FIRST CHOICE SOFTWARE

fcMerge

Version 4.0

Administration Series

Installation and Users Guide

About First Choice Software

First Choice Software, Inc. (FCS) is the premier supplier of add-on software and customizations to the Clarify Front Office suite. Founded in 1995, **FCS** is the first and only company to focus on writing applications to enhance the value of your Clarify investment. You can see our extensive product line on our website: www.fchoice.com.

Clarify/Amdocs

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

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

Many **FCS** products use the following Clarify tools:

- User Interface Editor
- Data Dictionary Editor
- Data Exchange
- Rule Manager
- ClearBasic
- ClearBasic Exchange
- ClearBasic Batch
- Clarify Business Objects
- eBusiness Framework
- Clarify Low-Level API (c)

Clarify, its product names, and tools are trademarks of Amdocs. Amdocs' trademarks may include the following:

- ClearConfigurator
- ClearConfigurator Workshop
- eSupport
- eOrder
- eResponse Manager
- e.link
- Flexible Deployment
- SalesBasic
- Notifier
- Classification Engine
- Routing Server
- ClearEnterprise Traveler
- DiagnosisEngine

8900 Business Park Drive
Austin, Texas 78759

Web: www.fchoice.com
Phone: (512) 418-2905
Fax: (512) 418-2983

Notice

First Choice Software, Inc. (FCS) intends the information in this document solely for its licensees and to aid prospective customers in evaluating **FCS** products. Any other use is prohibited. **FCS** makes a written warranty to its licensees, as is stated in its written license agreement. **FCS makes no warranty to any other party regarding the accuracy or usefulness of the information contained in this document.** **FCS** reserves the right to modify the content of this document without obligation to notify any party. **FCS is not liable for consequential or incidental damages.**

Table of Contents

Overview

During the operation of a Customer Service System, a common problem encountered is the accumulation of "alias" or "duplicate" contacts, sites, addresses, employees, and parts in the system. **fcMerge** from First Choice Software Inc. provides an effective way of identifying, marking and eliminating duplicate objects from your customer service process.

An "alias" or "duplicate" contact is the presence of multiple data records in the database representing one individual contact, typically as the result of a minor differences in the spelling of the contact's name. An example could be the presence of three contact records "Mary Smith", "Mary J. Smith" and "Mary Joan Smith" in the database, all representing the same individual customer "Mary Smith".

Similarly, a duplicate site is the presence of multiple site records in the database representing one specific site. For example, "Clarify - Austin" and "Clarify/Austin" might accidentally both be entered into your system.

The same is true for other duplicate objects, such as addresses, employees, and parts (products).

A customer service agent may inadvertently add these aliases to the database if they cannot find a contact or site for some reason in the database, or by interfaces performing batch loads of customer/site data from other systems. The problem with these aliases is that they can skew metrics, as well as result in customer service errors when contact/site attributes (such as fax number) are changed for one alias, but not the others. As a result, it is desirable to identify and potentially eliminate these aliases from your customer service processes.

Before You Upgrade to a New Version…

You should be aware of both the version of the product you currently have as well as the version you are upgrading to.

You may upgrade several revision levels at one time. To perform this multiple-upgrade, you must create a list of files to upgrade. Find the section of the manual with instructions for upgrading from your current software revision. Make a list of the required steps. Repeat this process with each update, adding these steps to the list. If a step is required by more than one version, you only have to list it once.

When your list is completed, you should perform all the steps (in the same order you would install the product (schema first, then files and forms, then resource configurations, then code compilation) for a clean install) using the files provided with the most recent version of the product.

For example, suppose you had version 1.1 of a product, and versions 1.1.1, 1.2, and 1.3 have all been released. You now wish to upgrade from version 1.1 directly to version 1.3. Suppose that the versions require the following steps:

Version 1.1.1 requires that you compile files a.cbs and b.cbs.

Version 1.2 requires that you add a new field to the case table, and recompile a.cbs.

Version 1.3 requires that you import a data file d.dat, import a new form file, c.dat (add it to the resource configuration), and compile c.cbs.

To perform the total upgrade in this situation you would perform the following steps:

1. Add the new field to the case table (schema changes always come first)
2. Import the d.dat file file (data imports come next)
3. Import the c.dat form, and add it to the proper resource configuration(s).
4. Compile the a.cbs, b.cbs, and c.cbs files.

All of these steps would be accomplished with the files provided with the 1.3 version of the product. If you have questions about this process, please contact First Choice Software.

What's New in Version 4.0

As of version 4.0, this product has been renamed from **Contact/Site Merge** to **fcMerge**.

This is a major new enhancement release. This release includes employee merge – duplicate employees (users) can now be merged into one.

This release also includes multiple bug fixes when performing a site merge (all of these fixes are in the site_merge.cbs code) :

- When processing bus_site_role records, trap for duplicate roles, such as "owner"
- When processing contact roles, if there are no related contacts, then customer has bad data - trap for this situation
- Handle the following situation with contact roles: The same role_name exists at the master site and the alias site and the role for the alias site is the primary role. So, make the master role the primary, and we'll still delete the role that pointed to the alias
- Fixed bug when updating the primary contact role that could occur when contact was also a role at another site (that wasn't the master or alias)

To upgrade to this release, you must apply the new schema in the *merge.sch* file, including

- 3 new fields on the employee table
- 2 new relations on the employee table
- 3 new fields on the empl_user view

If you are using the employee merge functionality you should edit and import (with data exchange) the *merge.dat* file. The file contains a new activity code for "Employee Merge"

You should import the new form *1927.dat* and make sure to add it to the proper resource configurations.

Finally, you should recompile all of the ClearBasic code that comes with this product.

What's New in Version 3.1

A bug was found with site merge that could cause problems with deinstalled parts. Previously, all bins from an alias site were being moved to the master site. This included the special "deinstall_rip" bin (which is invisible to the user, but holds all of the deinstalled parts). If the master site already had a deinstall_rip bin, then it would have (after the merge) two of them, and not all of the deinstalled site_parts would be visible for reinstallation. This has now been corrected.

To apply this change, copy the new *site_merge.cbs* file and recompile. Then stop and restart any running Clarify sessions to pick up the changes.

What's New in Version 3.0

This is a major new enhancement release. This release contains a number of major new features, including:

- Address merge – You can now merge duplicate addresses into one.
- Part merge – This feature allows you to merge duplicate parts. In addition, if you have revisions of each part that have the same name (e.g. both parts have a revision of '1.0', the product performs a revision merge on that revision.
- Support for Clarify versions 11.5 and 12.0.
- Support is provided, for Contact and Site merges, for Attachments Anywhere and Flashes Anywhere.
- In previous releases, this package included versions of the 711 and 716 forms. The .dat files are no longer included with this package, as they vary from Clarify version to Clarify version. Instructions are given below, however, on how to create a custom version of the form. The sample ClearBasic code for those forms is still included, however.

To install this release, you must apply the new schema in the *merge.sch* file.

If you are using the part merge product you should edit and import (with data exchange) the *merge.dat* file. The file contains a new configuration item needed for the part merge process.

You should import the new forms *1922.dat, 1923.dat, and 1926.dat*. Make sure to add them to the proper resource configurations.

Finally, you should recompile all of the ClearBasic code that comes with this product.

What's New in Version 2.1

This version contains a useful customer-requested enhancement. The problem encountered by the customer was that, in a small number of cases, their users would make a "dumb" master/alias link. When the bulk merge occurred (later that night), the merge would successfully be completed.

But the manager wanted to know *which* user was the one to mis-mark the master and alias. In previous versions of this product, the activity logs added for merging would show the user name of the user (or process) performing the merge (not the marking). Since they used batch marking, the activity log always showed the same "user" for the merge.

The enhancement for the current version now remembers who performed the marking for each alias, and annotates the activity log with that user. So, for example, if three different users marked three aliases all to the same master, each of those users would show up in one activity log as the "merger". Note that this enhancement works for both the "Merge" activity log as well as the "Change Contact" and "Change Site" activity logs.

To incorporate this enhancement you must add the new *x_marked_by* fields in both the contact and site tables. The field description for these new fields is listed in the schema file provided with the product.

Then you must copy the new 1920.cbs, 1921.cbs, merge.cbs and site_merge.cbs files, and recompile them. Finally, you must shut down any currently-running Clarify users, and have them restart Clarify.

What's New in Version 2.0.1

This version corrects a small bug that was found with the ClearBasic methods (BulkSave methods) used in the *swap* routines in contact and site merge. If you pressed the *swap* button more than three times in a row, the BulkSave could be corrupted. The workaround for this is that the *swap* button now always performs a save. If you need to re-swap the items, you can still do that.

To apply this change, simply copy the new 1920.cbs and 1921.cbs files supplied with this build, and recompile them.

What's New in Version 2.0

Version 2.0 of this product provides support for the new relations found in Clarify 11.x. There are four new contact relations (and none on the site table). If you have an earlier version of Clarify, this release does not affect you.

To apply this release, simply copy the new *merge.cbs* and *site_merge.cbs* files, and recompile them.

What's New in Version 1.9.4

This version fixes a small bug found in the merging of sites or contacts with flashes attached to them. When an alias contact or site had an active flash, the *alert_ind* flag was not being set to 1 (active) if it was previously unset. This patch corrects this problem.

If you have run into this problem, you can use SQL to update the alert_ind flag. If you need help with this, please contact First Choice Software.

To apply the patch, simply copy the new version of the *merge.cbs* file, and recompile it. Each user should then stop and restart Clarify to see the change.

What's New in Version 1.9.3

This version adds a nice new enhancement to fcMerge. If you do not desire this enhancement, there is no requirement to apply the patch.

This version adds the ability to "freeze" contacts or sites so that they cannot be used as aliases. For more details, please read the section on freezing contacts and sites below.

To apply the patch, you must perform the following:

1) You must add a new long integer field to both the contact and site tables. The description for these "x_no_alias" fields is listed in the *merge.sch* file.
2) You must also add view fields to the "rol_contct" and "site_view" views. In each case, the name of both the view field and the "from" field is x_no_alias. The description for these "x_no_alias" view fields is listed in the *merge.sch* file
3) You should add a checkbox to both the contact and site forms (#712 and #717). How to do this is listed in the "Freezing Contacts and Sites" section under "Implementation".
4) You must copy the new versions of *1920.cbs* and *1921.cbs*, and recompile them.

What's New in Version 1.9.2

This version adds a few requested enhancements. The contact master and alias lists are now sorted by last name & then first name. The site master & alias lists are now sorted by site name.
Previously, when a master or alias was unmarked via the *Unmark* button, filtering was added (first initials of the unmarked contact) to the candidate list & the candidate *List* button was pressed. By request of many of our customers, this "feature" has been removed.

To implement this patch, copy the two new code modules from the package and recompile them. They are 1920.cbs and 1921.cbs.

What's New in Version 1.9.1

This version corrects a small bug in both contact and site merge. If you have part request headers related to an alias site or contact, the merge was correctly merging all relations, but did not reassign the normalized fields that Clarify added to the object. So it would look as if the part request was still linked to the alias site (which it was not). This has been corrected in the code. For existing part requests, you can simply reassign those fields with the site or contact filters on the part request form.

To apply this patch, you must either complete 1 or 2 of the steps below:

1) If you are using Clarify 7.x or earlier, you must add the address_2 view field. See the installation section in this document for more details. If you have Clarify 8.x or later, you may skip this step.
2) Copy the new versions of merge.cbs and site_merge.cbs provided with this package to your machine and recompile.

What's New in Version 1.9

Version 1.9 of fcMerge adds two new enhancements to the product, and provides support through Clarify 10.0. It also fixes one small bug.

The bug could occur if you have a contact/contract_schedule role record (with the same alias) defined for both the alias and master contact.

Both Contact and Site Merge have added (in the GUI) a *swap* button. If you press this button, the selected alias becomes the master, and the selected master becomes an alias. This allows you to change easily which of many aliases is the master, in case you chose the wrong one first. Previously, you had to remove the master (and aliases) back to the candidate list, and redo the process manually.

On the Contact Alias form, the *More Information* button captions have been made into the classical three dots (…). Previously, they were the older standard (>?).

Version 10.0 of Clarify only adds one new Contact relation (contact2con_pym_role), and no new Site relations. You can apply this patch, and it will work with all Clarify versions from 4.5 -> 10.0.

To implement this patch, do the following:

1) With dataex, import the two new form files supplied (1920.dat and 1921.dat).
2) Copy the new code modules from the package and recompile them. They are: 1920.cbs, 1921.cbs, merge.cbs, and site_merge.cbs.

What's New in Version 1.8

This version corrects two very tiny bugs found in the fcMerge product.

If you attempt a site_merge (and are using logistics) the "default" inventory role records (there are three for each site) were merged, when they should have been ignored. This could cause confusion in some logistics forms, where the default would not be found.

If you attempt to do a batch merge, you can see an incorrect date in the log file (and possibly in the act_entry record. This was due to a Clarify-supplied method (Now) not being Y2K compliant. It has been changed to App.CurrentDate, which is.

To apply this patch, simply copy the new versions of the *merge.cbs* and *site_merge.cbs* files, and recompile. Also, the sample program *duns_inter.cbs* is also new.

What's New in Version 1.7.1

This version corrects a tiny bug in the fcMerge product. If you merged an alias (or aliases) with a master, and then later made the alias active again (this can only be accomplished with direct SQL manipulation), the alias was still listed as an alias of the master. This could cause it to be re-merged when that is not intended.

If the normal flow of this product is followed, the situation above will not be seen.

To apply the patch that removes this small hole, copy the new merge.cbs and site_merge.cbs files to your machine, and recompile them.

What's New in Version 1.7

Version 1.7 of fcMerge provides support for Clarify 9.0. Four new relations are supported:

For Sites:

1) site2ship_sit_role

For Contacts:

1) contact2e_addr
2) contact2r_rqst
3) contact2r_rqst

Also, a small bug was found with site_merge. Certain contact_roles that should be deleted were not. This would leave the roles showing after they should be removed. That has been fixed.

To apply the patch, simply copy the merge.cbs and site_merge.cbs files, and recompile them. Users must stop and restart Clarify to see the change.

What's New in Version 1.6.2

A small bug was fixed with the contact2site_part relation. Since that is an MTM relation,the old relation was not being unrelated (until the contact was purged or archived). This could result in a display problem on the site_part record.

To apply the change, simply copy the merge.cbs file from this package, and recompile.

What's New in Version 1.6.1

Per a customer request, version 1.6.1 enhances the merging of contact roles for contact merges. Previously, all contact roles associated with aliases were deleted. With the new patch, contact roles for aliases are retained, if the master is not known at the site and with the contact role.

If the master is already known at the site with the contact role, the contact role record is deleted.

To apply the patch, simply copy the *merge.cbs* file into your code directory, and recompile it. Users will have to exit Clarify and start it again to see the change.

What's New in Version 1.6

Version 1.6 of fcMerge adds support for Version 8 of Clarify (eFrontOffice 2). This new release adds one relation from contact (contact2doc_inst), and none to the site table.

To make your contact/site merge, simply copy two files (merge.cbs and site_merge.cbs) to your code directory and recompile them. No other changes are needed.

What's New in Version 1.5

Version 1.5 adds several new enhancements.

It adds two new activity log entries. When you merge a contact or a site, you can now have an activity log (with a new code of *Merge Contact* or *Merge Site* added to the master's activity log.

If you are merging a contact, and there is a case (or cases) associated with that contact, you can now have the site for the case also changed at the same time. It will be changed to the master contact's primary site.

There are now two new configuration items that control the above two features. You do not have to use either feature – they can be turned off by changing the appropriate configuration item's *i_value* to 0.

This version of fcMerge also has a new warning message added to the Contact Alias GUI. It has been handled for those customers who use *individual* sites. If you merge a contact at an individual site, without first merging the site (into a master site), you will orphan the individual site. To get around this, the GUI now warns you if you try to merge an individual site, without first merging the contact's site.

Finally, one small change has been made to the Contact Alias form. The final column (site type) has been changed to display the four letter code (CUST, INTR, …) instead of the corresponding integer.

What's New in Version 1.4

Version 1.4 corrects one small bug, and provides support for Clarify eFrontOffice.

The bug that was reported resulted from the fact that Clarify only allows one flash (per contact or site) with a given title. So, if a master and alias BOTH had a flash with the same title, the merge process would fail.

The solution for this is that if this case arises, the merge process appends a unique identifier to the aliase's title, thus making them different, and retaining both flashes for the master.

fcMerge

fcMerge is designed to manage contact, site, address, part, and employee aliases in a Clarify system. There are two general phases of the merge process. The first is the identification and marking of a contact or site as a duplicate. The second is the processing of marked duplicates to eliminate them from your customer service processes. Both of these steps are performed from the *Contact Alias*, *Site Alias, Address Alias*, *Part Alias,* and *Employee Alias* forms.

The functioning of all of these forms is almost identical. The only difference between them (other than the fact that they merge different objects) is that the filtering is different for the different forms.

Since the forms (and processing) are so similar, only the contact merge will be described in this document. The same process can be used for the other merge forms.

Contact Alias Form

The Contact Alias form is the control panel for the alias marking and processing. With the Contact Alias Form, a Clarify user or administrator can perform the following tasks:

- Select a "master" contact using attribute filtering. The user may filter the master contact candidates by contact first name, contact last name, site id and site name. A contact will be filtered out if the contact is already marked as an alias for another contact or marked as obsolete.
- Display a list of known "alias" contacts for this master contact. Upon selection of a master contact, any alias contacts previously marked for that contact will be listed.
- Select candidate contacts using attribute filtering. The user may filter the contact candidates by contact first name, contact last name, site id and site name. A contact will be filtered out if the contact is already marked as a master or alias for another contact or is marked as obsolete.
- Mark candidate contacts as a master or alias for a master contact.
- Unmark existing master or alias contacts from a master.
- Preview the "merge process", generating a report for online viewing or printing.
- Invoke the merge process for a particular master or all masters.

Should you wish to limit invocation of the preview or merge process to administrators, this can be done via Clarify privilege classes. The Contact Alias form can be launched from the Clarify Policies and Customers *New* or *Select* Menu Bar Items.

Workflow Flexibility

fcMerge can be used in a variety of customer service workflows dealing with duplicate contacts. Before continuing, some definitions may be helpful.

Definitions

Throughout this document, you will encounter references to “master”, “alias” , “duplicate” and “candidate” contacts. They are defined as follows:

Master Contact – This is a reference contact for which there are one or more duplicates marked. During the course of eliminating duplicate contacts, this is a contact you wish to keep.

Alias Contact – This is a contact that has been marked as a duplicate in the system. A contact marked as an “alias” is an “alias” of a master contact. That is, the contact is a duplicate of the master. During the course of eliminating duplicate contacts, this is a contact you wish to discard.

Duplicate Contact – See Alias Contact.

Candidate Contact – This is a contact that has not been marked as an alias or master. Before using fcMerge, all contacts are candidates.

For example, the document overview mentioned three contacts “Mary Smith”, “Mary J. Smith” and “Mary Joan Smith” in the database, all representing the same individual “Mary Smith”. Initially, these are all *candidates*. When a user selects “Mary Smith” as the reference contact, “Mary Smith” becomes the *master* contact. When the user selects “Mary J. Smith” and “Mary Joan Smith” as duplicates of “Mary Smith”, “Mary J. Smith” and “Mary Joan Smith” become *aliases* of “Mary Smith”.

Indicating Master and Alias Contacts

The first step in eliminating “duplicate” contacts is noting in the system which contacts are “duplicates” of another. This can be performed as an administrative task or be performed by the customer service representative while taking a call (or both).

Administrative. For your business, it may make sense to eliminate duplicate contacts periodically as an administrative task. Once every few weeks or months, one or more administrative personnel use the fcMerge GUI to mark duplicate contacts in the system.

End User. On the other hand, you may build the process of eliminating duplicate contacts into your front-line customer support process. In this way, as your customer service personnel locate contacts and notice duplicates, they can immediately mark them as such using the fcMerge GUI.

Previewing the Merge Process

After specifying a set of master and duplicate contacts, an administrator or user can gather metrics regarding the number of alias contacts currently marked. Using these metrics, the administrator can decide when the merge process should be run to process the alias contacts. The metrics reported will include the number of master and alias contacts and for each alias contact the number of alerts, commit logs, users, web users, phone logs, contract-site roles, demand headers, cases where the alias is a primary or alternate contact, biz org roles, contracts and contract roles.

Invoking the Merge Process

The second step in eliminating duplicate contacts is to processes the marked duplicates. Processing means that objects (such as cases) related to contacts marked as a duplicate will be re-related to their master contact.

By processing alias contacts, the Merge Process will for each alias contact, re-associate most objects related to the alias contact with the master contact. These objects are alerts, commit logs, phone logs, contract-site roles, demand headers, cases where the alias is a primary or alternate contact, biz org roles and contracts. A user or web_user related to an alias will have their status changed to inactive. Any contact roles related to the alias contact will be deleted. After processing of the related objects, the alias contact will be marked as obsolete in the Clarify database. Marking the contact as obsolete will prevent the contact from being included in future find-caller and Merge processing. When a case is associated with a master contact, a “Chng Contact" activity log entry will be created for the case.

Note: In the default product, if a case has a contact that is merged to a contact who is not at the same site, the contact, phone number, and alternate phone will be changed. The site_part, de_part, address, and contract for the case will **NOT** be changed. This differs from baseline Clarify (change contact) functionality. To completely emulate Clarify function, edit the file *merge.cbs*, and uncomment (remove the single quote at the start of the line) from the lines that contain #If and #End.

Administrative. One method of processing marked duplicates is for an administrator to periodically perform this task from the Contact Alias GUI. Clarify privilege classes can be used to allow only administrators to perform a merge.

End User. Similarly, in your business process, it may make sense for front-line users to perform a merge during the course of processing calls. Again, Clarify privilege classes may be used to control access.

Putting It All Together

fcMerge is quite flexible in how it can be implemented into your Clarify business workflows. You can mix and match the above marking and processing techniques to best suit your needs. In all cases, when processing duplicate contacts, a detailed log is generated, whether processing is performed by an administrator or by an end-user. The log may be stored on a user's workstation, or on a central file server. In addition, when a case contact is re-related, an activity log entry is created to record the event.

Attachments Anywhere and Flashes Anywhere

Many users of **fcMerge** also use either **Attachments Anywhere** or **Flashes Anywhere**, both available from First Choice Software (and both built-in to the First Choice *fcClient* line of thin clients).

If you do use one or both of these products, it is important to be able to also merge the attachments and flashes, using the advanced data modeling used by those two products.

The contact merge process and site merge process both contain code to handle these two products. However, since the code could be dangerous to execute for other users, it is currently commented out, so that it does not execute.

If you are using **Attachments Anywhere**, you should edit the *merge.cbs* and *site_merge.cbs* files. Search for "Attachments Anywhere". There are two sections that you will find, both of which should be uncommented for execution.

If you are using **Flashes Anywhere**, you should edit the *merge.cbs* and *site_merge.cbs* files. Search for "Flashes Anywhere". There are two sections that you will find, both of which should be uncommented for execution.

Note: You can add similar processing to the items described above to other merge processes, if desired (if, for example, you have attachments or flashes on parts). If you have questions about this process, please contact First Choice Software.

Advantages

fcMerge provides a means for Clarify users and administrators to work together in the cleanup of "alias" or "duplicate" contacts in a Clarify database. The Contact Alias Form provides an easy way for Clarify users and administrators to select and mark "alias" contacts for later processing using standard Clarify filtering and selection techniques. There is no need for the user to remember and manually data enter artificial keys to indicate an alias. The fcMerge Tools provide the Clarify administrator the information needed to schedule the "alias" merge, as well as the merge function itself. Both the metrics and merge are contained in the same tool.

Implementation

This section provides detailed requirements, what files are included in this product, installation and other implementation considerations.

Overview

fcMerge is made up primarily of five forms – the Contact Alias Form, the Site Alias Form, the Address Alias form, the Employee Alias form, and the Part Alias form.

Contact Alias Form

The Contact Alias form allows a Clarify user or administrator to manage the selection, marking and processing of “alias” contacts. Since this is a Clarify form, its operation can be controlled via standard Clarify Privilege Classes.

The Contact Alias form can be launched from the standard Clarify *Select Contacts* form, or optionally from the Policies and Customers *New* or *Select* menus.

From this form, a Clarify user or administrator can select and mark contacts as duplicates, and optionally process those duplicates.

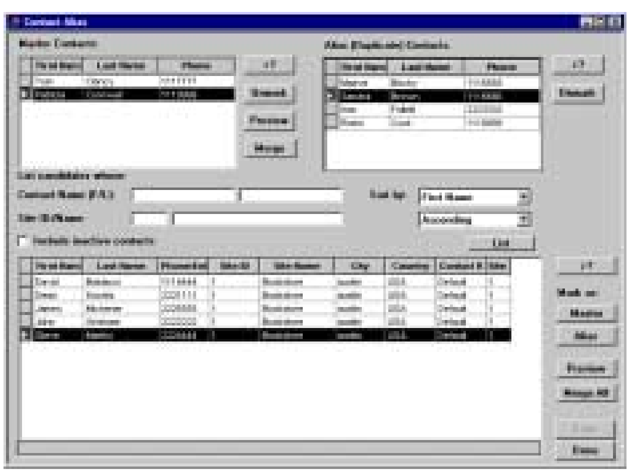
The Contact Alias functionality revolves around the three lists you can see on the form. These lists are:

Candidates List. This is the list of contacts that have not yet been marked as a master or contact. The list of contacts can be filtered by first name, last name, site id, site name and status. Should a contact have roles at multiple sites, all but the contact’s primary site is automatically filtered.

Master Contact List. This is the list of contacts that are currently marked as masters (i.e. there are duplicates of this contact marked in the system).

Alias (Duplicate) Contact List: This is the list of contacts that are aliases (duplicates) of the currently selected master.

The task of identifying and indicating contacts involves choosing a master contact from the candidate list, then choosing aliases (duplicates) of that master from the candidate list.



Contact Alias Form Controls

Candidate – List – Refresh the candidate contact list based on specified filtering criteria.

Candidate - >? – Display contact information for the selected candidate contact.

Candidate - Mark as Master – Mark the selected candidate contact as a master. The contact is removed from the candidate contact list and added to the master contact list.

Candidate - Mark as Alias – Mark the selected candidate contact as an alias (duplicate) of the currently selected master contact. The contact is removed from the candidate contact list and added to the alias contact list.

Master - >? - Display contact information for the selected master contact.

Master – Unmark – Unmark the selected master contact. If there are currently alias contacts marked for this master, a dialog is posted asking for a confirmation. Should the user wish to proceed, the aliases for the master are unmarked as well. The master contact and any aliases are removed from their respective lists and added to the candidate contact list.

Master – Preview – Generate a merge metrics report for the currently selected master contact. The report will be displayed on-line using the text editor tool specified (such as notepad) by the "FC Merge Report Viewer" config_itm. No changes are made to the database.

Master – Merge – Perform a merge for the selected master contact. The aliases contacts for the master contact have their status changed to obsolete, and any objects (such as cases) related to the alias contacts are re-related to the master. After processing, the master contact is returned to the candidate contact list. The alias contacts, now having obsolete status, no longer appear via the Clarify GUI.

> Note: **Merge** is available only after your master/alias selections have been saved to the database via the **Save** button. There is no undo for the merge operation.

Alias - >? - Display contact information for the selected alias contact.

Alias – Unmark - Unmark the selected alias contact. The alias contact is removed from the alias contact list and returned to the candidate contact list.

Preview – Generate a merge metrics report for all master contacts. The report will be displayed on-line using the text editor tool (such as notepad) specified by the "FC Merge Report Viewer" config_itm. No changes are made to the database.

Merge All - Perform a merge for all master contacts. The alias contacts for the master contact have their status changed to obsolete, and any objects (such as cases) related to the alias contacts are re-related to the master. After processing, the master contact is returned to the candidate contact list. The alias contacts, now having obsolete status, no longer appear via the Clarify GUI.

> Note: **Merge All** is available only after your master/alias selections have been saved to the database via the **Save** button. There is no undo for the merge operation.

Save – Save your master/alias selections to the database. The preview and merge options are only available when all master/alias selections are saved to the database.

Swap – Swaps the currently selected master with the currently selected alias. Allows you to change the master for a group of aliases quickly and easily.

Done – Dismiss the Contact Alias form.

Reporting and Auditing

fcMerge provides detailed reporting and auditing of merge processing. This reporting occurs when merge preview or processing occurs from both the Contact Alias form from the Clarify GUI.

Audit Report

When an audit report is generated as the result of a preview or merge, the file name has the following naming convention:

```
<type>.<timestamp>.<username>.log
```

For example, should ‘bill’ perform a merge on November 28, 1997 at 12:34 PM, the result preview audit file will be named:

```
preview.19971128123400.bill.log
```

Should ‘sa’ perform a merge on November 28, 1997 at 12:34 PM, the result merge audit file will be named:

```
merge.19971128123400.sa.log
```

Audit Report Example

The following is an example audit report. It indicates that it was generated by dballard from the Clarify GUI on 11/14/97 and was a Preview. It shows that four alias contacts (Maeve Birchy, Sandra Brown, Robin Cook and Ken Follett) were processed for master Patricia Cornwell and details the object reassignments that would be made should the merge process be run:

```
-------------------------------------------------
RUN BY dballard
RUN FROM GUI
START 11/14/97 11:31:49 AM
MODE preview
-------------------------------------------------
Contact alias 268435495 (Maeve Birchy)
.. Master contact 268435499 (Patricia Cornwell)
.. 1 phone logs to be reassigned.
.... To reassign phone log 268435465
.. 1 cases to be reassigned.
.... To reassign case 268435501 (id=45)
...... To write 'Chng Contact' act_entry
.. 1 contact roles to be deleted.
.... To delete contact role 268435495

Contact alias 268435496 (Sandra Brown)
.. Master contact 268435499 (Patricia Cornwell)
.. 1 contact roles to be deleted.
.... To delete contact role 268435496

Contact alias 268435498 (Robin Cook)
.. Master contact 268435499 (Patricia Cornwell)
.. 1 contact roles to be deleted.
.... To delete contact role 268435498
.. 1 contracts to be reassigned.
.... To reassign contract 268435460

Contact alias 268435502 (Ken Follett)
.. Master contact 268435499 (Patricia Cornwell)
.. 1 contact roles to be deleted.
.... To delete contact role 268435502
-------------------------------------------------
Metrics:

4 contact aliases to be processed.
0 alerts to be processed.
0 commit_logs to be processed.
0 web_users to be processed.
```

```
1 phone_logs to be processed.
0 ctrsitcon_roles to be processed.
0 demand_hdrs to be processed.
1 cases to be processed.
4 contact_roles to be processed.
0 ct_bus_roles to be processed.
0 cr_contact_roles to be processed.
0 case alt contacts to be processed.
0 users to be processed.
1 contracts to be processed.
0 site_parts to be processed.
-------------------------------------------------
END 11/14/97 11:31:50 AM
```

Audit Report Directory

The audit files are written to a directory specified by a system administrator. The administrator must specify a destination directory for the merge audit files. The directory you specify will be utilized by all users of the system. This directory is specified by loading a Clarify "FC Merge Log Directory" config_itm into the database via dataex. An example "merge.dat" file is provided for this purpose. You have two options for the directory you specify:

1) Specify a directory name. If you specify a directory name, it should be a shared directory writable by all users who will perform a preview or merge. Specify a directory name form appropriate for the client platform(s) you are using. You may specify multiple config_itm's, one for each client platform in use. For example, on an NT network with Win clients, you may specify a UNC name such as "\\hostname\share". On a UNIX network, you may specify a shared file system such as "/nfs/host/dir".
2) Specify no directory name. If you specify no directory name, the preview or merge audit files will be written to the working directory of the tool used to perform the merge (either the Clarify application or the merge tool).

It is recommended that you use a shared directory for all audit files.

Audit Report Viewer

When running a preview, the system will use the tool specified by the system administrator to display the report to the user on-line. The tool you specify will be utilized by all users of the system. This directory is specified by loading a Clarify "FC Merge Report Viewer" config_itm into the database via dataex. An example "merge.dat" file is provided for this purpose.

Specify a tool appropriate for the client platform(s) you are using. You may specify multiple config_itm's, one for each client platform in use. For example, on an NT network with Win clients, you may specify "notepad %s". For UNIX clients, you may specify "xterm –e vi %s". In both cases, preview will replace the "%s" with the name of the appropriate audit report file.

Freezing Contacts and Sites

As of version 1.9.3 of **fcMerge**, you have the ability to "freeze" contacts or sites. As of version 3.0 you may freeze addresses and parts. As of version 4.0 you may freeze employees. When an is frozen, you may not select it as an alias. In other words, you are denoting that this object is real and important, and should not ever be able to be merged into another master. A frozen object, however, *can* be used as a master.

The implementation of this feature is simple. After the data schema is applied (from the base installation, or when you upgrade from an earlier version), you must add a checkbox to each of the contact form (#712), the site form (#717), the address form (#706), the employee form (#703), or the part number form (#611).

Note: The **fcMerge** product does not ship with altered version of these forms. This is because these forms are so often customized (and the customization listed below is so simple) that it does not make sense to include the customized forms.

Each of the checkboxes added should have the following properties:

Contact form:

Type:	Checkbox
Caption:	Do not allow as alias
Name:	CHK_NO_ALIAS
Dest Cobj:	contact
Dest Field:	x_no_alias

Site form:

Type:	Checkbox
Caption:	Do not allow as alias
Name:	CHK_NO_ALIAS
Dest Cobj:	location
Dest Field:	x_no_alias

Address form:

Type:	Checkbox
Caption:	Do not allow as alias
Name:	CHK_NO_ALIAS
Dest Cobj:	address
Dest Field:	x_no_alias

Part form:

Type:	Checkbox
Caption:	Do not allow as alias
Name:	CHK_NO_ALIAS
Dest Cobj:	part_mod
Dest Field:	x_no_alias

Employee form:

Type:	Checkbox
Caption:	Do not allow as alias
Name:	CHK_NO_ALIAS
Dest Cobj:	employee
Dest Field:	x_no_alias

Make sure to save the forms, and add them to the proper resource configuration(s).

To use this feature, simply check the box for any contacts or sites that should not be able to be used as aliases. If you do try to then make them an alias, **fcMerge** will refuse to do so, and will pop up a message box to tell you why.

Configuration

- All fcMerge configuration occurs by setting “FC Merge” config_itm’s. A sample merge.dat has been provided.

FC Merge Log Directory

The following is an example of a Merge config_itm for specifying the directory for audit reports. In this case, a shared directory \\commonhost\sharedir has been specified for Windows clients.

```
OBJECT  TYPE="config_itm", NAME="LD_0"
 UNIQUE_FIELD = name, value_type
  FIELDS
   name ="FC Merge Log Directory";
```

```
    value_type=0;
    str_value="\\\\commonhost\sharedir";
    scope=0;
    description="Log Dir for Win Clients";
  END_FIELDS
  RELATIONS
  END_RELATIONS
END_OBJECT  NAME="LD_0"
```

name	Must be "FC Merge Log Directory"
value_type	0 = Dir for win clients 1 = Dir for unix clients
str_value	Shared directory name appropriate for client platform. For Win clients, to indicate a double-backslash, you must escape both, ie "\\\\". Finally, should you specify an empty string "", the Clarify working directory will be used (not recommended).
scope	Must be 0.
description	Any description.

FC Merge Report Viewer

The following is an example of a fcMerge config_itm for specifying an audit report viewer. In this case, "notepad" has been specified for Windows clients.

```
OBJECT  TYPE="config_itm", NAME="LV_0"
 UNIQUE_FIELD = name, value_type
  FIELDS
    name ="FC Merge Report Viewer";
    value_type=0;
    str_value="notepad %s";
    scope=0;
    description="Viewer for Win Clients";
  END_FIELDS
  RELATIONS
  END_RELATIONS
END_OBJECT  NAME="LV_0"
```

name	Must be "FC Merge Report Viewer"
value_type	0 = Dir for win clients 1 = Dir for unix clients
str_value	String for launching a text file editor or viewer appropriate for platform. You must specify a %s in the string which will be substituted at run-time for the name of the audit report file. For Win, "notepad %s" is suitable. For UNIX, "xterm –e vi %s" is appropriate.
scope	Must be 0.
description	Any description.

FC Merge Move Site Too

The following is an example of a fcMerge config_itm for specifying the behavior of contact merge when it merges any related cases.

```
OBJECT TYPE="config_itm", NAME="config_itm_merge_site_too"
 UNIQUE_FIELD = name, value_type
```

```
  FIELDS
   name ="FC Merge Move Site Too";
   i_value=1;
   description="When we contact merge, and change case contact, should we change to contact's primary
site?";
  END_FIELDS
  RELATIONS
  END_RELATIONS
END_OBJECT  NAME="config_itm_merge_site_too"
```

name	Must be "`FC Merge Move Site Too`"
i_value	Value for action. If set to 0, a contact merge will not affect the site for any moved cases. If set to 1, then the site for all merged cases will be moved to the new master's primary site.
description	Any description.

FC Merge Add Activity

The following is an example of a fcMerge config_itm for specifying the behavior of contact merge when it merges any related cases.

```
OBJECT TYPE="config_itm", NAME="config_itm_merge_add_act_entry"
 UNIQUE_FIELD = name, value_type
  FIELDS
   name ="FC Merge Add Activity"
   i_value=1;
   description="When we contact/site merge, should we add new 'Merge activity log?";
  END_FIELDS
  RELATIONS
  END_RELATIONS
END_OBJECT  NAME="config_itm_merge_add_act_entry"
```

name	Must be "`FC Merge Add Activity`"
i_value	Value for action. If set to 1, then a new activity log entry will be added to the master for each merged alias. If set to 0, no new activity logs will be added.
description	Any description.

FC Merge Change Instance

The following is an example of a fcMerge config_itm for specifying the behavior of contact merge when it merges site_parts for a merged part revision.

```
OBJECT TYPE="config_itm", NAME="config_itm_merge_change_instance_name"
 UNIQUE_FIELD = name, value_type
  FIELDS
   name ="FC Merge Change Instance";
   i_value=1;
   description="When we part_num merge, should we change the instance name on site_parts";
  END_FIELDS
  RELATIONS
  END_RELATIONS
END_OBJECT  NAME="config_itm_merge_change_instance_name"
```

name	Must be "`FC Merge Add Activity`"
i_value	Value for action. If set to 1, then the instance_name of any site_parts related to the alias will be changed to the new part's

	part and revision. If set to 0, nothing is changed in the part revision.
description	Any description.

Loading the config_itm's

Use the Clarify dataex tool to load your config_itm file. Assuming it is merge.dat:

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

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

Managing Access

Access to the operation of fcMerge can be controlled by standard Clarify privilege class. You can allow or deny the following operations by privilege class:

- Save of master/alias selection to the database
- Preview of merge process
- Invocation of merge process

By default, form command "buttons" are enabled for a privilege class. A task of defining a privilege class is specifying which "buttons" should be disabled for a particular class. The fcMerge button names used in this task are:

Button Name	Description
cmdSave	Saves the current master/alias configuration to the database.
cmdPreview	Runs a preview of the merge process for the selected master.
cmdPreviewAll	Runs a preview of the merge process for all masters.
cmdMerge	Invokes the merge process for the selected master.
cmdMergeAll	Invokes the merge process for all masters.

An Example

As an example, let us deny "merge" access to the "CSR" privilege class. To do this, log in as a System Administrator and:

1. Select Apps->Policies and Customers
2. Setup->Privilege Classes
3. Select "CSR" and click Open
4. Click "Buttons". The "Disabled Buttons" form posts.
5. List buttons whose "Window Name" starts with "FORM_1920" and click "List"
6. Scroll list to locate "Button Name" of "cmdMerge"
7. Select "cmdMerge" and click "Copy Selection". The "cmdMerge" button is copied to the "Buttons disabled" list.
8. Scroll list to locate "Button Name" of "cmdMergeAll"
9. Select "cmdMergeAll" and click "Copy Selection". The "cmdMergeAll" button is copied to the "Buttons disabled" list.
10. Click "Done". The "Disabled Buttons" form dismisses.

11. Click “Replace”.

The next time a user with “CSR” privilege class logs in or performs an “Update Desktop”, the Merge and Merge All buttons will be disabled on the Contact Alias form.

Note: Should you create multiple “user versions” of the 1920 (Contact Alias) form or 1921 (Site Alias) form, the buttons will be displayed for each version of the form. When selecting a button to disable such as “cmdMerge”, choose any user version of the form. The “cmdMerge” button will be disabled regardless of the specific user version of the form specified in the user’s resource config.

Batch Merging

This version of **fcMerge** ships with shell scripts (contact_merge.sh, site_merge.sh, employee_merge.sh) that can be used to automate and batch the merge or preview process. You can edit these files if you wish to perform the same operation for address, or part merge.

Each of the shell scripts uses the *merge.ini* file to specify execution parameters. See the documentation in the file for the specified parameters to use.

The scripts are executed with the following command lines:

contact_merge.sh –i merge.ini [-merge]
site_merge.sh –i merge.ini [-merge]
employee_merge.sh –i merge.ini [-merge]

Note: The **–**merge option is used to perform a merge. Without it, the script will only perform a preview operation.

Note: On Windows (using the MKS Toolkit), place “sh “ in front of the above commands.

Example: sh site_merge.sh –i merge.ini -merge

Batch Marking

Usually, most customers will *mark* the masters and aliases in the provided GUIs. Marking is the process whereby masters and aliases are associated together.

In some cases, however, customers wish to mark the masters and aliases in a batch mode. Then, they can use the GUIs to view the masters and aliases, and correct any mistakes found. Finally, they can (either manually or via the shell scripts) merge the marked contacts or sites or employees.

Batch marking is a bit tricky. The main problem is in knowing *which* contacts (or sites or employees) are masters and are aliases. This logic is so customer-specific that it is impossible to write a general solution to this problem.

What can be done (and is provided with this package) is to provide a robust sample program that illustrates the process of batch marking. Then, a customer can replace the selection logic within the program (to match their specific needs) to easily construct a batch marking program.

The sample provided is named *duns_inter.cbs*. This is an actual batch marking program used by a First Choice customer to help merge duplicate sites. The basic idea is that they have added a new field, x_duns_no to the site table. Each day, an ASCII file is received (from an outside service) that lists the site Ids, and the Duns numbers for those sites. The interface loads the x_duns_no field, and determines if more than one site has the same number. If they do, then they are the same site, and one (the first one) is marked as the master, and all subsequent ones are marked as aliases.

In modifying this program for your needs, the only section that should change is that the selection logic (of which contacts/sites are masters/aliases). The rest of the program (including the section that sets the fields/relations for master/alias) will stay the same.

Please review the commented source code of duns_inter.cbs for more information.

Requirements

This version of **fcMerge** requires the following:

Clarify Version: 4.5 or later

Clarify Tools: Data Dictionary Compiler (ddcomp)
User Interface Editor (uieditor)
Clear Basic Compiler (cbex)
Data Exchange (dataex)
Other Tools: For NT, your favorite unzip tool

Packaging

fcMerge is shipped to you as a zip file.

Installation Tree

It is recommended that you uncompress the zip file containing fcMerge into an fchoice subdirectory created at the top of the Clarify install tree. For example on NT, should your Clarify server install tree be “c:\clarify”, then:

1. Switch to “c:\clarify” directory.
2. Create an “fchoice” directory, if necessary.
3. Switch to “fchoice” directory.
4. Switch into “c:\clarify\fchoice” directory.
5. Create a “merge” directory, if necessary.
6. Unzip the file in the “merge” directory.

The following files are provided with this product:

File Name	Purpose
merge_user.pdf	This document
/code/1920.cbs	ClearBasic code for Contact Alias form
/code/1921.cbs	ClearBasic code for Site Alias form
/code/1922.cbs	ClearBasic code for Address alias
/code/1927.cbs	ClearBasic code for Employee alias
/code/1923.cbs	ClearBasic code for part alias
/code/1926.cbs	ClearBasic code for Address details
/code/610.cbs	ClearBasic code for Select Parts
/code/705.cbs	ClearBasic code for Select Addresses
/code/711.cbs	ClearBasic code for Select Contacts
/code/716.cbs	ClearBasic code for Select Sites
/code/merge.cbs	ClearBasic code for fcMerge
/code/site_merge.cbs	ClearBasic code for Site Merge
/code/employee_merge.cbs	ClearBasic code for Employee Merge
/code/address_merge.cbs	ClearBasic code for Address Merge
/code/part_merge.cbs	ClearBasic code for Part Merge
/code/merge_menu.cbs	ClearBasic code for adding menus to Policies and Customers menu bar.
/code/merge.dir	Directives file for CB compile.
/forms/1920.dat	Contact Alias form.
/forms/1921.dat	Site Alias form
/forms/1922.dat	Address alias form

/forms/1927.dat	Employee alias form
/forms/1923.dat	Part alias form
/forms/1926.dat	Address details form
/files/merge.dat	Configuration items
/schema/merge.sch	Schema changes
/scripts/contact_merge.sh	Script to merge contacts in a batch run
/scripts/site_merge.sh	Script to merge sites in a batch run
/scripts/employee_merge.sh	Script to merge employees in a batch run
/scripts/merge.ini	Initialization file for .sh files
/sample/duns_inter.cbs	Sample interface to demonstrate batch marking of masters/aliases
/sample/duns.in	Input file for sample batch marking interface

Installation

Now that you have unpacked the install package, you will be performing the following tasks during the installation of fcMerge:

1. Updating the Clarify Schema
2. Importing new and updated forms
3. Setting up resource configurations
4. Compiling ClearBasic code
5. Loading the Configuration File

Note: It is highly recommended that you first install the fcMerge on a test system and become familiar with its operation before installing it on a production system.

Updating the Clarify Schema

This section details the steps required to update the Clarify database schema to work with fcMerge.
The */schema/merge.sch* file details all of the necessary schema changes.

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

fcMerge requires that new schema be added to the database. You will need the Clarify Data Dictionary Editor, a text editor such as notepad or vi and the merge.sch file provided in the **fcMerge** install tree.

To make the required schema changes:

1. Start the Data Dictionary Editor.
2. Choose *Save To File* from the *File* menu.
3. Type **merge.new** and press the *OK* button.
4. With a text editor, edit the *merge.new* file (it will be stored in the DD Editor directory) and make the changes listed in the *merge.sch* file provided in the fcMerge install tree.
5. Choose *Apply Changes* from the *Actions* menu.
6. Select the *merge.new* file.
7. When asked, press *Proceed*.

8. Review the results presented. If any errors, 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

User Interface Modifications

You must always import (see below) the 1920.dat, 1921.dat, 1922.dat, 1923.dat, 1927.dat, and 1926.dat files. If you wish to have the merge forms displayed from the "Select" forms (Select sites (716), Select Contacts (711), Select Address (705), Select Employees (702), Select Part (610)), you may create these forms manually, and add a button to call on the merge form (with the same code as is found in the merge_menu.cbs file). To do this, you must perform the following:

1. Start the UI Editor.
2. Open the baseline (default) version of form 711.
3. Add a new command button with the following attributes:
   - Name: cmdAliases
   - Type: Command Button
   - Caption: Aliases…
4. Save the form with user version **merge_1.0**
5. Perform steps 2-4 for form 716, 705, 702, and 610

The form(s) provided with this product must be imported into the database with the data exchange tool. These form(s) 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 form file (.dat extension)

Note: The slash between *<path>* and *dataex* should be a forward slash for UNIX systems.

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

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.

Note: If you are using the optional "Freeze" functionality, you must add a checkbox to both the contact form (712) the site form (717), the address form (706), the employee form (703), and the part_num form (611). Please see the "Freezing Contacts and Sites" section above for details on the check boxes.

After all of the form files have been imported successfully, the new forms must be added to the proper resource configuration(s) in UI Editor. You must decide which user or groups of users should see the new versions of the imported forms.

To add the forms to the resource configuration(s):

1. Start the User Interface Editor.
2. Choose *Resource Configs…* from the *Select* menu.
3. Press the *List* button.
4. Select the resource configuration you want to modify and press the *Open* button.
5. Type **Merge** in the text box to the right of the starts with dropdown list and press the *List* button.
6. Select the **1920 merge_1.0**, **1921 merge_1.0, 1922 merge_1.0, 1923 merge_1.0, 1927 merge_1.0,** and **1926 merge_1.0** 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. Press the *Replace* button. After the save is completed, press the *Done* button.
9. Repeat steps 4-8 for any other resource configurations you wish to modify.
10. Exit the User Interface Editor.

Clear Basic Code Compilation

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

When Clarify starts up it runs the routine 'initialize_app' (if one exists). Only one routine with this name will be run on client start-up. The module <module> contains an 'initialize_app' routine for this application. If some other custom application has a version of this routine, only one of these versions will execute. It is REQUIRED that you merge any existing version of 'initialize_app' into the version in module <module>. Then, replace the original version of 'initialize_app' with the merged version and recompile it. This will prevent the original version from overwriting the merged version if you recompile your original code, and it will ensure that all your custom applications will still be executed.

Not doing this as specified may cause this application or previous customizations to function improperly.

> **Note**: You may have to edit the merge.dir file to change the Clarify version of the forms, depending on the version of Clarify you are using, and if you use the presupplied forms, or not. Make sure that the Clarify version of the form in your resource configuration matches the one in the directives file.

To compile the code modules, make sure that the code modules (*.cbs) and the directives file (merge.dir) are located in the same directory. Enter the following command to compile the code:

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

Where:

<path> is the path to the cbex program.

When the compilation is completed, you should 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 (there shouldn't be), check the above instructions to make sure that your compilation environment is set up correctly.

> **Note:** The Clear Basic compiler requires that the user who performs the compilation be a member of a resource configuration that includes the forms compiled against. Example: If user *sa* compiles the CB code, he must be a member of a resource configuration that holds the new forms.

> **Note**: The slash between *<path>* and *cbex* should be a forward slash for UNIX systems.

After you have finished compiling the code, you are ready to use the customizations.

Note: Any users who are currently running on the system should either stop Clarify and start it up again, or choose the *Update Desktop* menu item on the *Desktop* menu.

Limitations

There are no known issues with fcMerge.

Performance

You should find that the following optional indexes will optimize the performance for fcMerge:

Table	Column(s)	Index Name	Unique?
contact	alias2contact	alias_c_index	No
site	alias2site	alias_s_index	No
address	alias2address	alias_a_index	No
part_num	alias2part_num	alias_p_index	No
employee	alias2employee	alias_e_index	No

In addition, fcMerge can be VERY slow on a Clarify system that contains a significant amount of data. The reason for this is that the merge (or preview) process must search for objects to be moved from alias to master in the merge process.

For example, suppose that the case table contains millions of cases. When a contact (or site) is merged, the system must check every case to see if it is associated with the alias contact/site. If the relation column is not indexed, this will cause a table scan.

As an example, one user of site merge reported times of up to 1 hour to process a single alias site (due to the large amount of data in their Clarify system). Adding the proper indexes reduced that time to less than 6 seconds for the same merge.

In a single merge, there are a large number of tables that must be searched. But not all of them will contain a large number of rows, and thus not all of them need to be indexed.

The following tables list the tables (and columns) that will be used to perform the merge operation. If there are a large number (over 1000) rows in the table listed, then you should consider adding an index on the column listed to improve performance. You may use SQL (or the Clarify dbtune script) to add the index.

Contact Merge Columns:

Table Name	Column Name	Clarify Version
alert	alert2contact	All
commit_log	commit_cust2contact	All
web_user	web_user2contact	All
phone_log	phone_custmr2contact	All
demand_hdr	attention2contact	All
case	case_reporter2contact	All
contact_role	contact_role2contact	All
ct_bus_role	ct_bus_role2contact	All
case	alt_contact2contact	All
contact	caller2user	All

person	person2contact	All
employee	employee2contact	All
case	case_reporter2site	All
case	de_product2site_part	All
case	case_node2site_part	All
case	case_prod2site_part	All
case	entitlement2contract	All
contact_role	contact_role2site	All
site	cust_primaddr2address	All
mtm_site_part22_contact6	contact2site_part	All
mtm_site_part22_contact6	site_part2contact	All
mtm_contact2_contract1	contract2contact	All
mtm_contact2_contract1	caller2contract	All
contract	primary2contact	5.0+
act_entry	act_entry2contact	5.0+
email_log	cont_email2contact	5.0+
task	task2contact	5.0+
quick_quote	q_quote2contact	5.0+
con_opp_role	opp_role2contact	5.0+
con_csc_role	csc_role2contact	5.0+
lst_con_role	list_role2contact	5.0+
mtm_contact17_campaign6	contact2campaign	5.0+
mtm_contact17_campaign6	campaign2contact	5.0+
scr_run	scr_run2contact	6.0+
interact	interact2contact	6.0+
con_lsc_role	con_lsc2lead_source	7.0+
exc_con_role	exc_con_role2exchange	7.0+
doc_inst	doc_inst2contact	8.0+
E_addr	eaddr2contact	9.0+
R_rqst	r_rqst2contact	9.0+
R_rqst	R_contact2contact	9.0+
con_pym_role	cn_pm_role2contact	10.0+
con_rol_itm	parent2con_rol_itm	11.0+
con_rol_itm	child2con_rol_itm	11.0+
preference	contact2preference	11.0+
user_touch	contact2user_touch	11.0+
con_fin_accnt_role	contact2con_fin_accnt_fole	11.5+
con_blg_argmnt_role	contact2con_blg_argmnt_role	11.5+
con_sp_role	contact2con_sp_role	11.5+
con_pc_role	contact2con_pc_role	11.5+
con_ppb_role	contact2con_ppb_role	12.0+
order_action	contact2order_action	12.0+
fr_agr_con_role	contact2fr_agr_con_role	12.0+
intrxn	contact2intrxn	12.0+
proc_inst	contact2proc_inst	12.0+

Site Merge Columns:

Table Name	Column Name	Clarify Version
part_vend	part_vend2site	All
inv_locatn	inv_locatn2site	All
phone	phone2site	All
case	case_distr2site	All
doc_inst	doc_inst2site	All
ship_parts	carrier2vendor	All
site	child_site2site	All
trans_record	trans_record2site	All
productbin	direct_prdbin2site	All
act_entry	act_entry2site	All
productbin	prdbin2site	All
acd_call	acd_call_site2site	All
demand_hdr	open_reqst2site	All
demand_hdr	bill_reqst2site	All
site	support_office2site	All
employee	supp_person_off2site	All
site_part	all_site_part2site	All
site_part	site_part2site	All
case	case_reporter2site	All
contact_role	contact_role2site	All
inv_role	inv_role2site	All
mtm_site5_contract0	cust_loc2contract	All
mtm_site5_contract0	contract2site	All
alert	alert2site	4.5+
E_addr	eaddr2site	4.5+
per_site_role	per_site_role2site	4.5+
C_site_role	c_site_role2site	4.5+
sit_prt_role	sit_prt_role2site	4.5+
site_addr_role	site_addr_role2site	4.5+
bus_site_role	bus_site_role2site	4.5+
contr_schedule	bill_to2site	5.0+
quick_quote	q_quote2site	5.0+
contr_itm	quoted_at2site	5.0+
contr_itm	covered_by2site	5.0+
sit_csc_role	csc_role2site	5.0+
contr_schedule	ship_to2site	5.0+
exchange	partner2site	5.0+
scr_run	scr_run2site	6.0+
exc_sit_role	exc_sit_role2exchange	7.0+
part_num	part_num2site	7.0+
vendor_part	vendor_part2site	7.0+
shp_sit_role	shp_role2site	9.0+
bus_org	primary_site2bus_org	11.5+
order_action	site2order_action	12.0+
order_shipping	site2order_shipping	12.0+
site_ba_role	site2site_ba_role	12.0+
blg_argmnt	primary_site2blg_argmnt	12.0+

Address Merge Columns:

Table Name	Column Name	Clarify Version
site	part_vend2site	All
site	bill_addr2site	All
site	ship_addr2site	All
case	address2case	All
site_addr_role	addr2site_addr_role	All
bus_org	bus_primary2bus_org	All
bus_addr_role	address2bus_addr_role	All
per_addr_role	address2per_addr_role	All
contr_schedule	ship_addr2contr_schedule	All
contr_schedule	bill_addr2contr_schedule	All
contr_itm	bill_addr2contr_itm	8.0+
address_extn	address2address_extn	10.1+
addr_ba_role	address2addr_ba_role	11.0+

Part Merge Columns:

Table Name	Column Name	Clarify Version
backorder	part_info2backorder	All
bug	part_info2bug	All
case	case_cat_part2case	All
case	de_cat_part2case	All
catalog	part_info2catalog	All
contr_itm	part_info2contr_itm	All
demand_dtl	orig_part2demand_dtl	All
demand_dtl	part_info2demand_dtl	All
eco_dtl	part_info2eco_dtl	All
eco_mod_role	applies2eco_mod_role	All
eco_mod_role	result2eco_mod_role	All
entitlement	service2entitlement	All
mod_level	part_num_incl2part_num	All
mod_level	incl_part_num2part_num	All
mod_level	n_part_info2part_auth	All
mod_level	n_part_info2part_inst	All
part_qty	part_incl2part_qty	All
part_role	for_mod2part_role	All
part_used	inst_part_info2part_used	All
part_used	rem_part_info2part_used	All
price_qty	priced_part2price_qty	All
price_qty	context_part2price_qty	All
price_role	part_info2price_role	All
prog_logic	part_info2prog_logic	All
qq_line_item	part2qq_line_item	All
recv_parts	part_info2recv_parts	All
recv_prob	part_info2recv_prob	All
site_part	part_info2site_part	All
vendor_part	part_info2vendor_part	All
interact	part_info2interact	6.0+
mtl_log	mod_level2mtl_log	6.0+
doc_inst	part_attach2doc_inst	7.0+

lit_req_itm	part_info2lit_req_itm	7.0+
rpr_inst_qty	part_info2rpr_inst_qty	7.0+
bus_opp_role	mod_level2bus_opp_role	11.0+
lead	mod_level2lead	11.0+
flex_tmpl	mod_level2flex_tmpl	12.0+
offering	mod_level2offering	12.0+
part_action	mod_level2part_action	12.0+
proc_inst	mod_level2proc_inst	12.0+
process	mod_level2process	12.0+

Employee Merge Columns:

Any table that has a MTO or OTOP relation to table_employee or table_user**.** Use the First Choice BOLT product, your Data Dictionary, or SQL to determine these relations.

For example:
select m.type_name, r.rel_name
from adp_sch_rel_info r,
adp_tbl_name_map m
where
target_name in ('user','employee')
and rel_type in (2,3,10,11)
and m.type_id = r.type_id

Year 2000

fcMerge was designed to be Year 2000 compliant. All date/time comparisons are performed using database-supplied routines. All date/time storage is in the Clarify database using date/time fields. No Year 2000 issues are expected.

Internationalization

Future versions of **fcMerge** are expected to be localized. All strings presented will be read from string tables, allowing the **fcMerge** process to be performed in different languages for different end-users at the same time in the same database. If this is a requirement for your organization, please contact First Choice Software so that this can be scheduled.

How to Contact Us

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

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