Flashes Anywhere ™
Version 1.0.6

Workflow Series ™

Installation and User's Guide

8900 Business Park Drive
Austin, Texas 78759

Web: www.fchoice.com

Phone: (512) 418-2905
Fax: (512) 418-2983

Trademarks

First Choice Software, Inc. (FCS) develops add-on software and customizations to the following Clarify products:

- ClearSupport
- ClearHelpDesk
- Policies and Customers
- Product Manager
- ClearLogistics
- ClearQuality
- ClearContracts
- ClearSales
- ClearCallCenter

FCS uses the following Clarify tools:

- User Interface Editor
- Data Dictionary Editor
- Data Exchange
- ClearBasic Exchange
- ClearBasic Batch

Clarify, Inc. and all of its product and tool names are trademarks of Clarify, Inc.

Publication Notice

The information in this document does not constitute a warranty of performance. First Choice Software, Inc. reserves the right to modify the content of this document without obligation to notify any person or party of such revisions or changes. First Choice Software, Inc. assumes no liability for losses incurred as a result of the use of information contained in this document.

Table of Contents

General Overview

Flashes are a very useful feature in Clarify. They allow you to put important reminders on a variety of objects. These reminders can be displayed at key points in the Clarify workflow

Clarify provides flashes for only a small number of objects, however. Since version 5.0 of Clarify there have been flashes on contacts, sites, and contracts. Later, Clarify expanded this to include flashes on leads, accounts (bus_orgs), and opportunities.

Customers have many times said that they would like to include flashes on a variety of other objects such as parts, site_parts, cases, and bugs.

In addition, Clarify customers have been frustrated with the fact that each flash can be related to one and only one object. For example, suppose that you wish to use the same flash text on four different sites, and one account. Base Clarify would require you to create five different flashes, even though they basically are the same. And, in the future, if you wish to modify the text, or delete the flash, you must modify all five of them.

Finally,

This document describes **Flashes Anywhere** (FA), a Clarify add-on program that allows an administrator to easily and quickly set up what types of objects have flashes associated with them, the sort fields and output fields for each. FA allows the user to associate a single flash with as many objects as desired. The example given above is a snap with FA.

With **Flashes Anywhere**, you may create flashes on any table or view in the Clarify system – including user-defined ones.

Flashes Anywhere also allows the user to select flashes, and easily maintain your flashes.

This product is a Clarify-based application that is internationalized and localized. The application ships with a set of default strings in US English. This document describes how to easily modify the database to allow users to operate the product in different locales (simultaneously).

What's New in Version 1.0.6

This version of **Flashes Anywhere** fixes one small bug. On the select flashes forms, trying to filter on flashes using the flashes text field could result in an 'Inconsistent Datatypes' error. This error would only occur on Clarify versions before version 12, and only on Oracle databases. Simply put, Oracle does not allow filtering on LONG datatypes. As of Clarify 12, CLOBs are used instead of LONGs, so it does work. The code will now check the database type and schema revision, and hides the alert_text filter box, and changes the label appropriately when needed.

To apply this patch, copy the following 2 files to your machine:

- ❑ flash2801.cbs
- ❑ flash_globals.cbs
- ❑ flash_sqldb.cbs

Recompile these files into the database. You will need to re-start the Clarify Clients for this change to take effect.

What’s New in Version 1.0.5

This version of **Flashes Anywhere** fixes one small bug. On the select flashes forms, when querying for flashes and filtering on non-string fields, the code will now use an “equals” operator rather than a “starts with” operator. Previously, only “starts with” operators were supported, which would result in incorrect SQL being generated when filtering on numeric columns.

This has been fixed.

To apply this patch, copy the following 2 files to your machine:

- ❑ flash2801.cbs
- ❑ flash2804.cbs

Recompile these files into the database. You will need to re-start the Clarify Clients for this change to take effect.

What’s New in Version 1.0.4

This version of **Flashes Anywhere** fixes two small bugs. The first is that the *get_flash.cbs* module had a small bug whereby flashes without date ranges would not be found when queried. This has been fixed.

Version 1.0.4 of **Flashes Anywhere** also renames function “get_table_num” to “get_table_number. This change is required for Flashes Anywhere to maintain compatibility with other First Choice products.

This update is only needed if multiple First Choice Products are installed on the database, and then, only if a “Type Mismatch” error occurs when using Flashes Anywhere.

To apply this patch, copy the following 4 files to your machine:

- ❑ flash2801.cbs
- ❑ flash2804.cbs
- ❑ flash_sqldb.cbs
- ❑ get_flash.cbs

Recompile these files into the database. You will need to re-start the Clarify Clients for this change to take effect.

What’s New in Version 1.0.3

Version 1.0. of Flashes Anywhere fixes a small issue with the *get_flashes_list* function. Due to an issue with how the baseline Clarify *CDate* function works with epoch dates (such as those that show up as */*/* *:*:*), an Invalid Procedure Call error could be returned. This has been fixed. In addition, an Index out of Range error could also occur. Both of these issues have been fixed.

Only one file has changed. *Get_flash.cbs* is slightly modified. To apply the patch, copy the new version of the file to your machine, recompile it in the database, stop all Clarify clients, and start them again.

What’s New in Version 1.0.2

Version 1.0.2 of Flashes Anywhere fixes one small bug when posting flashes where there are two output columns defined.

Only one file has changed. *Flash2802.cbs* is slightly modified. To apply the patch, copy the new version of the file to your machine, recompile it in the database, stop all Clarify clients, and start them again.

What’s New in Version 1.0.1

Version 1.0.1 of Flashes Anywhere contains one small change necessary for Clarify 10.0 and later. Clarify modified some of their metadata tables, and the change in this product allows Flashes Anywhere to work with Clarify 10.0 and later. If you are on earlier versions of Clarify you may make this change (it will do no harm), but it is not necessary.

Only one file has changed. *flash_sqldb.cbs* is slightly modified. To apply the patch, copy the new version of the file to your machine, recompile it in the database, stop all Clarify clients, and start them again.

Flash Tables

Flashes Anywhere allows you to enable any object (table) or view in Clarify to have flashes associated with them. Clearly, some knowledge of the data model is still required.

In addition to assigning flashes to tables, **Flashes Anywhere** allows you to have flashes associated with views. At first glance, it may not be obvious why you would want to associate flashes with views, rather than tables, but there is a very good reason.

Suppose that you wish to have flashes on the site table. Under normal circumstances, you would just associate the flashes with the site table. But you might want an end-user to be able to either sort or view the sites by attributes other than those on the site table. For example, you might want to find all of the sites in a particular state, or in a particular country. Using the base site table, these attributes are not available to you.

If you use, a view (such as site_view) however, all of these fields are visible. You simply have to specify the main table of the view (in this case site) and the objid column for that table in the view. Please see below for more information.

Note: If you use a view for the flash, you must, when you display the flashes, tell FA that you are displaying flashes off of a view. This is easy to do, but you must remember it, or you will not display the data correctly.

Query Columns

When you specify a table or view to be flash-enabled, you must specify at least one query column. These columns are used when you are using both the *Select Flash* form (to find flashes for a specified object), and the *Flashes* form (to locate the objects that will be associated with a flash).

For example, if you are placing flashes on a case object, you might very well set up the *Case ID* and *Title* (as well as other columns) as columns on which you might want to query. This way, you can find cases (or all flashes) based on a *Case ID* or case *Title*.

If the columns that you wish to use are not present in the base table, you may very want to use a view.

You may select from one to five query columns. For each column that you select, you must enter both a label for the query column, and select the column from the dropdown list. You must use the columns in ascending order, starting with 1. In other words, if you want to use 2 query columns, you should use the first and second one. If you want 3, use the first three.

Output Columns

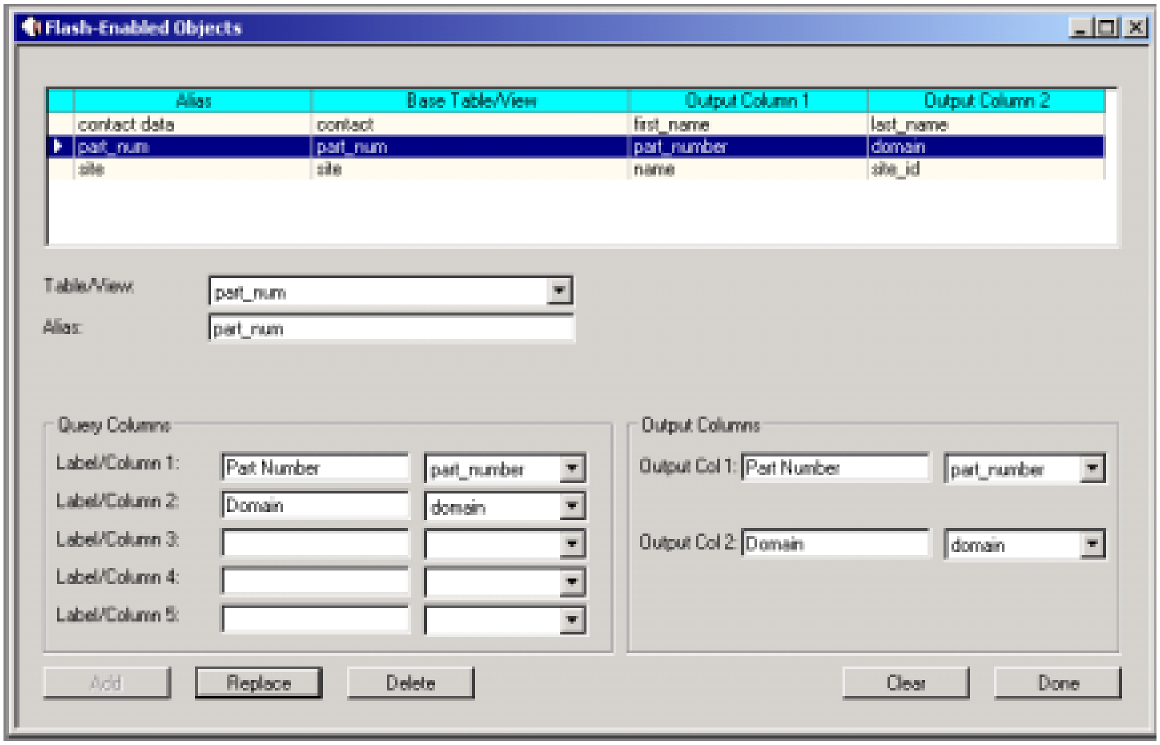
For each table or view you decide to make flash-enabled, you must specify at least one (and as many as two) output columns. These are output columns from the table or view that are used to describe the object. For example, for a case object, you might decide to show the case ID number and the case title. When you display a flash, and the objects that are related to it, you would see the object name (case), and the two data values for the output columns.

It is suggested that you always show two output columns for each object.

Flashes Anywhere Forms

This section describes the **Flashes Anywhere** forms provided with the product. The first form described is an administration form, whereas the other forms are end-user forms.

Flashes Anywhere Administration Form



Alias	Base Table/View	Output Column 1	Output Column 2
contact data	contact	first_name	last_name
part_num	part_num	part_number	domain
site	site	name	site_id

Table/View: part_num

Alias: part_num

Query Columns

Label/Column 1:	Part Number	part_number
Label/Column 2:	Domain	domain
Label/Column 3:		
Label/Column 4:		
Label/Column 5:		

Output Columns

Output Col 1:	Part Number	part_number
Output Col 2:	Domain	domain

This form is used to administer the flash-enabled objects for **Flashes Anywhere**. Using this GUI, an administrator can describe those objects which may have flashes associated with them.

To add a new flash-enabled object, the administrator selects the object from the *table/view dropdown list.* By default, the alias field will be set to the table or view name, but that can be changed. The alias is the field that is displayed to the end users for the table. So, for example, if you add flashes to the probdesc table, you more than likely will want to change the alias to something that the users would recognize, such as *Solution*. If you select a view for flashes, you must select, via the dropdown list box that will be displayed, the objid field of the driving (main) table of the view.

Finally, you must enter at least one query column and one display column. You must enter both a display label, as well as the column name.

When you have entered all this data, you may then press the **Add** button to add the object. After this process, you may then add/modify/remove flashes for that object.

You may modify an existing flash-enabled object by selecting the row in the grid at the top of the form, modifying the items at the bottom of the form, and pressing the **Replace** button. You may similarly delete flash-enabled objects.

The **Clear** button clears all the data in the fields at the bottom of the form, and allows for clean entry. When you are finished with this form, press the **Done** button or the **X** box at the top right of the form.

Flash Form

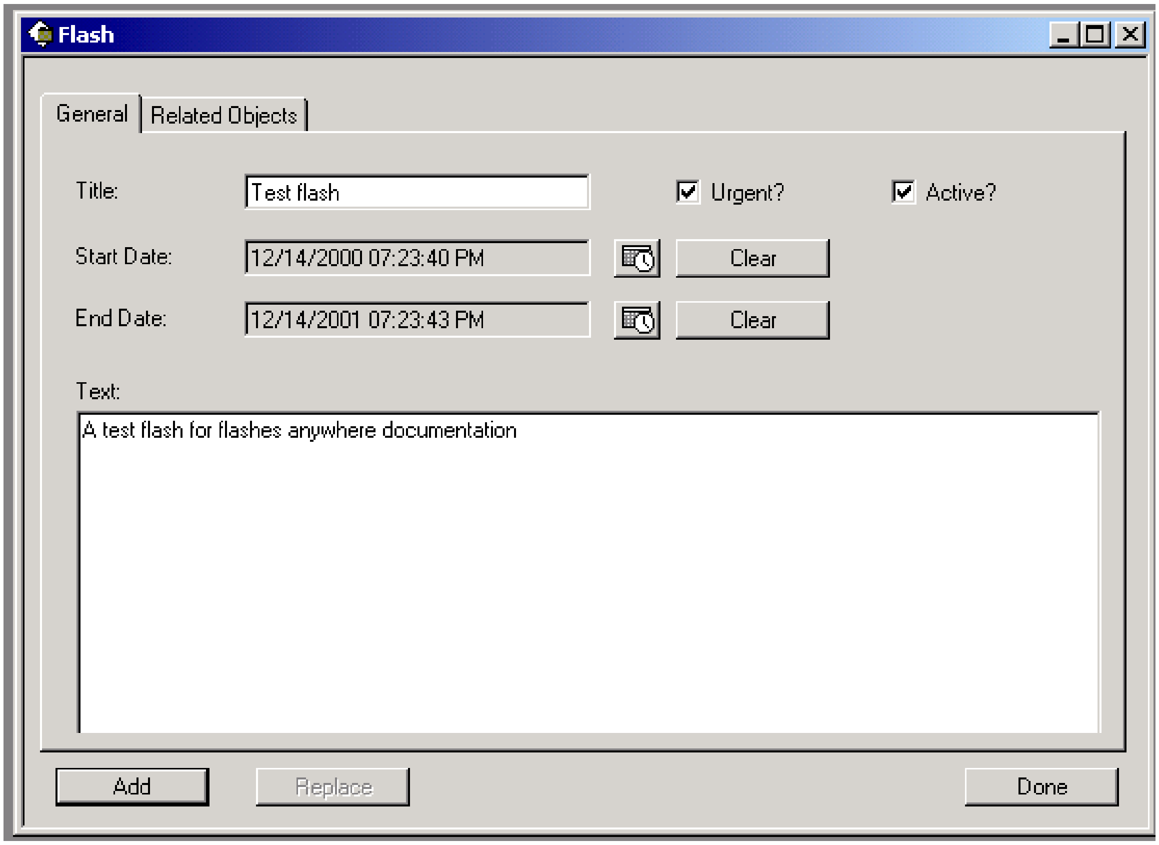
This form is used to create and modify flashes with **Flashes Anywhere**. The form can be displayed from three different locations:

1) From a new menu item installed as part of **Flashes Anywhere**
2) From the *New* button on the *Select Flashes* form
3) From double-clicking a flash (for example, in the provided sample code for the New Case form

The form has two tabs which are used to enter data about the flash. The form also has three buttons. After the flash data has been entered, you may press the **Add** button to create a new flash. If you are modifying an existing flash, you may press the **Replace** button. Finally, when you are finished with the form you may leave it by pressing the **Done** button.

There are two tabs on this form. The following two sections describe them.

General Tab



Title: Test flash

☑ Urgent?

☑ Active?

Start Date: 12/14/2000 07:23:40 PM

End Date: 12/14/2001 07:23:43 PM

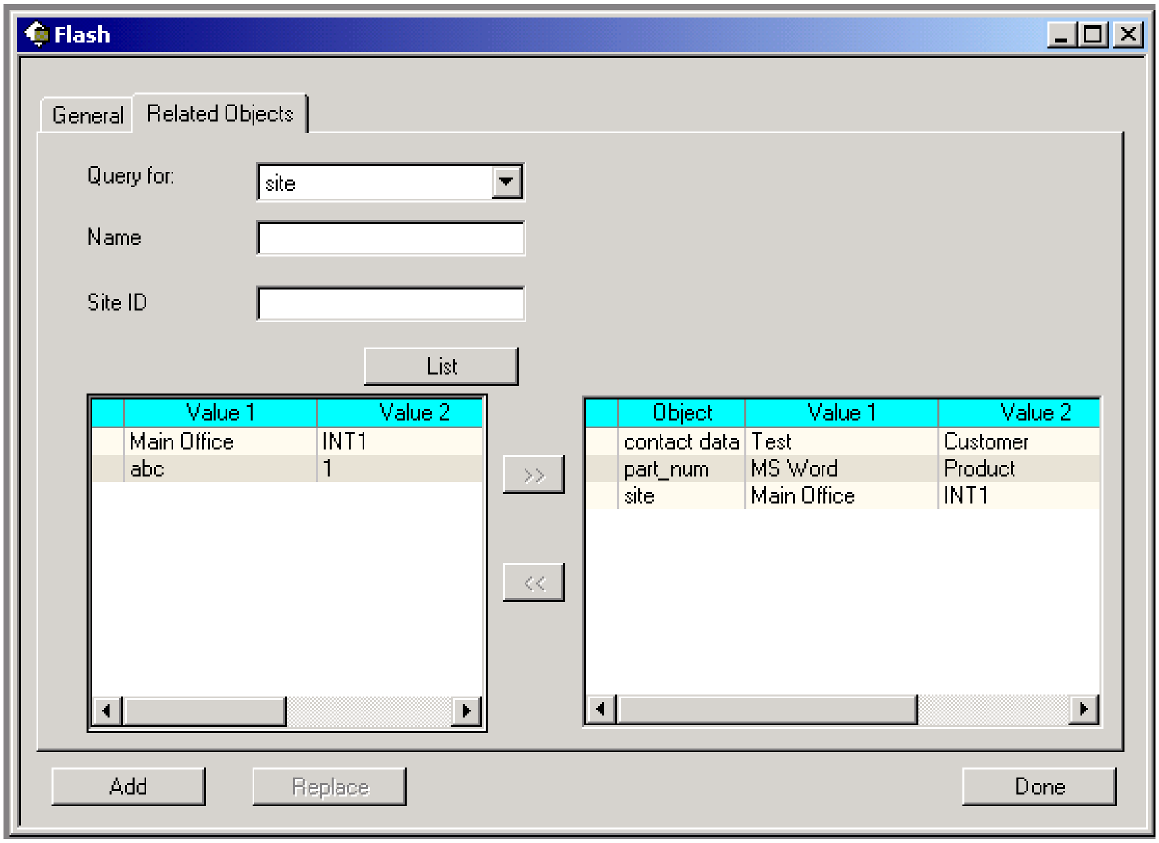
Text:

A test flash for flashes anywhere documentation

This tab is used to enter general information about the flash. The following information can be entered: The flash title (which is required), general text about the flash, if the flash is active or not, if the flash is urgent or not, and an optional date range for the flash. If you enter a start date or end date for the flash, you must enter both. If no date range is given for the flash it is always considered to be active.

The data on this tab corresponds to the base Clarify flash data.

Related Objects Tab



Query for: site

Name

Site ID

Value 1	Value 2
Main Office	INT1
abc	1

Object	Value 1	Value 2
contact data	Test	Customer
part_num	MS Word	Product
site	Main Office	INT1

This tab is different from base Clarify, and it provides much of the power of **Flashes Anywhere**. Using this tab, a user can relate the flash to as many objects desired, and each related object can be from any of the flash-enabled objects.

In the example shown above, the flash is related to a contact named “Test Customer.” It is also related to a part named “MS Word”, and is related to the Main Office site. In reality, this is a silly example. One is not likely to relate the same flash to a site, contact, and part number. More realistic might be to relate a flash to 4 different sites and 2 different accounts. Of course, the beauty of **Flashes Anywhere** is that the choice is yours.

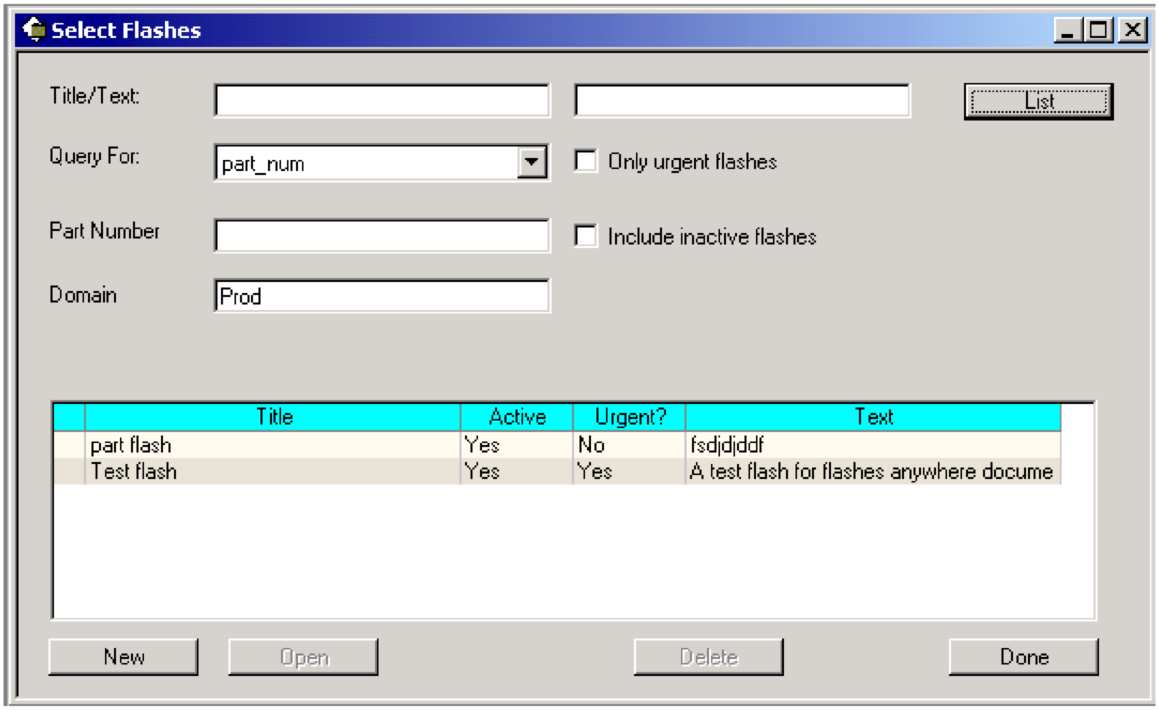
Using this tab is simple. The grid on the right of the tab shows all of the objects related to the flash. The controls on the left allow you to filter and find the objects in the database to relate to the flash.

To relate an object, select the flash-enable table or view in the dropdown list. Based on the table/view selected, from one to five filter (query) fields will be displayed. In the example above, we set it up so that site objects could be queried based on the site name or site ID.

Enter as much or as little data as your want, and press **List** to see the results. Highlight the row (or rows) in the left-hand grid, and press the **Add** button (>>). Individual rows can be added by double-clicking them.

If you wish to remove a row from the right hand grid, either double-click it, or highlight it and press the **Remove** button (<<).

Select Flashes Form



Title/Text:

Query For: part_num

☐ Only urgent flashes

Part Number

☐ Include inactive flashes

Domain: Prod

Title	Active	Urgent?	Text
part flash	Yes	No	fsdjdjddf
Test flash	Yes	Yes	A test flash for flashes anywhere docume

The *Select Flashes* form allows you to see the flashes in the system. It also allows you to edit them, delete them, and create new flashes.

To locate an existing flash in the system, you may enter information about the flash (title or text), and press the **List** button. You may also check boxes to only show urgent flashes (if unchecked, both urgent and non-urgent flashes are shown), and to show inactive flashes (if unchecked, only active flashes are shown).

Finally, you may limit your query to show flashes that are related to a specific object. First, change the dropdown list to the table or view desired. Depending on the object selected, one to five filter fields will be displayed. Enter the data and press the **List** button.

For example, if you wanted to find all the flashes that are related to parts in the *Product* domain, you could enter the data as shown above. There are two flashes that the system found. If you had entered any data in the other fields, **Flashes Anywhere** will sub-query based on those fields.

To open a flash, simply double-click it (or highlight it and press the **Open** button). To delete a flash, highlight it and press the **Delete** button. To create a new flash, press the **New** button. When you are finished with the form, press the **Done** button.

Other Issues

This section details some very important issues for **Flashes Anywhere**.

Displaying Flashes

There is very little value in creating flashes (and relating them to objects) if you cannot display them. Part of the beauty of **Flashes Anywhere** is that the display of flashes (as the title suggests) can be anywhere.

You can display flashes on any form of the system, and at any time desired. For example, you can show flashes when a form is loaded, or when a form is closed. You may display flashes when a button is clicked, or a when a dropdown list is selected. Finally (and most commonly), you can display flashes when a message is received.

How, where, and when you display the flashes is up to you.

To assist with this effort, **Flashes Anywhere** includes a new API named get_flashes (and one called get_flashes_list) that allow you to easily find the flashes that are related to one (or more) specific objects in Clarify. For example, suppose that you want to see all of the flashes that are related to a specific contact or a specific site, or a specific part. Using this API you can return back a list of flashes that can be displayed.

The manual page for this API is included later in this document.

Sample New Case Form

FA includes a number of samples, written to illustrate how to use the product. The first of these is a sample New Case form.

This sample code displays flashes (on a new tab) whenever you perform a Find Caller operation. When the Find Caller is complete, the form then attempts to find flashes for the selected contact or site. If it finds any, it then places them on the new flash tab, and fronts the tab for your viewing.

The form also queries for flashes whenever you select a part (or site_part) for the case. If you select a part or site_part (from the **product/contract** tab, or the **Serial No** button), the form queries for flashes on the contact, site, or part_num.

Note that the example assumes that you have used the administrative GUI to create flash-enabled objects for the contact, site, and part_num tables.

If you ever want to drill down and view the flash details, simply double-click the specified flash.

Note: The sample code is named flash411.cbs, but it can just as easily be used on form 775 if you use the Service Manager product.

Note: This form is for illustration purposes only. You may very well wish to modify it, or add more function to it.

To apply the customization, perform the following steps:

1. Start the UI Editor
2. Make sure that the proper resource configurations are created
3. Save a new user version of form 411, and all tabs of form 411
4. Add the form 411 and the new tabs to the resource configuration(s) desired
5. Add two new contextual objects to form 411, and then propagate them to the tabs
    a. LOR_FLASH – Type of “alert”
    b. LOS_FLASH – Type of “List”

6. Create a new child form of form 411. Make sure it is the same size as all of the other tabs of form 411. Add a grid to the tab. The source of the grid should be the contextual object *LOS_FLASH*. The following columns from the contextual object *LOR_FLASH* should be placed in the grid:

Header	Contextual Object	Field Name	Data Type
What From?	LOR_FLASH	disp_from	String
Title	LOR_FLASH	title	String
Text	LOR_FLASH	alert_text	String
Urgent?	LOR_FLASH	hot	Yes and No

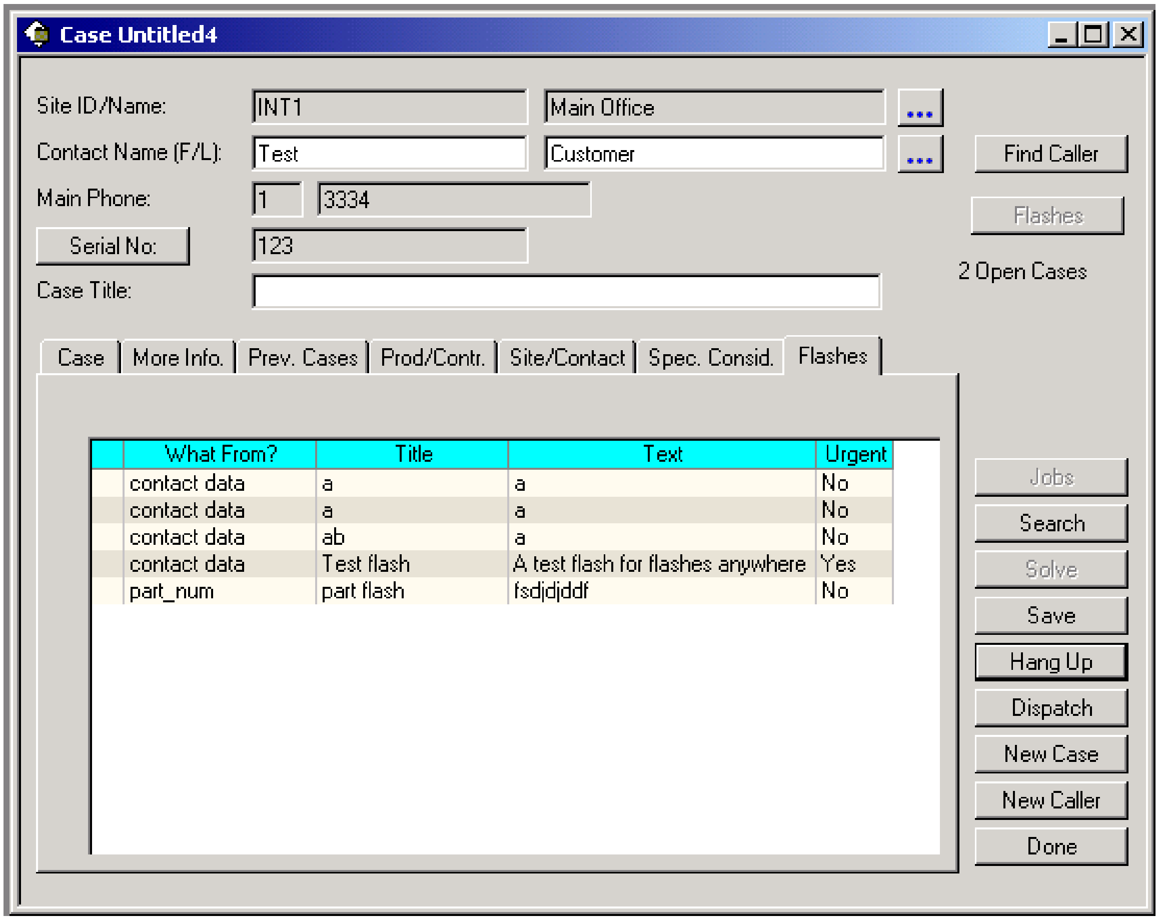
   You may put other columns in the grid, but the first column is important. It contains the “reason” that the flash is displayed.

   For the purposes of the demo code, this form is assumed to be form 1005. If it is a different number, you must edit flash411.cbs and change the line that contains “TAB_D_1005.Value = True”. Change the 1005 to your form number.

7. Add the new child tab to the resource configuration(s)

8. Compile the flash411.cbs and flash1005.cbs module against the two code forms. If you use a different form number other than 1005, you may wish to modify the name of the flash1005.cbs module, but it is not required.

A sample of the sample New Case flash display customization is as follows. This tab is displayed after the Find Caller is performed, and a site_part is selected for the case. Flashes for both the contact and the part number are displayed.



Site ID/Name:	INT1	Main Office
Contact Name (F/L):	Test	Customer
Main Phone:	1	3334
Serial No:	123	
Case Title:		

2 Open Cases

What From?	Title	Text	Urgent
contact data	a	a	No
contact data	a	a	No
contact data	ab	a	No
contact data	Test flash	A test flash for flashes anywhere	Yes
part_num	part flash	fsdjdjddf	No

Sample Case Form

Flashes Anywhere also includes a sample code module for the Case Edit form (420). Note that it can also be used for form 776 if you use the Service Manager Product.

Like the sample above, this code module displays flashes for contact, site, and part_num. It is used to illustrate how to display flashes on the Form_Load event. Like the sample above, a new tab is needed to display the flashes. In the sample provided, the form is numbered 1007.

You should install the customization using the steps above. Substitute 420 for 411 and 1007 for 1005.

View Flashes for One Object

In base Clarify, several forms have **Flash** buttons located on them. The purpose of these buttons is to show the flashes for the selected object. For example, if you open a site, and press the flashes button, you can see the flashes for that site, and can add new flashes for that site.

The objects that have flashes in base Clarify are: contact, site, contract, lead, opportunity, and bus_org (account).

You have two choices with **Flashes Anywhere**. You can remove these buttons, and manage flashes with the "New Flash" and "Select Flashes" menu items. The buttons are not needed for FA.

Alternatively, you can replace the code for these buttons with code that will call on the **Flashes Anywhere** form.

What you **MUST NOT DO** is to leave the buttons the way they are. They will create flashes that are not compatible (in their data model) with FA. If you do create flashes this way, you can update the flashes with the upgrade script (described later in this document). But you should make sure to get rid of the default buttons, or override them.

Flashes Anywhere also provides a sample of how to do this. The form flash717.cbs is provided to show how to have a **Flash** button call on the *Select Flashes* form.

To compile this code, make sure to save a new user version of the Site form (717). Add it to the proper resource configuration(s). Then compile flash717.cbs against it.

Note: You may also add new buttons to forms for flashes, using the sample provided. For example, if you have flashes on parts you might choose to add a **Flashes** button to the part number form. You are not required to be limited to the buttons provided by Clarify.

Note: If you do not wish to use Flashes on a particular object (for example, contact) that Clarify supports flashes for, you should remove the **Flashes** button on that form.

API Reference

API Names

get_flashes

get_flashes_list

Included in

get_flash.cbs

Standard API

```
Public Function get_flashes(obj_name1 As String, obj_id1 As Long, _
                            obj_name2 As String, obj_id2 As Long, _
                            obj_name3 As String, obj_id3 As Long, _
                            obj_name4 As String, obj_id4 As Long, _
                            obj_name5 As String, obj_id5 As Long, _
                            ret_list As List) As Integer

Public Function get_flashes_list(obj_names As List, obj_ids As List, _
                                 ret_list As List) As Integer
```

Description

These APIs are used to find the flash(es) that are related to the specified object or objects. You can specify 1-5 objects (with get_flashes) or an unlimited number of objects (with get_flashes_list). These are specific rows in tables. For each, you must specify the table or view name and the objid of the row.

Each API returns a list of flashes. Each item in the list is a row from table_alert that is a single flash. The column “disp_from” (which is empty in the database) is filled with the alias_name of the reason that the flash is included in the list. For example, if a flash is in the list because it is related to the specified part number, the alias of the part_num table will be placed in that column in the returned row.

Parameters

Parameter Name	Required?	Description
obj_name1	Yes	Table/view name for object 1
obj_id1	Yes	Objid of object 1
obj_name2	No	Table/view name for object 2
obj_id2	No	Objid of object 2
obj_name3	No	Table/view name for object 3
obj_id3	No	Objid of object 3
obj_name4	No	Table/view name for object 4
obj_id4	No	Objid of object 4
obj_name5	No	Table/view name for object 5
obj_id5	No	Objid of object 5
obj_names	Yes	List of table/view names for the objects
obj_ids	Yes	List of objids for the objects
ret_list	Yes	Return list of flashes

Returns

Value	Meaning
0	No errors

Examples

- Get the flashes for the site with objid of 2684357

```
Dim ret_int As Integer
Dim ret_list As List

ret_int = get_flashes(“site”, 268435457, “”, 0, “”, 0, _
                      “”, 0, “”, 0, ret_list)
```

- Get the flashes for the site with objid of 2684357 and a site with objid of 268435458 and a contact with objid of 268435600

  ```
  Dim ret_int As Integer
  Dim ret_list As List

  ret_int = get_flashes(“site”, 268435457, “site”, 268435458, _
                        “part_num”, 268435600, “”, 0, _
                        “”, 0, ret_list)
  ```

- Do the same with the list version of the API

  ```
  Dim ret_int As Integer
  Dim ret_list As List
  Dim obj_names As New List
  Dim obj_ids   As New List

  obj_names.ItemType = "string"
  obj_ids.ItemType = "long"

  obj_names.AppendItem "site"
  obj_ids.AppendItem 268435457
  obj_names.AppendItem "site"
  obj_ids.AppendItem 268435458
  obj_names.AppendItem "part_num"
  obj_ids.AppendItem 268435600

  ret_int = get_flashes_list(obj_names, obj_ids, ret_list)
  ```

Upgrade Script

When you install **Flashes Anywhere** it is possible that you will have flashes already entered in your system. If you do not have any flashes in your system, please skip this section.

The flashes in baseline Clarify are not compatible with **Flashes Anywhere**. If you already have flashes in your system you have two choices:

1. Delete all of the existing flashes, and create them again with **Flashes Anywhere**. If you do this, you can skip this section
2. You can run the upgrade script

Flashes Anywhere provides an upgrade script that will convert existing flashes from Clarify format **to Flashes Anywhere** format.

Before you run the upgrade script, you **MUST** create, for each table that has flashes in Clarify, a flash-enabled object in the FA administrative GUI. For example, if you have site flashes in Clarify, you should create “site” as an object for FA. **YOU MUST ALSO MAKE SURE TO LEAVE THE ALIAS THE SAME AS THE TABLE NAME**. After you run the upgrade script, you may also then change the alias name to anything you want.

The upgrade script is a ClearBasic script. To run the upgrade script, type the following:

```
<path>\cbbatch –user_name <user>
               -password <pass>
               -db_server <serv>
               -db_name <db>
               -f <path2>\fa_upgrade.cbs
               -r fa_upgrade
```

Where:

<path> Is the path to the dataex program (usually the Clarify rule manager directory)

<user> Is the system administrator user

<pass> Is the system administrator password

<serv> Is the database server name

<db>   Is the database name

<path2>Is the path to the fa_upgrade.cbs file

The script will print out status of the upgrade. The following is a sample output run.

```
Upgrading Clarify data for Flashes Anywhere
Process started          : 12/14/2000 09:02:13 PM

Upgrading flashes...
Cannot process account flash. There is no FA flash record set up for account.

Processing Completed     : 12/14/2000 09:02:14 PM
Number of Flashes Considered: 8

Number of Warnings       : 1
Number of Flashes Bypassed  : 5
Number of Flashes Upgraded  : 2
```

Note: The warning shown above will be shown if you do not create a flash-enabled object for the table in the FA administrative GUI.

Special Considerations

Special considerations are an older, less useful version of flashes. Base Clarify supports both flashes and special considerations. While you can continue to leave special considerations on your forms in Clarify, it is suggested that you remove them from the New Case and case forms, as well as from the Site, Contact, and Contract forms.

Flexible Deployment

Flashes Anywhere is written to be compatible with flexible deployment. If you wish to run the product in three-tier mode, you must edit the supplied directives file to comment the two-tier line, and uncomment the three-tier line. This compiles a global module on the app server. For detailed instructions, please see the compiling code section.

Implementation

This section lists the requirements for installation along with all the files included with the product and information about new strings, tables, relations, forms, etc. It also addresses any other implementation considerations.

Strings

FA utilizes a set of strings for labels, captions, and messages. Every string that is displayed by FA is contained in a new database table, and the input for that table is contained in the file *flash_strings.dat.* To modify any of the strings, edit the *flash_strings.dat* file with a text editor, and change the string. Then import the file as detailed below.

There are two primary modifications that are commonly made to this file:

- **Changing the English value for the string**. Simply locate the string object in the file and change the STRING="<the string>" line. Make sure to retain any parts of the string that start with a percent sign and contain a number. For example, %1s, %2d. These are used for parameter substitution.
- **Adding a new version of the string for another locale**. Make a copy of the English version of the string (from the OBJECT line to the END_OBJECT line). Change the locale from EN_US to the desired locale, and change the STRING="<the string>" line. For more information, contact First Choice Software.

Forms

Flashes Anywhere supplies a number of new forms. They are:

Form Name	Form Number	Description
Administration Form	2800	Set up FA tables/views
Select Flashes Form	2801	Select and locate flashes
Flash Form	2802	Add/modify flashes
General Tab	2803	General information for flashes
Related Objects Tab	2804	Objects related for flashes
Message Box	1947	Internationalized message box

Menu Items

The following new menu items will be added to the Clarify system. These will be added in the ClearBasic method "App_Initialize" in the global module. The necessary code is located in a CB globals module *flash_global.cbs*. This module will be compiled against the database.

Menu	Menu Item	Description
Policies and Customers Setup	Flashes Anywhere	Allows an administrator to set up flash

	Admin	tables and views
Clear Support New	Flash	Create a new flash
Clear Support Select	Flashes	Select flashes

Requirements

This version of **Flashes Anywhere** requires the following:

Clarify Version: 5.0 or later

Clarify Tools: Data Dictionary Editor (ddeditor)
User Interface Editor (uieditor)
Clear Basic Compiler (cbex)
Data Exchange (dataex)

Other Tools: your favorite unzip tool

Packaging

Flashes Anywhere is shipped to you as a zip file .

Files Included in Flashes Anywhere

The following files/directories are provided with this product:

File Name	Purpose
cb	Directory containing the ClearBasic code
docs	Directory containing documentation
files	Directory for data files to be imported
forms	Directory containing the forms files
schema	Directory containing the schema modifications

The following files are provided in the *cb* directory:

File Name	Purpose
flash.dir	Clear Basic directives file for compilation
flash2800.cbs	Source code for the FA administration form
flash2801.cbs	Source code for the Select Flash form
flash2802.cbs flash2803.cbs flash2804.cbs	Source code for the Flash form
flash411.cbs flash1005.cbs	Source code for New Case form sample
flash420.cbs flash1007.cbs	Source code for the Case Edit form sample
flash717.cbs	Source code for the Site form sample
1947.cbs	Internationalized message box source code
flash_global.cbs get_flash.cbs	Global source code for FA
flash_sqldb_fd.cbs	Global source code for flexible deployment systems.
string.cbs	Routines for I18N strings
fa_upgrade.cbs	Upgrade routine for flashes

The following files are provided in the *docs* directory:

File Name	Purpose
flash_user.pdf	This document

The following files are provided in the *files* directory:

File Name	Purpose
msg_box_strings.dat	I18N strings for the I18N message box form
flash_strings.dat	I18N strings for the FA forms
flash_data.dat	I18N locale information

The following files are provided in the *forms* directory:

File	Purpose
fl2800.dat fl2801.cbs fl2802.cbs fl2803.cbs fl2804.cbs	Forms for FA
1947.dat	Internationalized message box form

The following files are provided in the *schema* directory:

File	Purpose
flashes.sch	Schema changes for new tables, columns, and relations

Installation

FA files may be installed on any server machine that can execute the Clarify Data Dictionary Editor (ddeditor), the Clarify User Interface Editor (uieditor), the Clarify Data Exchange Tool (dataex) and the Clarify ClearBasic Compiler (cbex). The ClearBasic source files (*.cbs) should be installed in the same directory as the directives file (query.dir).

Once the install package is unzipped, the following tasks need to be performed during the installation of FA:

Update the Clarify Schema
Import configuration files
Import updated forms
Save other forms
Set up resource configurations
Compile ClearBasic code
Set up environment variables
Upgrade flashes (optional)

Note: It is highly recommended that the FA product be installed on a test system to become familiar with its operation before installing it on a production system.

Installation Tree

It is recommended that the zip file containing the FA files be uncompressed into a fchoice subdirectory created at the top of the Clarify install tree. For example, if the target Clarify server install tree is “c:\clarify”, then:

Switch to “c:\clarify” directory.
Create an “fchoice” directory, if necessary.
Switch to “fchoice” directory.
Create a “FA” directory, if necessary.
Switch to “FA” directory.
Unzip into “c:\clarify\fchoice\ FA” directory.

After uncompressing, the following installation tree would be established:
C:\clarify\fchoice\ FA

Updating the Schema

This section details the steps necessary to modify the Clarify database schema for FA.

Backup the Database

Prior to making schema changes, it is always a good idea to backup your database using standard backup procedures.

Import the New Schema

To make the required schema changes:

1. Start the Data Dictionary Editor.
2. Choose *Save To File* from the *File* menu.
3. Type **my_db.sch** and press the *OK* button.
4. With a text editor, edit the *my_db.sch* file (it will be stored in the DD Editor directory) and make the changes listed in the flashes.*sch* file provided in the FA install tree.
5. Choose *Apply Changes* from the *Actions* menu.
6. Select the *my_db.sch* file.
7. When asked, press *Proceed*.
8. Review the results presented. If any errors are found, fix them and repeat steps 5-7.
9. If there are no errors, press the *Apply Changes* button.
10. On the next form press the *Upgrade* button.
11. When the upgrade completes, press *Done*.
12. Exit the Data Dictionary Editor

Note: For earlier versions of Clarify, you may have one (or more) indexes on table_alert that are defined on relations to other tables (such as site) and that are **UNIQUE**. If these exist, they must be dropped. Newer releases of Clarify have removed these indexes.

Import the Data Files

There are several files that must be imported into the Clarify system with the Data Exchange tool. These files are:

msg_box_strings.dat	Internationalized strings for message boxes
flash_strings.dat	Internationalized strings
flash_data.dat	Locale data

To import the file, execute the following command:

```
<path>\dataex
        –user_name <user>
        -password <pass>
        -db_server <serv>
        -db_name <db>
        -imp <file>
```

Where: <path> Is the path to the dataex program
<user> Is the system administrator user
<pass> Is the system administrator password
<serv> Is the database server name
<db> Is the database name
<file> Is the name of the file to import

Note: If the *<file>* is not in the current directory, the path to the directory it is in must precede it.

Note: Each file should import with 0 errors and 0 warnings. If there are any errors or warnings, the dataex.mes file should be investigated for the reason.

Import the User Interface Forms

FA is provided with a set of new forms. In addition (depending on which samples you may apply – see the sections above) there are some existing Clarify forms that must have new user versions saved for them.

The new forms to be imported:

- 1947.dat
- fl2800.dat
- fl2801.dat
- fl2802.dat
- fl2803.dat
- fl2804.dat

The form files are imported with the following command:

```
<path>\dataex
        –user_name <user>
        -password <pass>
        -db_server <serv>
        -db_name <db>
        -imp <file>
```

Where: <path> Is the path to the dataex program
<user> Is the system administrator user
<pass> Is the system administrator password
<serv> Is the database server name
<db> Is the database name
<file> Is the name of the file to import

Note: If the *<file>* is not in the current directory, the path to the directory it is in must precede it.

Note: Each file should import with 0 errors and 0 warnings. If there are any errors or warnings, the dataex.mes file should be investigated for the reason.

Save Versions of Existing Forms

You may wish to save new user versions of forms (for the sample code).

Note: If you already have saved new user versions of these forms, you do not have to save new user versions of these forms. Simply integrate the supplied FA code for those forms with your customized code.

To save new user versions of forms, perform the following steps:

1. Start UI Editor
2. In the Form Select form, change the form selector (dropdown list) to **ID**
3. Enter the form number in the text box
4. Press the *List* button
5. Double click on the line in the grid
6. When the form posts, select the *Save Form As…* menu item from the *File* menu
7. Enter *fl1.0* in the text box and press the *Save* button
8. Repeat steps 3-7 for any other forms
9. Exit UI Editor

Set up Resource Configurations

After all of the form files have been imported successfully, they must be added to the proper resource configuration(s) in UI Editor.

Note: If you have no custom resource configurations in your database, you must add one. Please see the Clarify UI Editor documentation for more information.

First decide which user or groups of users should see the new versions of the imported forms. Then add the forms to the resource configuration(s) using these directions:

1. Start the User Interface Editor.
2. Choose *Resource Configs…* from the *Select* menu (it's File/Open -> resource configuration for Clarify 6.0+)
3. Press the *List* button.
4. Select the resource configuration you want to modify and press the *Open* button.
5. Make sure the dropdown list has "User Version" selected. Type *fl* in the text box to the right of the "Starts With" dropdown list and press the *List* button.
6. Select the forms you wish to add to the resource configuration from the left grid control, and press the *Copy>>* button.
7. If any confirmation dialogs appear, press the *OK* button to continue.
8. Repeat steps 5-7 with *fc* (for form 1947)
9. Press the *Replace* button.
10. After the save is completed, press the *Done* button.
11. Repeat steps 4-10 for any other resource configurations you wish to modify.
12. Exit the User Interface Editor.

Compile ClearBasic Code

You can now compile the supplied Clear Basic code against the forms you have imported.

Before you compile the code, you should edit the flash.dir file supplied to point to the location of the FA cbs files. Verify that the Clarify version (the 1st number after the form number for the lines that contain a letter "F") is the same as the form that is saved in the desired resource configurations. Also, the user version (the 2nd string after the "F") should be the same. Comment out any lines that represent forms and code that haven't been added to the database. To comment out a line, put a single quote as the first item on the line.

Note: There is one line that must be commented, and one that must be uncommented in the directives file, depending on if you use Flexible Deployment (FD), or not. The default line (for normal 2-tier Clarify) is uncommented. If you are using FD, you must comment this line, and uncomment the FD line. There are instructions in the directives file that explain how to do this.

To compile the code modules, make sure that the code modules (*.cbs) and the directives file (flash.dir) are located in the same directory. Change directories to that location and enter the following command to compile the code:

```
<path>\cbex –dir flash.dir  –batch  –overwrite
```

Where: <path> is the location of the Clarify cbex program.

Note: A login may be required if a clarify.env file is not in the compilation directory.

When the compilation is completed, look at the *cbex.log* file to check on the status of the compilation. There should be two rows for each file compiled that start with the word **SUCCESS**. If there are any failures, check the above instructions to make sure that your compilation environment is set up correctly.

Global Clear Basic Code

If you already have a global module with an *InitializeApp* routine in it, you must incorporate the code in *flash_global.cbs* into the existing one. Clarify only allows one InitializeApp routine to exist. If you do not have such a routine, you may simply compile the provided *flash_global.cbs* module.

Upgrading Flashes

If you have flashes in your existing database you must upgrade them to work with **Flashes Anywhere**. Please see the section above for more information.

Environment Variables

On each client machine that is running the Clarify client you should set up the environment variable **LC_ALL**. This variable should contain the locale name that the client desires for displaying any internationalized strings. First Choice Software uses the standard X/Open locale names. By default, US English should be used. The value to use for US English is:

```
LC_ALL=EN_US
```

Note: If you only want to use US English, you do not have to set up the environment variable. Please see the section below for more information.

Limitations

There is one known limitation with the FA product.

The limitation has to do with internationalization. There are some limitations in the customization tools that Clarify provides. It is not currently possible to change grid column headers at run time.

As such, the strings “Value 1” and “Value 2” are used for the column headers for the related items tab.

This has been opened as an issue with Clarify. When it is corrected, a patch will be issued for this product. In the meantime, workarounds are available. If this is an issue, contact First Choice Software.

Internationalization

Flashes Anywhere is internationalized and localized, using the First Choice Software Inc.’s Internationalization library. Thus, all strings that are displayed on custom forms may be changed to another language/locale, simply by changing an environment variable. Strings that are displayed can also be stored in the database in multi-locale format.

Flashes Anywhere is localized and internationalized, but it is not shipped in a translated state. Any translations to other languages are the responsibility of the customer.

There are a variety of environment variables that can be set to control various aspects of the internationalized code. **Flashes Anywhere** uses only two such variables.

LC_ALL – Sets all internationalization features to the desired locale

LC_STRING – Sets string handling features to the desired locale

Note: You may set either variable. If neither environment variable is set, **Flashes Anywhere** defaults to US English (LC_ALL = EN_US).

Note: First Choice Software uses the standard locale identifiers defined by X/Open. Each individual user may set their environment variables as they wish. Thus, two different users may be seeing **Flashes Anywhere** displayed in different languages at the same time against the same database.

Note: First Choice Software has a product named the Internationalization Toolkit available for system administrators who want to localize the strings stored in the fc_string table. Please contact us for further information about this utility.

Note: All internationalization is optional. The base **Flashes Anywhere** product is shipped in US English. No additional work must be done for that environment.

How to Contact Us

For more information about other First Choice Software, Inc. products, or if you have any questions or feedback about this product, please contact us at:

First Choice Software, Inc.
8900 Business Park Drive
Austin, TX 78759
(512) 418-2905
www.fchoice.com