

Create Vertical and Customize Applications with Barista

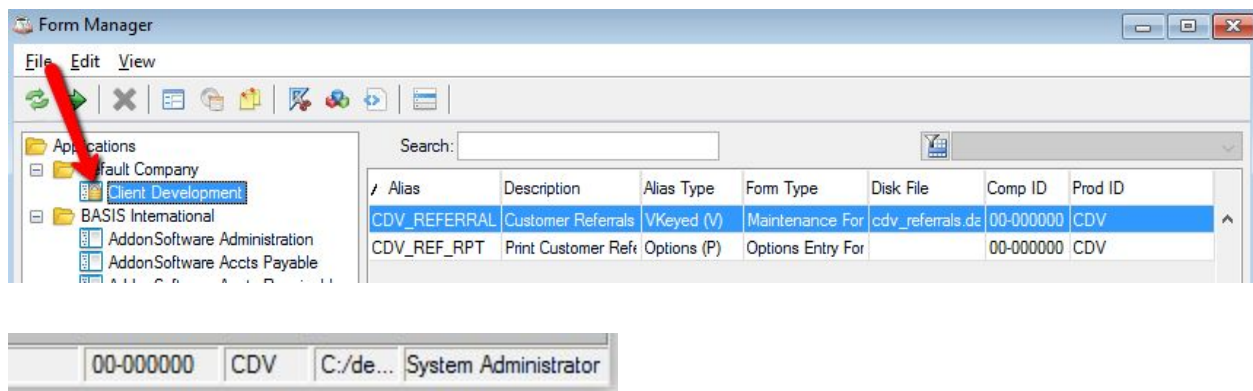
Part 3: Customizing and Installing Applications

Customizing existing applications is similar to vertical development with one critical difference. When developing, you set Barista to work in "Application Replication" mode. In this mode, you make modifications to forms and/or callpoints that already exist in the "base" application, and Barista saves the modifications to the devproject directories, leaving the base application unaltered.

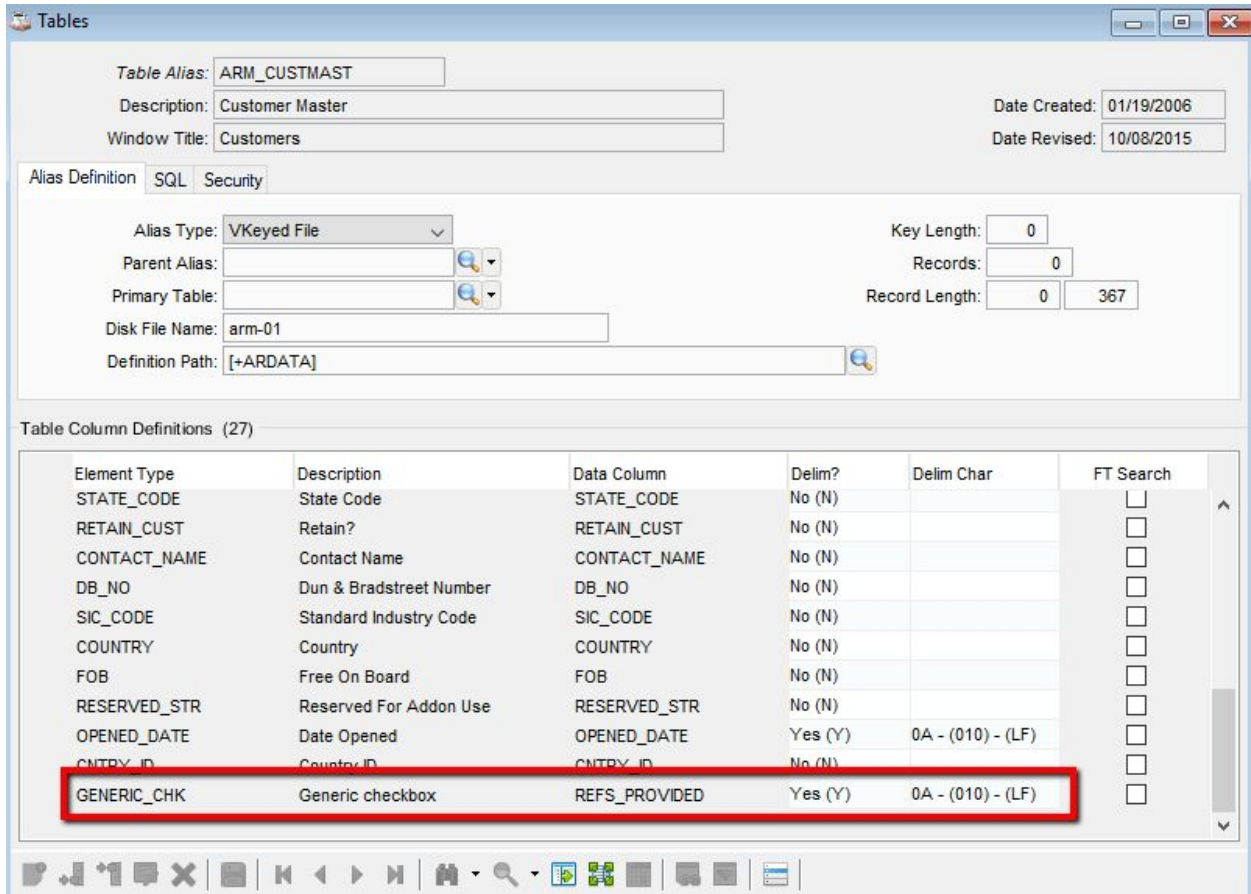
In this section, we will alter the standard AddonSoftware Customer Master form to include a checkbox that interfaces with our customer referrals table.

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

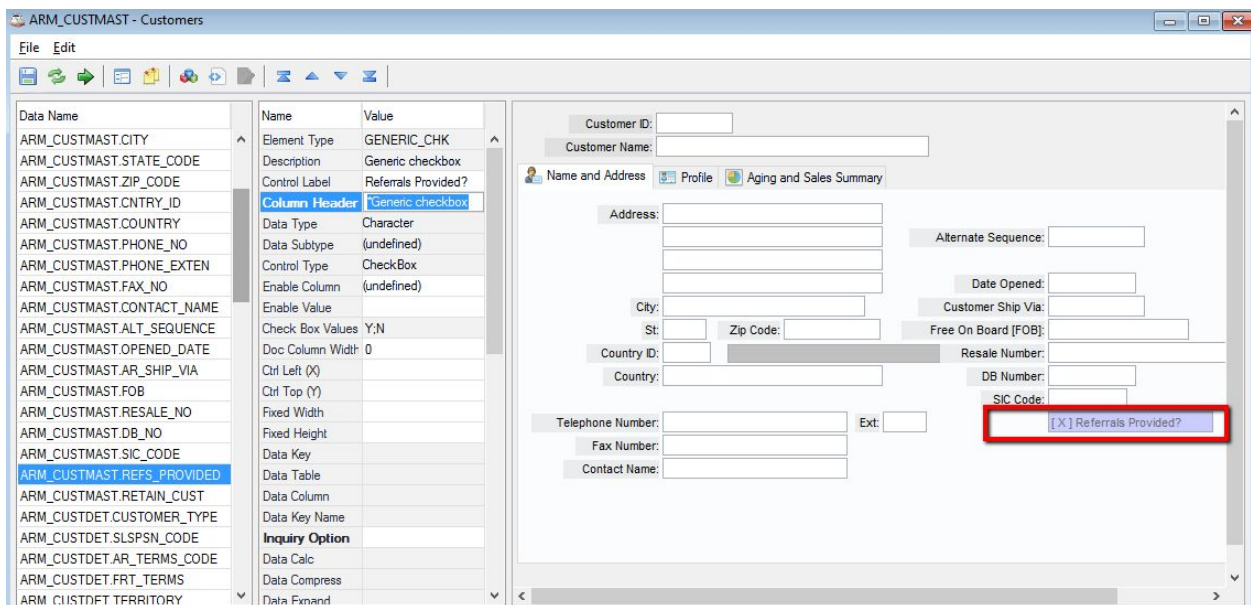
Begin by calling up the Form Manager. Right-click the Client Modifications application and notice the padlock that appears and the company/product indicators in the status bar at the bottom of the MDI. When in this mode, all changes you make in other application modules are saved into the devproject data directories, rather than in the directories for the base project.



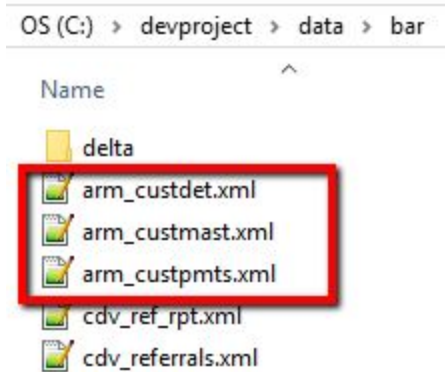
Now select the AddonSoftware Accts Receivable node from the tree. Add a new checkbox element to the table. In this example, the GENERIC_CHK Element Type is used with a Data Element name REFS_PROVIDED. Save, and then select the Create/Update option from the Options menu to rebuild the table.



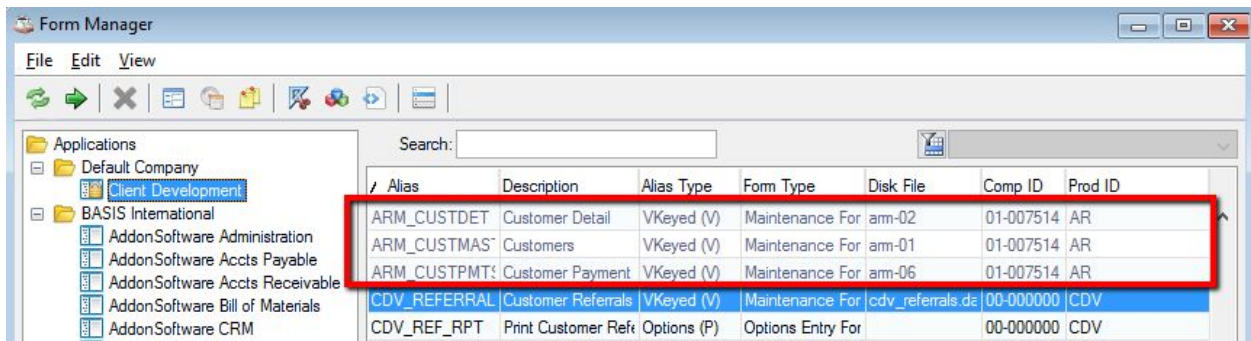
Now use the Form Designer to place the new field on the Name and Address tab, and build the form.




When Barista builds the form in replication mode, it writes arm_custmast.xml (along with auxiliary forms arm_custpmts.xml and arm_custdet.xml) to our devproject/data/bar/ directory. This replication is what makes it possible to install or update standard Addon and then re-install our custom version of the form.




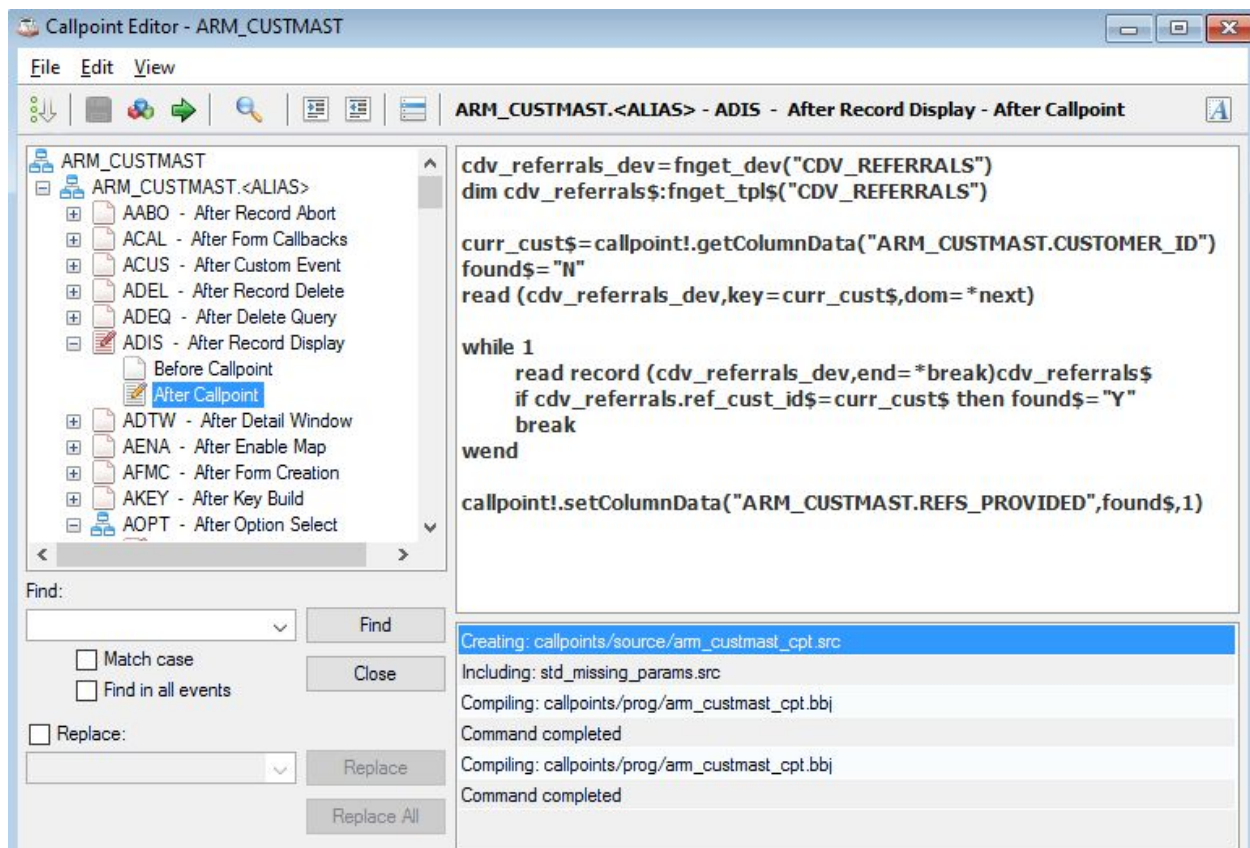
Barista also creates a cross reference in your project of tables/forms from Addon that you've customized. In the Form Manager, you'll now see entries for Addon's Customer Master forms in addition to the new forms you created. Note that the font color is gray, and the product ID's are from Addon, so it's easy to distinguish customizations from new forms.



Next, add callpoint code to the Customer Master form that opens and reads our Referrals file and sets the checkbox accordingly. Notice that the Callpoint Editor behaves differently when in replication mode. The standard Addon callpoint code is available to view or copy/paste into your project callpoints, but it cannot be modified. Instead, you create callpoints to run before, after, or instead of the standard callpoint code. Running your code instead of the standard code is typically done by copy/pasting the standard code into your "before" callpoint, making whatever modifications you need, and then using the callpoint!.setStatus("SKIP") method to instruct Barista to skip any further callpoint processing for that callpoint.

 **Important note:** When coding callpoints or back-end programs where alternate keys are used, use the `knum=<"KEY_NAME">` syntax rather than `knum=<int>`. Using named keys ensures that keys added during custom development can successfully merge with keys that may be part of a new version of Addon.

 **Important note:** If any of your custom callpoint code uses subroutines or custom objects, make sure you also copy over the "use" statements and/or the entire `<CUSTOM>` callpoint from Addon, so you'll have the necessary subroutines in your callpoint program.



This "after" callpoint .cdf file is saved with our devproject application, and the complete tokenized and source code for the callpoint is compiled/saved in the `c:/devproject/callpoints/` directories. When we run this form, Barista first runs the standard Addon ADIS callpoint, and then runs our customized "after" callpoint.

Customer ID: 00-0100
 Customer Name: Everest Industries

Name and Address | Profile | Aging and Sales Summary

Address: 123 Main St Ste 111
 City: San Bernardino
 St: CA Zip Code: 93121
 Country ID: Country:

Telephone Number: 714.832.8113 Ext: 0234
 Fax Number: 714.931.9220
 Contact Name: Mr. James Rockville

Alternate Sequence: EVEREST
 Date Opened: 02/15/2015
 Customer Ship Via: UPS GCD
 Free On Board [FOB]:
 Resale Number: SR-EW399291
 DB Number:
 SIC Code:
 Referrals Provided?

Quotations Orders Invoices

Let's take another look at the contents of the directories in our devproject. Note that it now includes those components of standard Addon that we've customized:

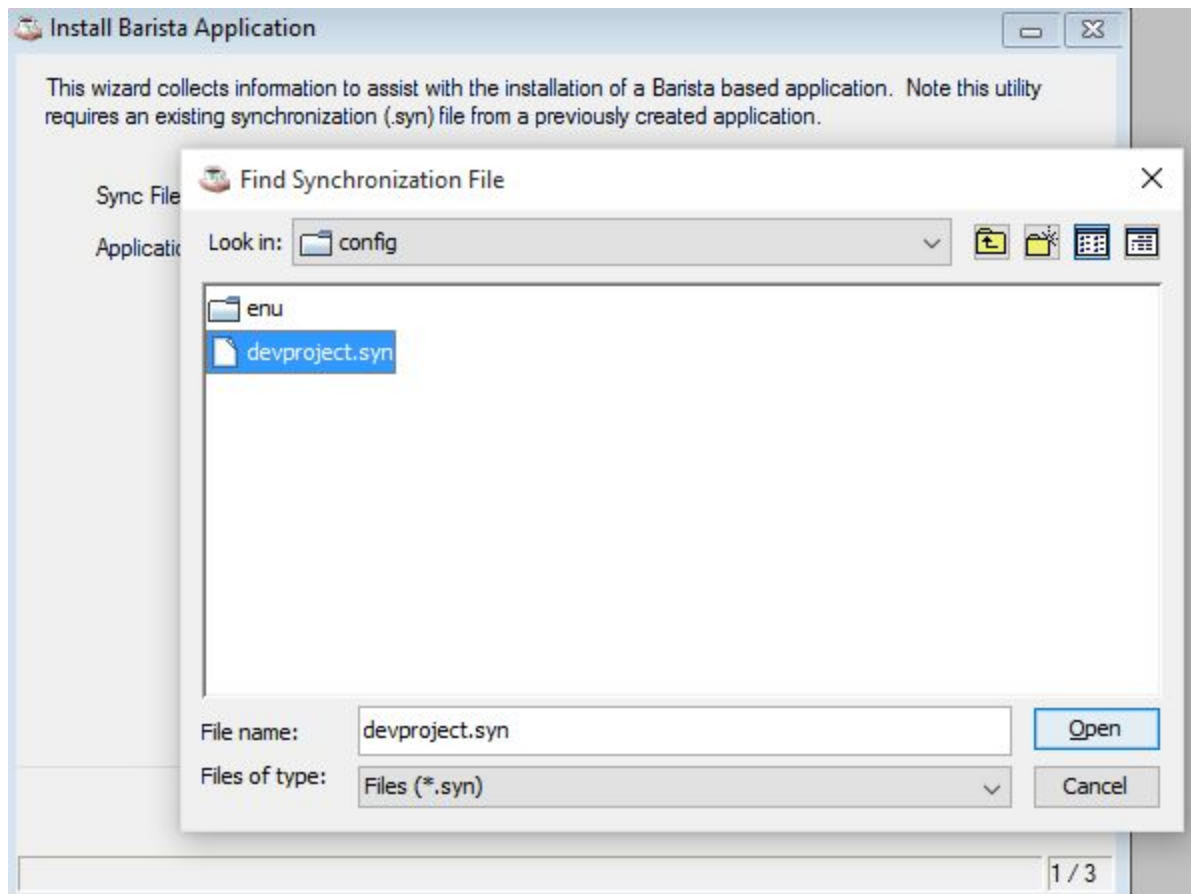
C:/devproject/	File
callpoints/prog/	arm_custmast_cpt.bbj cdv_referrals_cpt.bbj
callpoints/source/	arm_custmast_cpt.src cdv_referrals_cpt.src
config/	devproject.syn
config/enu/	devproject.men
data/	cdv_referrals.dat
data/arc/enu/	cdv_referrals.arc cdv_ref_rpt.arac
data/bar/	cdv_referrals.xml cdv_ref_rpt.xml arm_custmast.xml arm_custdet.xml arm_custpmts.xml
data/bar/delta/	arm_custmast.xml arm_custdet.xml arm_custpmts.xml
data/cdf	cdv_referrals.cdf arm_custmast.cdf
data/def/enu/	cdv_referrals.def cdv_ref_rpt.def
data/sync/	ads_messages~cdv_inval_dt.xml

lib/	<empty>
prog/	cdv_ref_rpt.src
prop/	<empty>

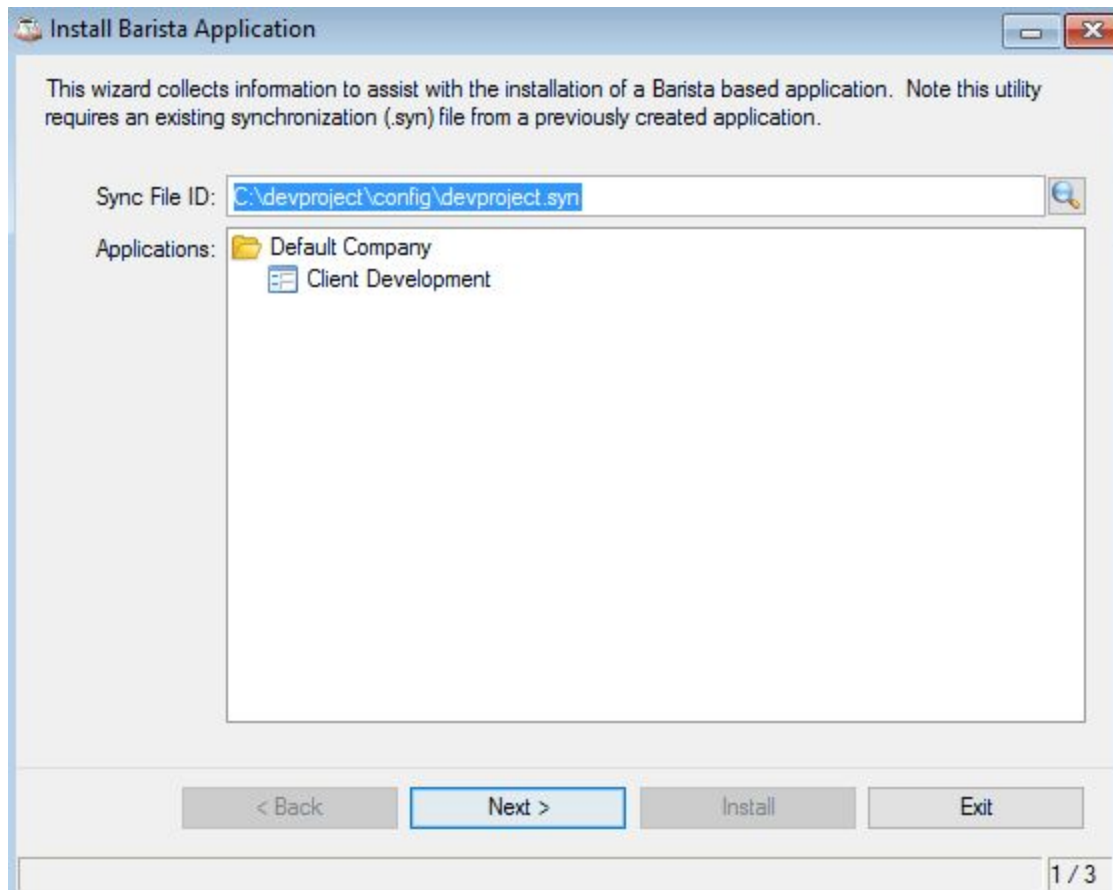
You can also customize code that isn't related to forms, i.e., reports, updates, publics, etc. Begin by copy/pasting the program you want to customize into the devproject/prog folder, then make any desired changes. When you created the devproject, Barista will have placed the devproject/prog path at the front of the PREFIX list, so any programs in the devproject/prog directory will be located/executed rather than their counterparts in the base application directories.

Installing Applications

We now have our own Client Referrals application that interfaces with a customized version of Addon. The Barista Install Application wizard (found in the Barista Utilities menu) facilitates installing our application at another site. Begin by copying the entire devproject file structure to the install location. Note that if the install path is something other than c:/devproject/, you will need to modify pathnames in the .syn file and any sync/adm_modules*.xml files accordingly. The wizard prompts for the location of our .syn file; use the browse button to locate it:

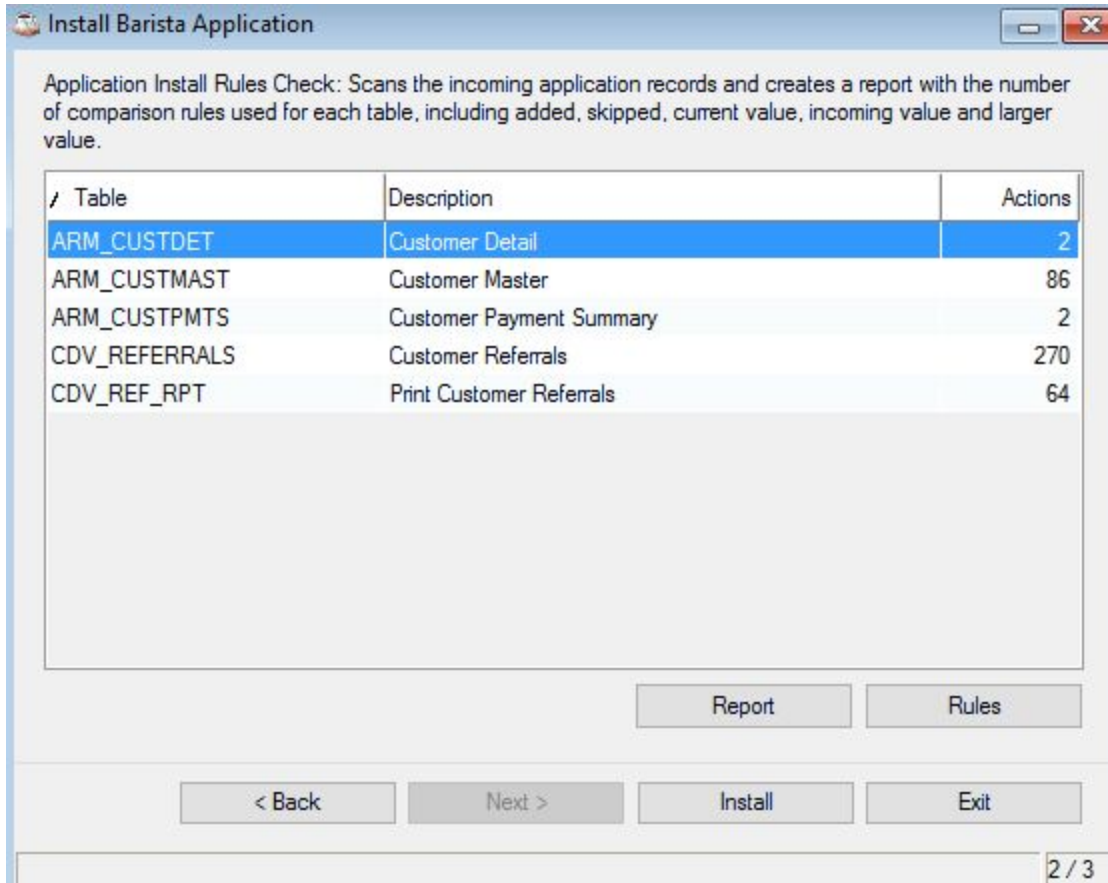


Once selected, Barista uses the contents of the .syn file to provide information to the wizard.



Barista analyzes the incoming project files and creates a report you can review that will show the dictionary actions that will take place. The `install_rules.txt` file determines whether, in the event of a record that's already in the dictionary, the incoming project or the current dictionary value should prevail. The number of dictionary writes for a new form, such as `cdv_referrals`, will typically be much higher than for a modified form such as `arm_custmast`, because Barista keeps track of what was changed in a "delta" .xml file.

In our example, the `devproject` will contain a `data/bar/arm_custmast.xml` file reflecting the entire customer form, including the new checkbox, as well as a `data/bar/delta/arm_custmast.xml` that's much smaller because it only contains what was changed in order to add and position the checkbox and add the callpoint code.



The installation process first updates the Barista dictionary, then exports table definitions to the BASIS dictionary for use by the SQL engine, and finally builds the forms and callpoints.



An important function of the Install Application Wizard is that it also has the intelligence to rebuild the physical Customer Master table to include the new checkbox field.

Because the `cdv_referrals.dat` data file exists in the `devproject/data/` directory, it is now part of the installed application. You may want to clear any test data out of the file, or erase and re-create it.

The final result is a customized version of AddonSoftware that interfaces to a new vertical application. If the user upgrades to a new version of Barista or Addon, or if/as you make additional forms, tables and reports in the vertical application, you can re-install the devproject application.

See also:

[Part 1: Create and Synchronize Barista Applications](#)

[Part 2: Creating Vertical Applications](#)