

How to: Use Password-protected Overrides in Barista/Addon

In legacy Addon, users had the ability to override certain prohibited processes by typing “SPECIAL” at appropriate points during input or an update. Barista/Addon has this capability as well, but it is far more secure, user-configurable, and integrates with Barista’s messaging functionality.

Create a Configuration Record

Configuration Value, and Comments aIn Barista Administration => Maintenance => Configuration Records, create a record similar to the one shown in Figure 1. The Configuration ID *must* be “UPDT_OVERRIDE.” The Description, re user-defined. Note that the Configuration Value is stored in encrypted form on disk.

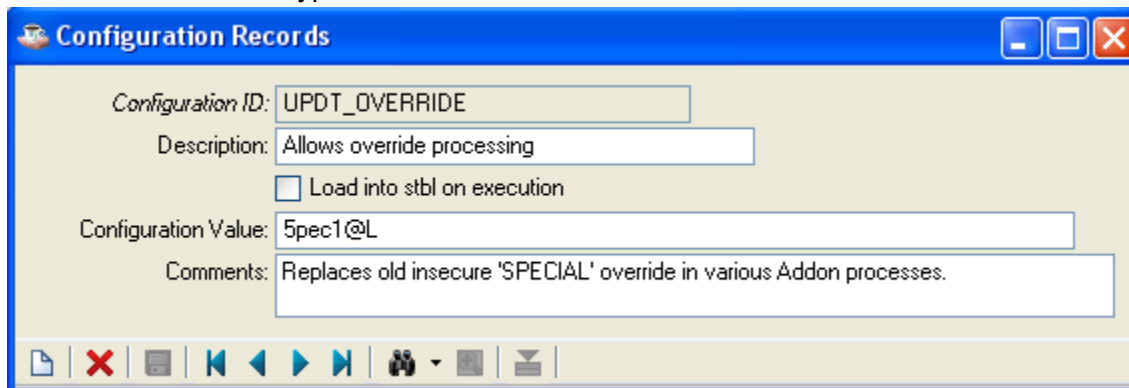


Figure 1: Barista Configuration Record for UPDT_OVERRIDE

Create System Messages

Processes that check for certain error conditions display a Barista System Message if the condition exists. The only button present on such messages is the OK button, so regular users have no obvious way to override these warnings. However, an administrator or other user who has been given the override, that is, the Configuration Value for the UPDT_OVERRIDE record, can override these messages by pressing Alt-<enter>, and then providing the Configuration Value.

Figure 2 shows how to create a Barista System Message that interfaces with the UPDT_OVERRIDE Configuration Record. The key here is to set up a single button message type, but provide two values in the Message Return Codes. The first value is returned when the user dismisses the form in the usual way (i.e., clicking the OK button). The second value *must* be UPDT_OVERRIDE, which enables the override feature in Barista. Separate the values with a semicolon. If the only return code is UPDT_OVERRIDE, it *must* begin with a semicolon.

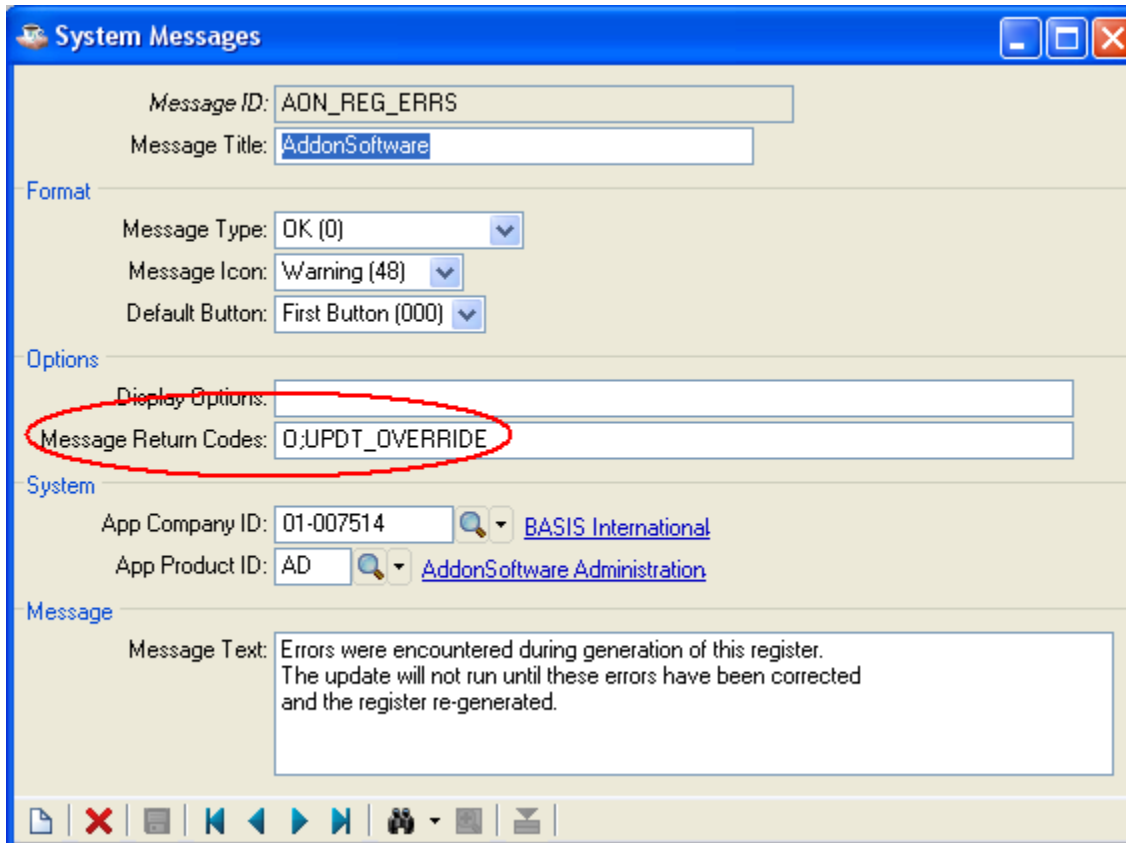


Figure 2: Setting up a Barista System Message to allow override processing.

When presented with such a message, if the user presses Alt-<enter> a set of parentheses appears in the message title bar. The user should enter the override here, then dismiss the message (Figure 3). Barista intercepts the override and validates it. If validation fails, processing continues normally, that is, with no special processing allowed.

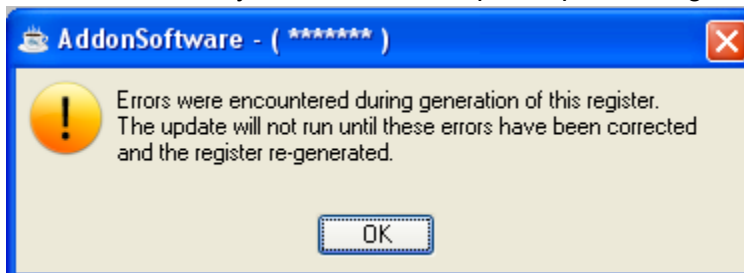


Figure 3: Entering an override on a Barista System Message

Handling Overrides in Application Code

In the application callpoint or backend code, you will need to add a bit of additional logic to analyze the value(s) returned from the message call and proceed accordingly. The two code fragments shown in Figure 4 give an example of how to invoke the message, check to see if override validation passed, and proceed accordingly. In this sample we invoke the message if errors have been encountered during processing. Only if Barista returns "PASSVALID" from the message display will we be able to run the next (update) program.

```

rem --- Destroy the progress meter and run the next overlay

call pgmdir$+"adc_progress.aon","D","","","","",0,0,0,0,status
update$=pgmdir$+"apu_invoiceentry.aon"

if errors
    gosub Display_Message
    if special_flag$<>"Y" then update$=""
endif

```

```

Display_Message: rem --- Display Message Routine

msg_id$="AON_REG_ERRS"
gosub disp_message
if pos("PASSVALID"=msg_opt$)<>0 special_flag$="Y"
return

```

Figure 4: Application code interface with Barista System Message override

By following the above steps, you can allow for exception processing throughout the application. Note that this Barista override is global. In other words, changing the Configuration Value in the UPDT_OVERRIDE Configuration record changes the override for any and all programs.