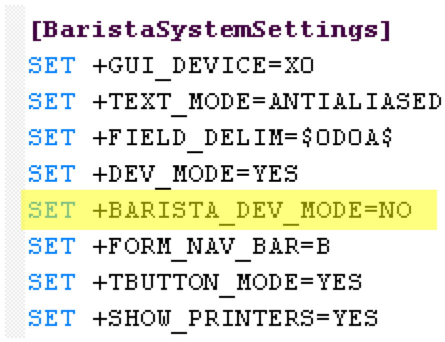
How to: Create a Barista System Message for Addon

AddonSoftware® makes use of the Barista System Messages table to store various messages used throughout the application. This document describes the process of creating a new System Message record, creating an .xml version of the record, and committing that .xml to the SVN repository for inclusion in the standard product.

Note: While this document addresses System Messages, the same process (creating an .xml version of the record and committing to the repository) applies to the other Barista Administrative tables (Input/Display Masks, Sequence Numbers, etc.).

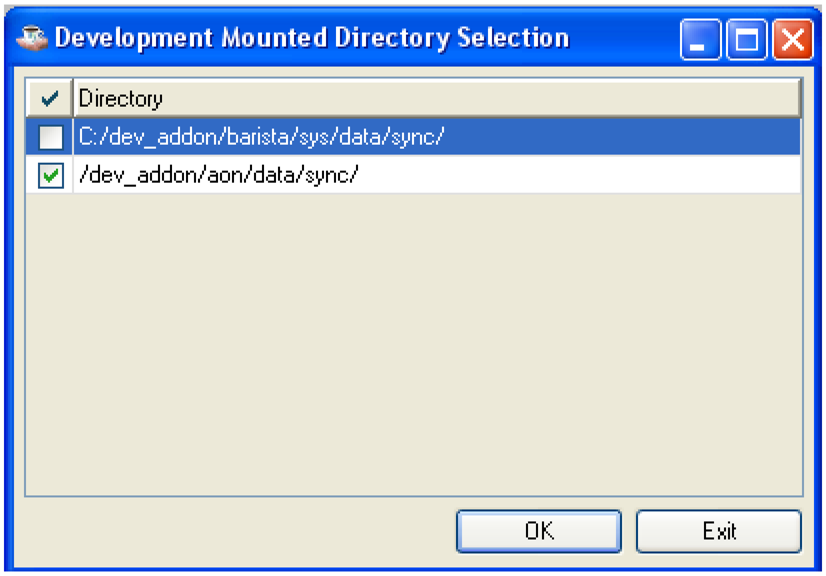
<u>Development Environment Global</u>
Before adding or changing administrative records that are part of the standard Addon product, you need to set a global that will cause Barista to make an .xml version of each saved record. You can either alter the +BARISTA_DEV_MODE global directly in your barista.cfg, or add it to your addon.syn file so that it will be "pushed" into your barista.cfg with each synchronization pass. The default for this global is "YES." In this mode, you can add and/or alter Barista Administrative records without being prompted for an .xml save location. Set the value of the stbl to "NO" if you want Barista to prompt you for a save location for any administrative records.



```
[BaristaSystemSettings]
SET +GUI_DEVICE=XO
SET +TEXT_MODE=ANTIALIASED=OFF
SET +FIELD_DELIM=$0D0A$
SET +DEV_MODE=YES
SET +BARISTA_DEV_MODE=NO
SET +FORM_NAV_BAR=B
SET +TBUTTON_MODE=YES
SET +SHOW_PRINTERS=YES
```

Figure 1: Setting +BARISTA_DEV_MODE

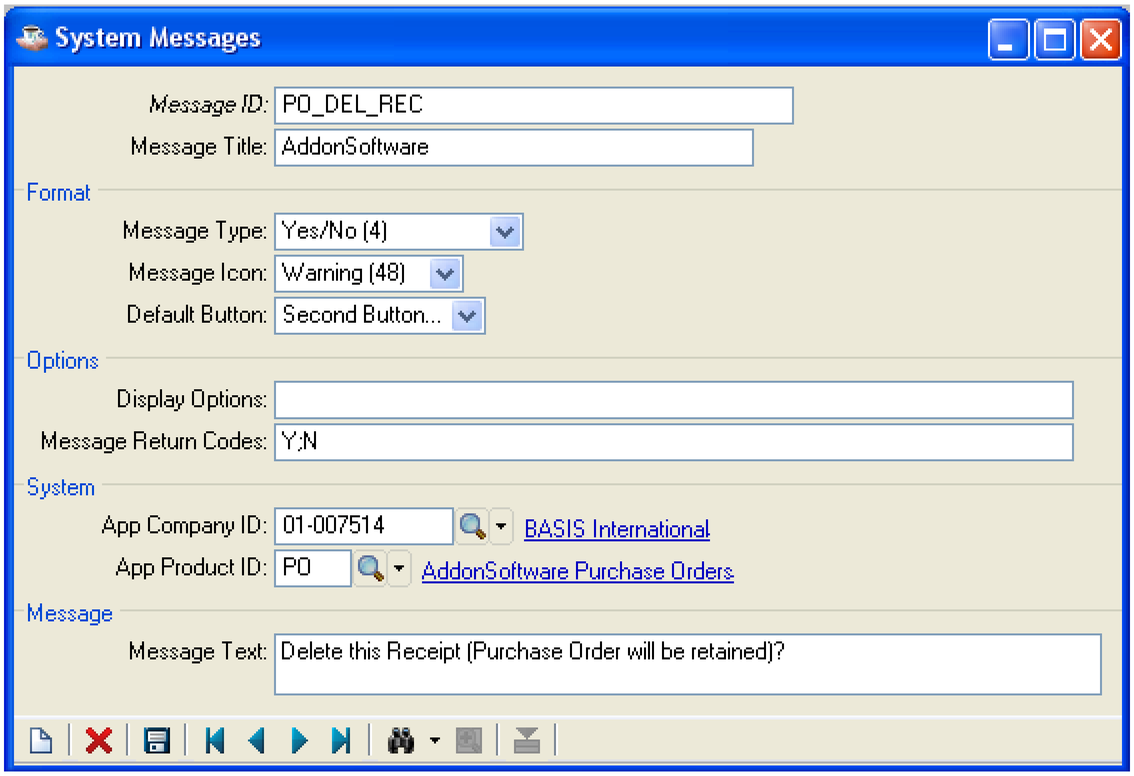
<u>Launching System Messages</u>
Given the above setting, when you launch a Barista Administrative form, you'll be presented first with a grid showing the currently mounted directories. Check only the directory pathname corresponding to your Addon SVN workarea:



✓	Directory
☐	C:/dev_addon/barista/sys/data/sync/
✓	/dev_addon/aon/data/sync/

Figure 2: Mounted Directory Selection

Once you click OK, you'll be taken into the desired form -- in this case, System Messages. In this example, we'll add a basic warning message to which the user can reply Yes or No. The application code invokes the message by calling a Barista public, then takes action based on the return code.



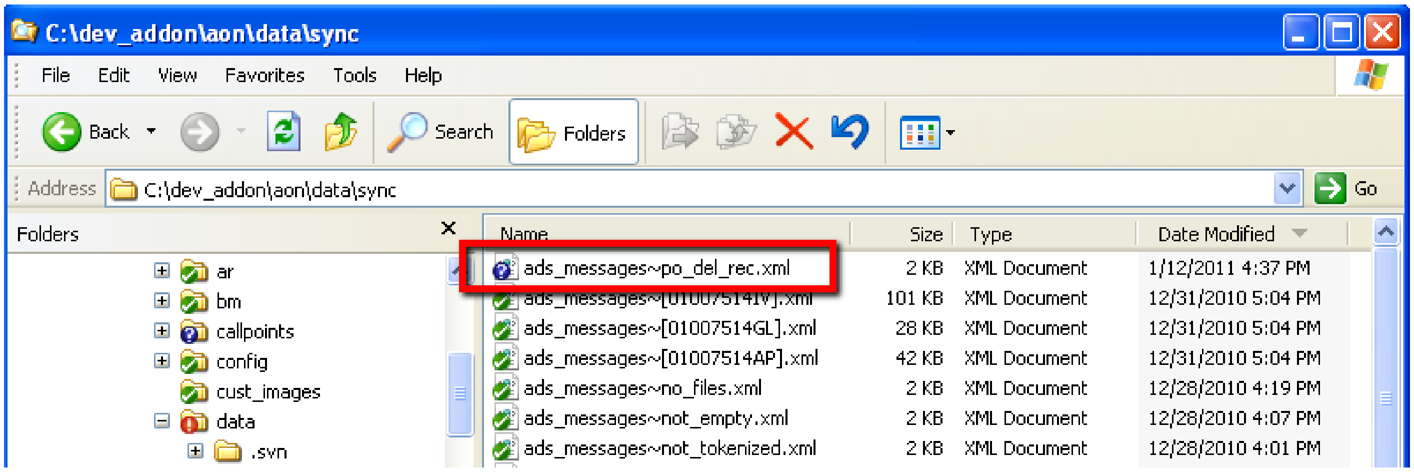
Message ID:	PO_DEL_REC
Message Title:	AddonSoftware
Message Type:	Yes/No (4)
Message Icon:	Warning (48)
Default Button:	Second Button...
Display Options:	
Message Return Codes:	Y;N
App Company ID:	01-007514 BASIS International
App Product ID:	PO AddonSoftware Purchase Orders
Message Text:	Delete this Receipt (Purchase Order will be retained)?

Figure 3: Barista System Messages form

- Message ID: Create an intuitive message ID. This ID is passed to the Barista messaging public.
- Message Title: Use AddonSoftware.
- Message Type:
    - OK
    - OK/Cancel
    - Abort/Retry/Ignore
    - Yes/No/Cancel
    - Yes/No
    - Retry/Cancel
    - Custom: Allows you to supply your own button text
- Message Icon: Select the appropriate icon for the message.
- Default Button: Defines which button should be highlighted/selected as the default when Barista displays the message.
- Display Options: For use with the Custom Message Type; supply the desired button text for each button, separated by semicolons.
- Message Return Codes: Supply the return codes corresponding to the buttons on the form.
- App Company and Product ID: Select the BASIS Company ID and appropriate Product ID. If the message will be used in multiple modules, use “AD.”
- Message Text: Supply the text to appear in the message body. Note that tokens such as %1 are supported. The text for the token is supplied by the application. If using tokens, the application program should use Translate!.getTranslation(“<KEY>”,”<value>”) syntax rather than hard-coding the token literals. This allows the entire message -- including the tokens -- to be internationialized.

<u>Add/Commit the .xml to the repository</u>

When you save the message, Barista will not only write it to the System Messages table, but will also (because of the +BARISTA_DEV_MODE setting) create an .xml version of the record in the <workarea>/aon/data/sync/ directory (Figure 4). You must add and commit this file to the SVN repository so that the new message is part of the standard product, and is sync’d into Barista when Addon is built.



Name	Size	Type	Date Modified
ads_messages~po_del_rec.xml	2 KB	XML Document	1/12/2011 4:37 PM
ads_messages~[01007514IV].xml	101 KB	XML Document	12/31/2010 5:04 PM
ads_messages~[01007514GL].xml	28 KB	XML Document	12/31/2010 5:04 PM
ads_messages~[01007514AP].xml	42 KB	XML Document	12/31/2010 5:04 PM
ads_messages~no_files.xml	2 KB	XML Document	12/28/2010 4:19 PM
ads_messages~not_empty.xml	2 KB	XML Document	12/28/2010 4:07 PM
ads_messages~not_tokenized.xml	2 KB	XML Document	12/28/2010 4:01 PM

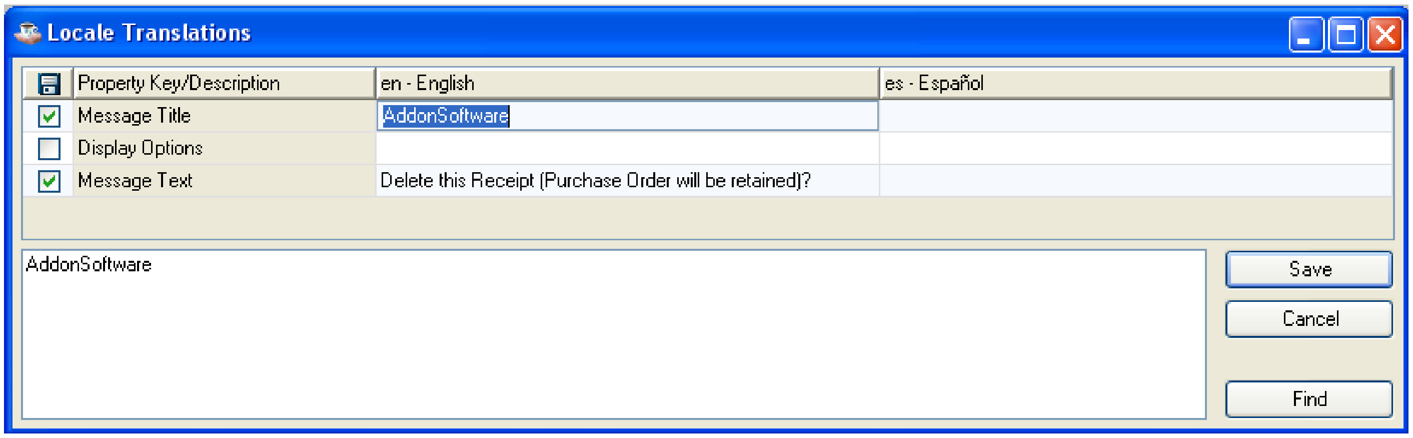
Figure 4: xml version of new System Message record in data/sync/ directory

<u>Update the properties files</u>

Since Barista/Addon is designed to run in different languages, you'll also want to make sure to update the properties files for the new System Message so it can be translated. From the System Messages form, call up the Locale Translations form by pressing F6 or selecting Edit Language Text from the Edit menu.

Copy the values from the base language cells into the corresponding cells for the other language(s). Note: Message Text is a multi-line edit control in the System Message form, so can't be edited directly in the grid cells because of the potential for embedded tabs and linefeeds. To copy the base language Message Text to the other language(s), select the grid cell for the base language, then copy the text from the larger text area below the grid. Select the corresponding cell for the other language(s) and paste into the larger text area.

For more information about the Edit Language Text and BBjabber utilites, see http://links.basis.com/ajtyz.

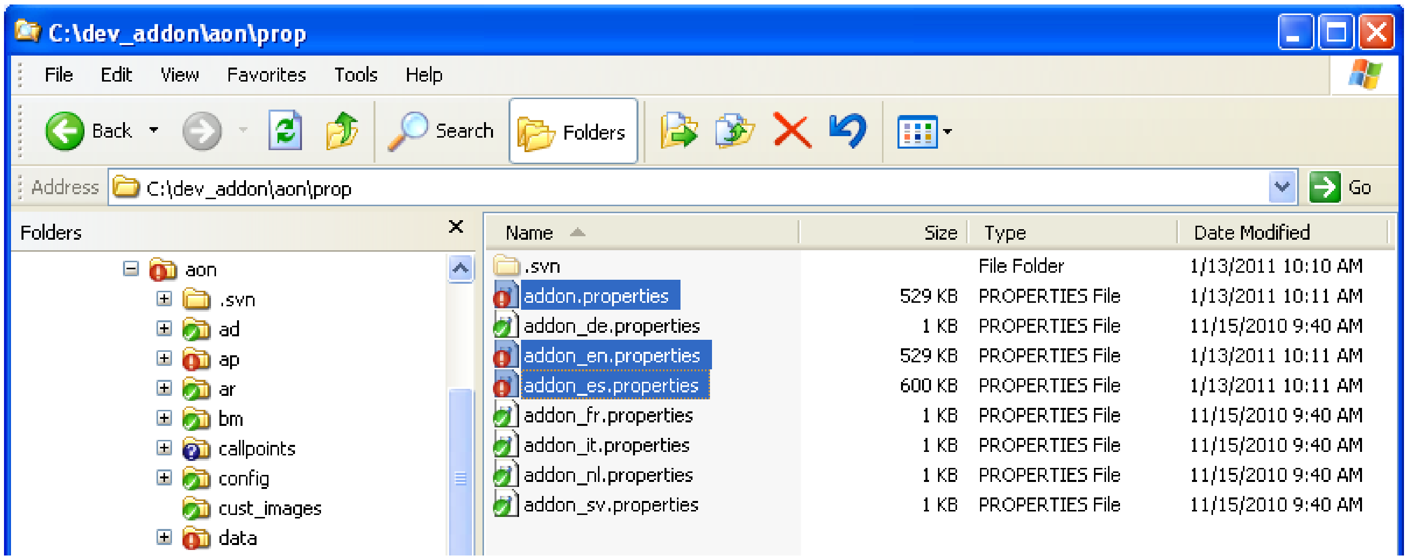


	Property Key/Description	en - English	es - Español
✓	Message Title	AddonSoftware	
	Display Options		
✓	Message Text	Delete this Receipt (Purchase Order will be retained)?	

AddonSoftware

Figure 5: Use the Locale Translations form to update the properties files

Barista updates the *.properties files in the aon/data/prop/ directory. As always, run a diff on the changed files to make sure you only see the expected modifications, and then commit them back to the repository.



aon
- .svn
- ad
- ap
- ar
- bm
- callpoints
- config
- cust_images
- data

Name	Size	Type	Date Modified
.svn		File Folder	1/13/2011 10:10 AM
addon.properties	529 KB	PROPERTIES File	1/13/2011 10:11 AM
addon_de.properties	1 KB	PROPERTIES File	11/15/2010 9:40 AM
addon_en.properties	529 KB	PROPERTIES File	1/13/2011 10:11 AM
addon_es.properties	600 KB	PROPERTIES File	1/13/2011 10:11 AM
addon_fr.properties	1 KB	PROPERTIES File	11/15/2010 9:40 AM
addon_it.properties	1 KB	PROPERTIES File	11/15/2010 9:40 AM
addon_nl.properties	1 KB	PROPERTIES File	11/15/2010 9:40 AM
addon_sv.properties	1 KB	PROPERTIES File	11/15/2010 9:40 AM

Figure 6: Modified properties files