



Barista Application Development

Part 2: Developing Barista Applications

Once you have installed Barista and created an application/project file structure, you can begin your development. All of the data associated with your project is saved or will be backed up into the various project directories, providing both portability and security.

Specifically, portability means you can install applications in numerous locations by copy/pasting the directory structure (c:/devproject, in our example) to a given site and then run the Barista Install Application Wizard to install/synchronize the application into Barista.

Because project data resides in its own directory structure, it is also "safe" in the sense that it will not be lost or overlaid if you elect to re-install Barista.

The steps below illustrate vertical development that augments and interfaces with AddonSoftware. We will create a Customer Referrals table and report, using the devproject created earlier, with an emphasis on where Barista saves the various files within the devproject directory structure. These examples assume a working knowledge of Barista development. Visit

<http://www.basis.com/products/devtools/barista/documentation/> for additional Barista documentation.

We begin in the Barista Development menu, in the Tables form, to create the definition for a Customer Referrals table. Once you have defined the basic table information, entered the columns for the table in the detail grid, created one or more keys, and created the physical table, you can view, edit, and run the associated form. We will use the DEVPROJECT stbl and Client Development (CDV) module specified for the

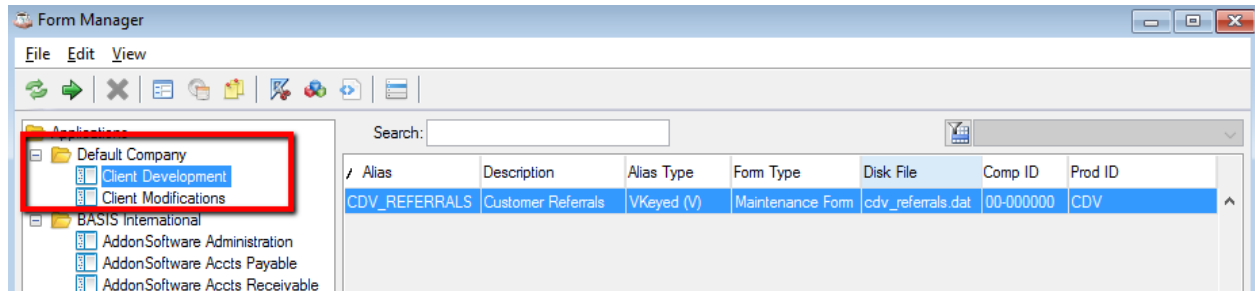
devproject application (Note: Barista uses "[+DEFAULT]" for the default Definition Path, and "DEV" for the default Product ID. *Be sure to change these to valid values for your project.*)

The screenshot shows the 'Tables' application window with the 'Alias Definition' tab selected. The 'Table Alias' is 'CDV_REFERRALS', 'Description' is 'Customer Referrals', and 'Window Title' is 'Customer Referrals'. The 'Date Created' and 'Date Revised' are both '10/08/2015'. The 'Alias Type' is 'VKeyed File'. The 'Parent Alias' and 'Primary Table' are empty. The 'Disk File Name' is 'cdv_referrals.dat'. The 'Definition Path' is '[DEVPROJECT]', which is highlighted with a red box. The 'Key Length' is '0', 'Records' is '0', and 'Record Length' is '0' and '175'. Below the form is a table of 'Table Column Definitions (9)'.

Element Type	Description	Data Column	Delim?	Delim Char	FT Search
CUSTOMER_ID	Customer ID	REF_CUST_ID	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
REFERRAL_ID	Referral ID	REFERRAL_ID	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
TEXT_30	Text entry	REFERRED_BY	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
TEXT_30	Text entry	COMPANY_NAME	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
TEXT_30	Text entry	CONTACT_NAME	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
PHONE_NO	Telephone Number	PHONE_NO	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
CONTACT_DATE	Contact Date	CONTACT_DATE	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
TEXT_30	Text entry	CONTACTED_BY	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>
CUSTOMER_ID	Customer ID	NEW_CUST_ID	Yes (Y)	0A - (010) - (LF)	<input type="checkbox"/>

The screenshot shows the 'Tables' application window with the 'Security' tab selected. The 'Table Alias' is 'CDV_REFERRALS', 'Description' is 'Customer Referrals', and 'Window Title' is 'Customer Referrals'. The 'Date Created' and 'Date Revised' are both '10/08/2015'. The 'App Company ID' is '00-000000' and the 'App Product ID' is 'CDV', both highlighted with a red box. The 'Version' is '1.0'. The 'Default Company' and 'Client Development' links are visible. The 'Export to inquiry system', 'Restrict table record auto add', and 'Restrict table auto create' checkboxes are checked. The 'Restrict table rebuild' checkbox is unchecked.

The Form Manager contains tree nodes for both of the modules defined with the devproject application. Selecting the Client Development module shows the new table definition. With some minor cosmetic work on the form, it is ready to run:



Customer Referrals

Customer ID: 00-0100 Everest Industries

Referral ID: 000001

Referred by: Mr. James Rockville

Company: McKinley Industries

Contact name: Sarah Jackson

Phone: 503.555.1234

Contact Date:

Contacted by:

New Cust ID: -

We can add our new form to the Barista menu as well, again making sure to use the Company ID and Product ID from devproject.

Menus

Client Development (DEVPROJECT) ▾

- C:/devproject/config/enu/devproject.men
- Customer Referrals (CUST_REF)
 - Customer Referrals (CDV_REFERRALS)

Option Information

Option ID:

Option Type:

Aliases:

Aliases	Description
BMU_IVCOSTING	BOM Inventory Costing Update
BMU_PRGOBSEDET	Purge Obsolete Lines
CDV_REFERRALS	Customer Referrals
COUNTER_PRINTERS	Select Valid Counter Printers
CRM_ACTIVITIES	CRM Activities
CRM_CONTACTS	CRM Contacts

Group Control

Target:

Arguments:

Config File Path:

Working Directory:

Description:

Image ID:

☐ Automatically expand menu ☐ Don't show in menu

App Company ID: Product ID: Version:

Help ID:

☐ Allow multiple running copies of this process

☐ Don't allow running in browser

CSS File Path:

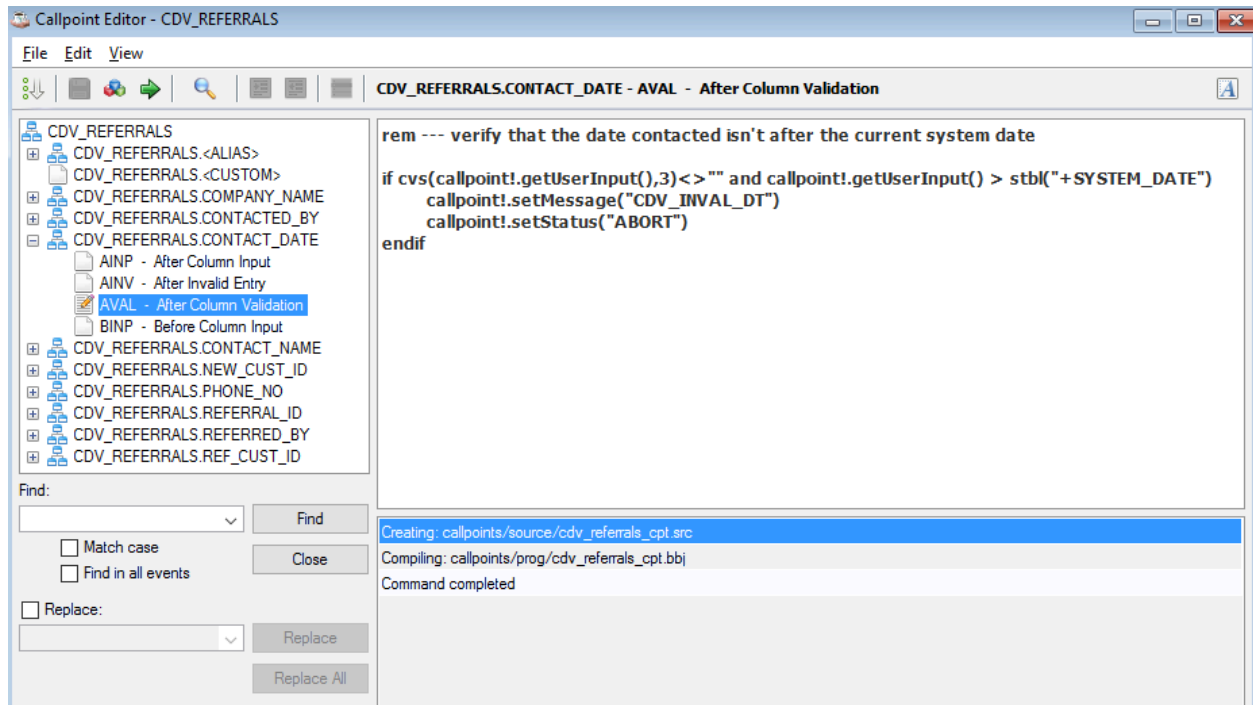
Menu Options


- Sales Order Processing
- Shop Floor Control
- AddonSoftware Administration
- Customer Referrals
- Barista Administration
- Barista Utilities
- Barista Development

Customer Referrals

- Customer Referrals

Next, we'll enhance the form by adding code to the After Validation (AVAL) callpoint for the contacted date. With this code, we'll retrieve the contacted date just entered, and verify that it is not greater than the current system date. If it is, we will display a system message and abort the entry:



 **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.

We create the message via the Barista Administration, System Messages form:

System Messages

Message ID:

Message Title:

Format

Message Type:

Message Icon:

Default Button:

Options

Display Options:

Message Return Codes:

System

App Company ID: [Default Company](#)

App Product ID: [Client Development](#)

Message

Message Text:

Our form now warns the user about invalid dates:

Customer Referrals

Customer ID: [Everest Industries](#)

Referral ID:

Referred by:

Company:

Contact name:

Phone:

Contact Date:

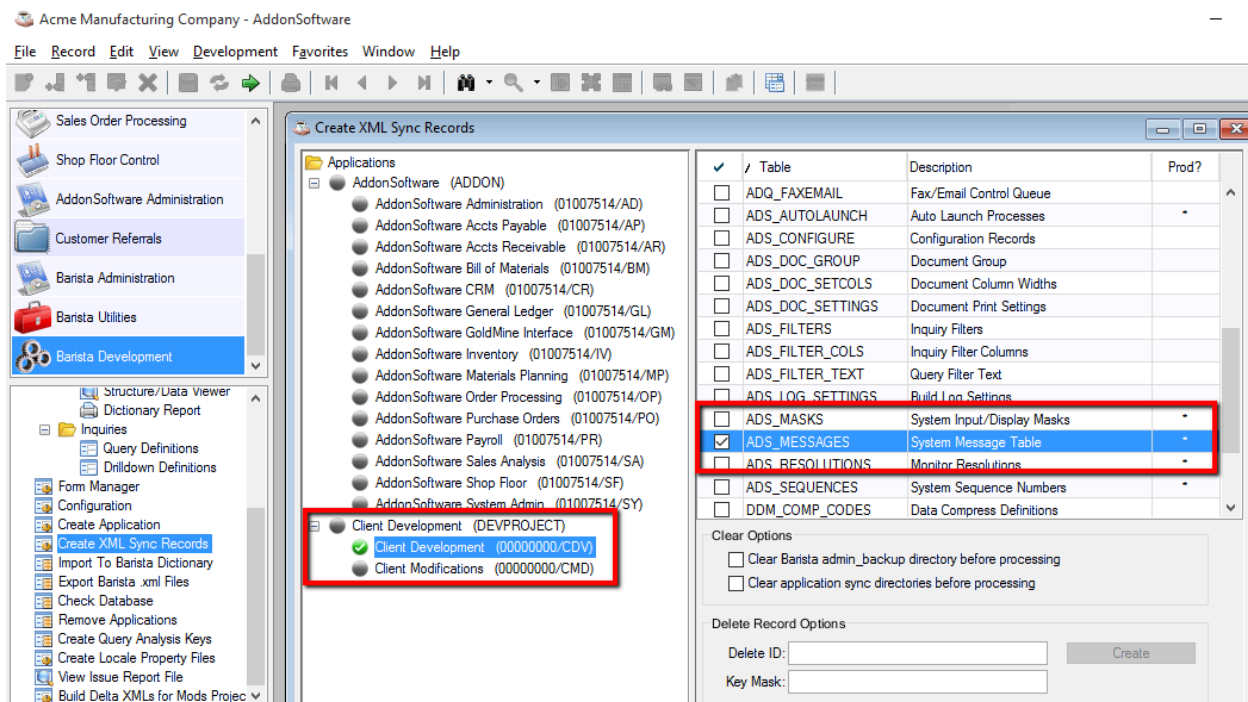
Contacted by:

New Cust ID:

Client Referrals

Date contacted must be on or before current system date.

The callpoint definition file is saved in the data/cdf/ directory of our devproject structure, but the system message was added to a Barista Administrative table. In order to ensure that the message is included with the application when porting it elsewhere or syncing back into a new Barista install, we need to run the Create XML Sync Records program found in the Barista Development menu. This utility creates xml versions of the selected Barista Administrative records. Any record that contains a Company and Product ID will generate an xml in the corresponding application's data/sync directory. All non-application-specific records will be backed up in Barista's admin_backup directory. Copy these as needed to your application's data/sync directory.



The message record we created for our form contains our CDV Product ID, so the backup saves this xml in devproject/data/sync/. Now, when installing at a new site, or re-syncing after a fresh Barista install, Barista will process the contents of the devproject/data/sync/ directory, and write a record to the System Message table based on this xml.

```
<?xml version='1.0' encoding='UTF-8' ?>
<meta name='barista_version' content='15.0' />
<barista_admin_record>
  <barista_record_table>ADS_MESSAGES</barista_record_table>
  <msg_id>CDV_INVALID_DT</msg_id>
  <msg_title>Client Referrals</msg_title>
  <msg_type>0</msg_type>
  <msg_icon>48</msg_icon>
  <msg_default_but>000</msg_default_but>
  <asc_comp_id>00000000</asc_comp_id>
```

```

        <asc_prod_id>CDV</asc_prod_id>
        <msg_text>Date contacted must be on or before current system date.[$0a$]
    </msg_text>
</barista_admin_record>

```

The last step in our vertical development is to create a report that lists referrals, using an Option Entry form to specify the desired customer ID range for the list. We will use the document_base.src file that comes with Barista (located in <barista install>/sys/lib/) and modify it for our purposes. We need to copy/paste the file into devproject/prog/, and give it the same name as the "Run program" used in the Option Entry form.

temp_1_0001010456

Acme Manufacturing Company
Print Customer Referrals
From: 000100
To: 000200

Customer ID	Name	Referral Company	Referral Contact	Phone
000100	Everest Industries	McKinley Industries	Sarah Jackson	503.555.1234
000100	Everest Industries	St. Helens Contractors	Karen Clymer	503.555.2333
000200	Western Sport Distributors	Allied Soccer Supply	Jim Nayseum	503.555.0987

3 record(s) listed

After this initial development, the files in the devproject application are distributed as follows:

<u>C:/devproject/</u>	<u>File</u>
callpoints/prog	cdv_referrals_cpt.bbj
callpoints/source	cdv_referrals_cpt.src
config/	devproject.syn
config/enu/	devproject.men
data/	cdv_referrals.dat
data/arc/enu/	cdv_referrals.arc
	cdv_ref_rpt.arc
data/bar/	cdv_referrals.xml

	cdv_ref_rpt.xml
data/cdf	cdv_referrals.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>

This directory structure contains everything we need to install the devproject application at a different site, or to re-sync it in our own development environment after a fresh Barista install.

Back to:

[Part 1: Create Barista Applications](#)