The following two sections describe each of these processes.

Specify a session-specific classpath in Enterprise Manager.

1. Log into Enterprise Manager.
2. Navigate to the Classpath tab for Server Information.
3. Add a new Java Classpath Settings entry that contains the pathname to the addonprop.jar file in the <new>/aon/prop/ dir.

Tip: The classpath name is user-defined, but a good convention would be to use the same "AON_Vnnnn" name supplied for the new database. See [Session-specific Classpath](#) for more information.

Add the session-specific classpath to Barista Installation Manager (BIM).

1. Close out of pre-built Addon.
2. Launch the Barista Installation Manager if it is closed.
3. Select the installation to edit.
4. Press the Edit button to bring up the edit window.
5. In the Properties field, prepend an argument referencing the session-specific classpath name listed on the Classpath tab in Enterprise manager (e.g., -CPAON_Vnnnn).

Note: The -CP literal is mandatory. It is followed by the name of the classpath and then a space.

6. Click the Save button to save your changes and close the window.

[\(back to contents\)](#)

APPENDIX B - Understanding How Upgrades Handle Menus

Currently menus are not merged as part of the upgrade process. Whether you use the Addon Upgrade Wizard utility or manually perform the upgrade steps, you need to handle customized menus from the old system.

Note: The basic menu file is named `addon.men`. It resides in this path: `<install_name>/aon/config/<language>/`

The Addon Upgrade Wizard has a checkbox option for you to direct how the utility treats menus. Here is a summary of how it works:

1. **If you do not** check the “Use old modified menu” option in the Addon Upgrade Wizard
 - `addon.men` from the `<old>` install location will remain in place.
 - `addon.men` from the Addon upgrade will be placed into the `<new>` install location.
 - Result: The menus displayed in the “new” install will be the current Addon standard.
2. **If you do** check the “Use old modified menu”,
 - The current Addon standard `addon.men` will be renamed to `addon.men.bak`.
 - `addon.men` from the `<old>` install location will be copied to your `<new>` install location.
 - Result: The menus displayed in the “new” install will match those of the “old” install.

If you are upgrading manually, you will have to do the menu copying by hand. These steps describe the process:

1. `<Optional>` Save a copy of the new `addon.men` files so you can run a diff in case you want to incorporate any standard menu changes into your customized menus.
2. Copy the `addon.men` files from the `<old>` install location to the `<new>` install location (e.g., `/var/aon_prod/vnnnn/aon/config/<language>/addon.men`).

Either way, you need to decide whether to use the old, customized menu, the new Addon standard, or a combination. It comes down to what will require the least effort. Consider the following questions:

- How heavily modified is the old menu?
- How heavily modified is the new, standard menu?
- How many of the changes in the new standard menu pertain to how the particular install is used? (For example, if the changes are all to a module that is not used in the install, your old customized menu might be the best choice.)

Tip: You can diff the copies of `addon.men` to help you answer the questions. For example, if you installed the standard Addon menu, compare `addon.men.bak` to `<new>/aon/config/<language>/addon.men` to see if the `.bak` (new standard menu) file contains any changes you want to incorporate into your menu. █

Once you have decided which menu to use as a base, use Menus task (from the Maintenance submenu of the Barista Administration menu) to incorporate any menu changes.

[\(back to contents\)](#)

APPENDIX C - Installation and Upgrade (Manual Method)

The following outline describes steps that one would need to run if they were not using the recommended Addon Wizards for upgrading or installing Barista Addon. The order in which they are listed is important for success. If you chose to use these steps instead of running the appropriate Addon Wizard, replace the Addon Wizard steps with these, but **be sure to also complete the steps preceding and following the Addon Wizard** (see section [Two: Installation and Upgrade](#)).

Note: Execute the following manual steps using the pre-built version of Addon installed in <basis_home>.

1. Run the [Copy aon Directory](#) utility to copy the aon directory from the download location <basis_home> to the new install location <new>. The utility provides a default path for the new installation. This can be overridden, but it is recommended that all production versions of Addon be installed under a common top-level directory. For example, `/usr/aon_prod/vnnnn` where nnnn is the version number.
2. Manually copy the `/<new>/aon/cust_images` directory to a version-neutral location as illustrated in [Figure 1](#).
3. Run the [Update addon.syn File](#) utility.

Note: This tool updates pathnames in the `/<new>/aon/config/addon.syn` file.

- a. In the “Update addon.syn File” field select the new addon.syn file using the find button or type it in, e.g. `/<new>/aon/config/addon.syn`.
- b. For upgrades, you may opt to merge the contents of your current addon.syn file with the new one in order to migrate any custom STBLs or changed values. In this case, do the following:
 - i. Check the “Merge with old addon.syn file” box.
 - ii. In the “Old addon.syn File” field, select the old addon.syn file using the find button or type it in, e.g. `/<old>/aon/config/addon.syn`.
- c. Modify the following STBL entries, if applicable, in order to keep images and saved documents in a version-neutral location as illustrated in [Figure 1](#):
 - +AON_IMAGES
 - +CUST_IMAGES
 - +DOC_DIR_PDF
 - +DOC_DIR_HTM
 - +DOC_DIR_XML
 - +DOC_DIR_CSV
 - +DOC_DIR_TXT
 - +DOC_DIR_XLS
 - +DOC_DIR_DEFAULT
 - +DOC_DIR_ARCHIVE

- d. Modify the +MDI_TITLE STBL to provide a unique MDI title.
4. Run the Addon [Update sync XML Files](#) utility.

Note: This tool copies Barista Administrative data that was backed-up using Barista's [Create Sync File Backup](#) utility to the <new> install location. It also updates the locale_path and dd_mount_dir values inside the various adm_modules~.xml files in <new> install location.*

 - a. In the “New sync Location” field, browse to or keyin the new sync directory (e.g. /<new>/aon/data/sync).
 - b. For upgrades
 - i. Check “For upgrade, sync backed up?”
 - ii. In the “Old sync Location” field select the old sync directory using the find button or type it in, e.g. /<old>/aon/data/sync.
 - iii. If the sync backup was created with a Barista version prior to v11, in the “Old sync Backup Dir” field, select the directory you used to create the backup or type it in (e.g. /<my_temp_backup>).

Note: If the sync backup was created with v11 Barista or later, the “Old sync Backup Dir” field is disabled.
 5. If you are upgrading from a previous version of Barista Addon
 - a. If you want to retain user settings information (form coordinates and sizes, inquiry column settings, etc.), do the following:
 - i. Run the [Copy Directory Recursively](#) utility to copy the <old>/barista/settings directory to <new>/barista/settings.
 - b. If you have customized the old Addon menu, and wish to continue using that menu rather than the new one, then do the following:
 - i. <Optional> Save a copy of the new addon.men files so you can run a diff in case you want to incorporate any standard menu changes into your customized menus.
 - ii. Copy the addon.men files from the <old> install location to the <new> install location (e.g., /var/aon_prod/vnnnn/aon/config/<language>/addon.men).
 - iii. See [APPENDIX B - Understanding How Upgrades Handle Menus](#) for more information.
 6. If this is a new installation, run the [Set Up New Firm](#) utility to create a new firm in Addon.
 - a. Verify the Data Location is correct.
 - b. Enter the desired New Firm ID (e.g. 01 or 02).

[\(back to contents\)](#)

APPENDIX D - Understanding the Utilities

Both Addon and Barista include utilities to assist with the installation and upgrade processes. This section describes the various utilities. Many of these have been knit together in the Addon Installation/Upgrade Wizards (AIUW). But it is possible, and may sometimes be necessary, to run one or more of these utilities separately. Each is a separate menu item, and they must be run in the order outlined in the [Installation/Upgrade Process Flow](#) section of this document.

Barista Utilities

Barista provides four utilities to facilitate the installation/upgrade processes. They are found on different menus as noted in each section.

- **Create Sync File Backup**

Description:

The Create Sync File Backup utility creates xml versions of all administrative data so that the various BBJ tables can be re-populated after a re-install or upgrade.

Details:

Barista maintains administrative data in a series of BBJ files within the Barista directory structure. Some of these records are “branded,” meaning they are specific to an application and carry a Company and Product ID (e.g., system messages, masks, sequence numbers, etc.). Others are “non-branded,” used by Barista itself (e.g., users, countries, archived document data, etc.).

The branded xml files are created within the file structure corresponding to the company/product ID in the record. Non-branded xml files are created and stored in the admin_backup directory inside the Barista file system (version 11), in a user-specified directory (version 10), or in the same directory as the branded files (version 9).

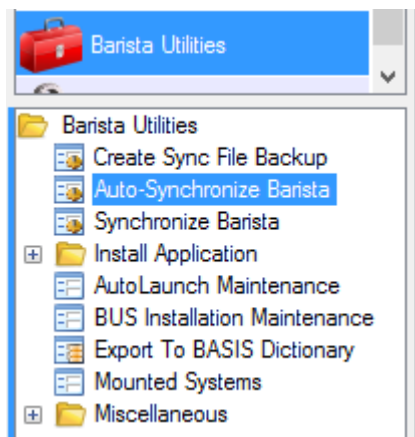
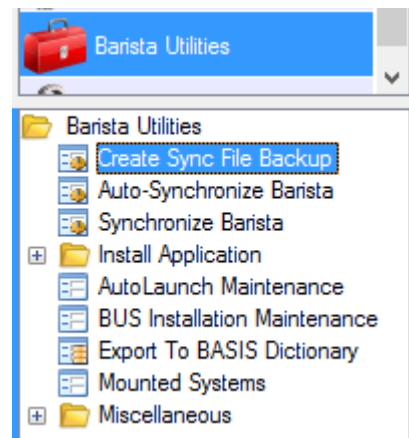
The [Create Sync File Backup](#) utility **should *always*** be run before a re-install or upgrade. ([back to contents](#))

- **Auto-Synchronize Barista**

Description:

The Auto-Synchronize utility builds tables and menus, maintains config files, updates the BASIS data dictionary, compiles callpoint code for forms, and optionally rebuilds forms.

Details:



On the first launch of a new Barista installation, Barista examines its parent directory structure, as well as the contents of the barista.syn file (if present) to locate any application .syn files. These .syn files are presented as candidates for synchronization in the Auto-Synchronize process (recall that a given instance of Barista can support or be sync'd with one or more applications). The Auto-Synchronize process performs the following for each application selected:

- Builds/restores Barista administrative tables from the contents of the xml's created during the [Create Sync File Backup](#).
- Updates the barista.cfg and BASIS config.ini files with information from selected application .syn files.
- Imports any application menus.
- Builds the Barista dictionary tables from the contents of Barista resource files in selected application/data/bar directories.
- Updates the corresponding BASIS data dictionary.
- Compiles callpoint code.
- Optionally, rebuilds any forms referenced in the Barista resource files.

The Auto-Synchronize process can also be launched manually from the Barista Utilities menu. This is typically only necessary if new information is added to an application's .syn file and needs to be "pushed" into barista.cfg. Since the Auto-Sync has already run to completion once, the subsequent invocations will only perform the first three steps above (known as a "quick-sync").

[\(back to contents\)](#)

- **Install Application Wizard**

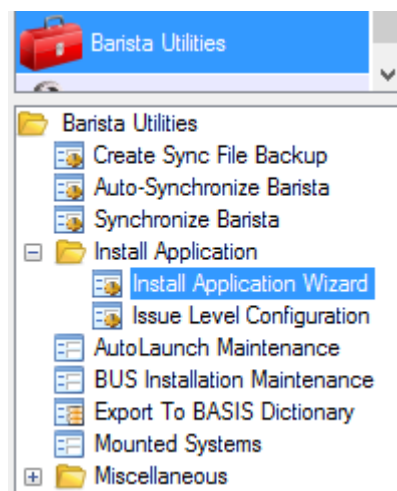
Description:

The Barista Install Application Wizard syncs in applications not already incorporated into Barista. It does this by performing the same basic tasks as the Auto-Sync as well as reporting potential data conflicts, and merging in tables.

Details:

Any application(s) not already sync'd into Barista by virtue of the Auto-Synchronization process can be incorporated into a given instance of Barista by running this wizard. The Barista Install Application Wizard performs the same basic tasks as the Auto-Sync, but also analyzes each Barista resource file prior to the import to check for potential conflicts with data currently in the Barista dictionary. Any conflicts are presented in an issue report, and must either be resolved (typically by making changes to the incoming application), or set to "low-level" in order for the install to continue.

In addition, the Barista Install Application Wizard will merge data from the incoming Barista resource files with the current data dictionary, and will also rebuild physical data tables if the imported application contains new or changed table columns, keys, etc. This makes it



possible to customize an application, re-install the base application, and then re-install the customizations.

WARNING

Any modification needing to be preserved across upgrades needs to be made in an application/project created via the Create Application Wizard. Prior to version 12.10, modifications made using Barista's Default Company (App Company ID = 00000000) and Standard Development (App Product ID = DEV) can **NOT** be re-installed via the Barista Install Application Wizard.

Note: While table definitions should be correct after installing an application, forms may need to be adjusted and rebuilt via Form Designer, depending on the various properties (tab order, x/y position, etc.) of the merged field(s).

[\(back to contents\)](#)

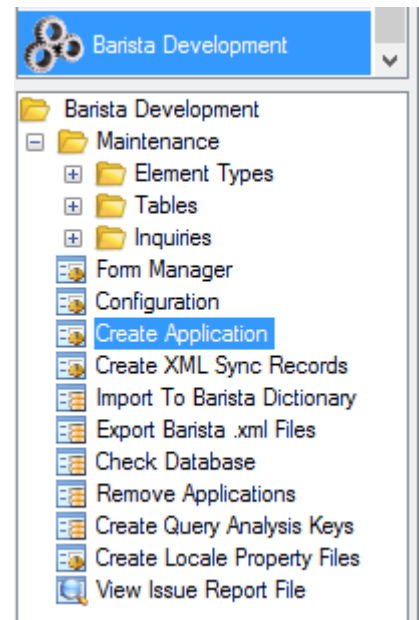
- **Create Application Wizard**

Description:

The Create Application Wizard guides you through the setup of a new Barista Application/Project file structure. Use this wizard to create vertical applications, or to create an application to store customizations to a base product such as Addon.

Details:

Barista applications exist in their own file structure outside of the Barista filesystem. Using this utility preserves customizations in their own file structure so that they can be re-installed after upgrading the base application.



WARNING

Any modification needing to be preserved across upgrades needs to be made in an application/project created via the Create Application Wizard. Prior to version 12.10, modifications made using Barista's Default Company (App Company ID = 00000000) and Standard Development (App Product ID = DEV) can **NOT** be re-installed via the Barista Install Application Wizard.

Starting with Barista Addon version 11.00, **you should always establish a parent/child relationship when creating a project for customizations to a base application.** In other words, if you intend to customize Addon, you should create a new project and specify the <install>/aon/config/addon.syn as the parent application. Establishing this relationship allows Barista to prompt you for your child project when making modifications via Form Manager, ensuring that your modifications are replicated/saved to the child project and helping with future upgrades.

[\(back to contents\)](#)

Addon Utilities

Addon provides multiple utilities to facilitate the installation/upgrade processes. They are found on individual menus as shown to the right and noted in each section.

Install/Upgrade Utilities

- **Install Wizard**

Description:

The Addon Install Wizard simplifies creating a new Addon installation outside the Basis download location <basis_home>.

Details:

This wizard automatically runs the following utilities so they don't have to be run individually:

- [Copy aon Directory](#)
- [Update addon.syn File](#)
- [Update sync XML Files](#)
- [Set Up New Firm](#)

[\(back to contents\)](#)

- **Upgrade Wizard**

Description:

The Addon Upgrade Wizard simplifies upgrading an existing Addon installation to a newer version of Barista Addon.

Details:

This wizard automatically runs the following utilities so they don't have to be run individually:

- [Copy aon Directory](#)
- [Update addon.syn File](#)
- [Barista Installation Manager's Create Barista Installation](#)
- [Update sync XML Files](#)
- [Auto-Synchronize Barista](#)

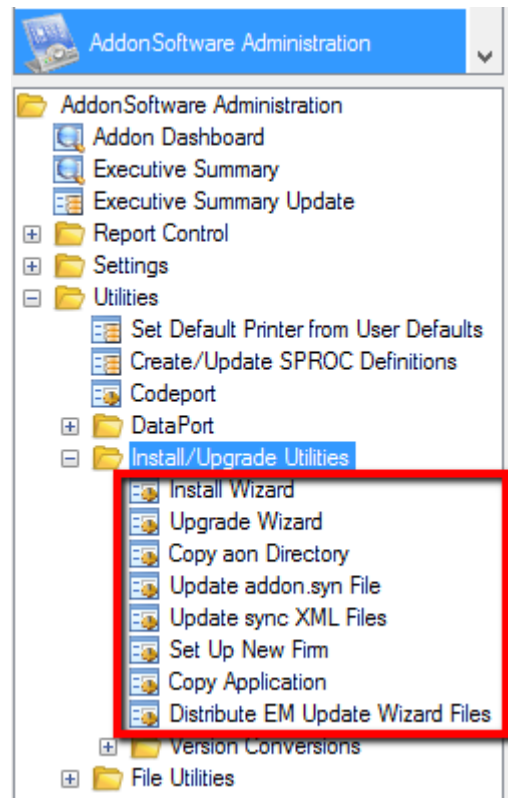
[\(back to contents\)](#)

- **Copy aon Directory**

Description:

This utility makes a physical copy of the Addon application in a location outside of the installation directory. It is the first step to correcting an install that was mistakenly installed in the download area (<basis_home>/apps/aon).

Details:



Pre-built Addon is installed in the home directory relative to your installation of BBj (<basis_home>/apps/aon). Regardless of what Barista installation you run this utility from, it copies the bulk of the Addon install (the Addon programs, images, and a minimal set of data files) from that default home directory.

A default pathname for the new copy of Addon is provided and can be overridden if desired. In addition, you can specify whether or not you want to copy the demo data and help files. (Help files need not be copied so long as a running copy of Addon remains in the default installation directory.)

Consider the following when deciding whether or not to copy demo data:

- If upgrading from a previous version of Barista Addon, you typically won't want the demo data, as you'll be copying/converting existing production data.
- Copy the demo data and use the [Set Up New Firm](#) utility to clear the demo records and create new defaults from the firm 99 data if you are:
 - Upgrading from Addon version 6 or 7 and plan to convert the legacy data using the [DataPort](#) utility
 - Upgrading from a non-Addon application to Addon and will be converting the non-Addon data with a custom tool
 - Installing Addon for the first time and have no previous data (In this case, can use the Set Up New Firm utility instead of the Clear Firm from System and Copy Data from Firm to Firm utilities.)

Note: The utility does not copy any files in the <basis_home>/apps/aon/data/arc/, <basis_home>/apps/aon/data/def/*, or <basis_home>/apps/aon/callpoints* directories, as these files will all be recompiled when synchronizing the new Addon with Barista.*

Tip: Use the [Update addon.syn File](#) utility to adjust the relative pathnames of the addon.syn file copy that gets created by this task.

[\(back to contents\)](#)

- **Update addon.syn File**

Description:

This utility provides an interface to adjust STBL and PREFIX values (such as paths) for a specified addon.syn file. If executed as part of the Addon [Install Wizard](#) or the Addon [Upgrade Wizard](#), this utility defaults all of the “../apps/aon” pathnames to the location selected during the Copy aon Directory.

Details:

The addon.syn file copied during the [Copy aon Directory](#) utility contains relative pathnames that may or may not resolve correctly when Barista Addon is installed in a new location. The Update addon.syn File utility defaults all of the “../apps/aon” pathnames to the location selected during the Copy aon Directory. If you will be placing data files in some other location, you can override the defaults.

In addition, if upgrading from a prior version of Barista Addon, you can opt to merge values from your previous addon.syn file into the new one, preserving values you may have changed or added.

[\(back to contents\)](#)

- **Update sync XML Files**

Description:

This utility corrects the pathnames referenced in adm_modules*.xml files in a specified data/sync directory. **!**

Details:

The Update sync XML Files utility corrects the pathnames referenced in the adm_modules*.xml files so that they contain the pathname for the new location.

For upgrades, records created during the Barista [Create Sync File Backup](#) are copied to the new Addon location so that the [Auto-Synchronize Barista](#) process will be able to restore the various Barista Administrative tables (masks, messages, sequence numbers, users, etc.) to their last-used state.

[\(back to contents\)](#)

- **Set Up New Firm**

Description:

This utility facilitates setting up a new firm in an Addon installation.

Details:

The Set Up New Firm utility should be used instead of the Clear Firm from System and Copy Data from Firm to Firm in order to ensure the proper Barista files are updated.

[\(back to contents\)](#)

- **Copy Application**

Description:

This utility copies an application (vertical or customization project) to a new directory location, and updates *.syn and adm_modules*.xml files for that location.

Details:

Besides copying the application to a new location, the Copy Application utility also provides an interface to adjust STBL and PREFIX values (such as paths) for the specified *.syn file. And, it corrects the pathnames referenced in the adm_modules*.xml files so that they contain the pathname for the new location. Afterwards the copied application can be re-installed from its new location using the Barista [Install Application Wizard](#).

WARNING

Prior to version 12.10, modifications made using Barista's Default Company (App Company ID = 00000000) and Standard Development (App Product ID = DEV) can NOT be re-installed.

[\(back to contents\)](#)

- **Distribute EM Update Wizard Files**

Description:

This utility moves files from a specified directory to the correct location for the specified database.

Details:

The Update Wizard in BASIS's Enterprise Manager (EM) currently copies/converts all files in a given database to a single destination location. For an Addon upgrade, the destination directory will therefore contain not only all of the Addon database files, but also Barista files (administrative and dictionary tables), as well as any files belonging to vertical or customization projects.

Launch the Distribute EM Update Wizard Files utility from the new instance of Barista Addon. The utility removes the redundant Barista files (the new copy of Barista already contains the files it needs), and distributes files for other projects to the corresponding data directories.

[\(back to contents\)](#)

File Utility

- **Copy Directory Recursively**

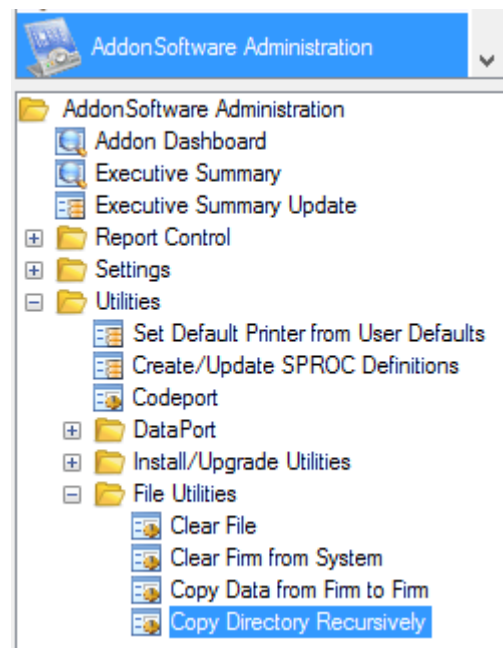
Description:

This utility copies a directory and all of its subdirectories from one location to another.

Details:

The Copy Directory Recursively utility can be used in place of OS level commands to recursively copy a directory from one location to another.

[\(back to contents\)](#)



DataPort Utility

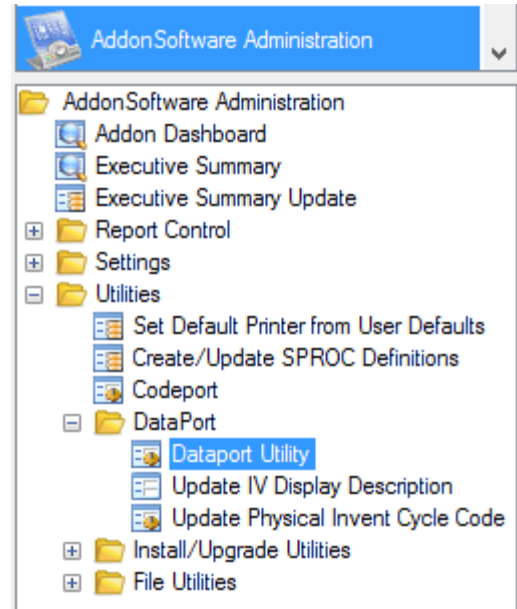
- [DataPort](#)

Description:

DataPort processes all of the legacy Addon files from Addon version 6 or 7 and converts them for use in the new Barista Addon.

Details:

This utility allows for specifying Source and Target directories. In addition, individual files may be selected so that all or only some files in a directory may be included in a data port. This flexibility is especially useful in cases where a few files in the legacy data may require cleanup and therefore multiple rounds of porting.



Note: DataPort requires that the legacy version of Addon have an up-to-date Addon dictionary.

Two important functions of DataPort are converting all packed dates into CCYYMMDD format, and normalizing the legacy parameter and code files.

Note: Without modification, DataPort does NOT convert data from non-standard Addon or from verticals.

[\(back to contents\)](#)

APPENDIX E - Using Git in the Upgrade Process

BASIS has a Git archive you can use to aid in the upgrade process. The Git archive serves two basic purposes. First, prior to an upgrade, you should check out the Git archive and roll it back to the version of Addon you're currently running. Copy/paste your Addon files over top of the Git archive, and look for differences. Analyze differences to make sure they are properly incorporated into the new Addon. Second, if you modified Addon forms/callpoints in a custom modification <cust_mod> project, you'll want to use the Git archive in combination with some scripts (also available in the archive) to help you upgrade your custom project.

As mentioned in [this note](#) in the upgrade instructions, to use Git in the upgrade process you should copy the <cust_mod> project without installing it. This allows you to use the Git archiving process to update the modified callpoint code with changes in the new version of the standard Addon callpoint code *before* the modification project is installed (see [Using the BASIS Git Archive](#) for information about the archive, and [Upgrading Mods with Git](#) for information on using it).

[\(back to contents\)](#)

* * * * *