



**Mail Merge For Clarify™
Version 1.0.1**

Workflow Series™

Installation and User's Guide

Trademarks

First Choice Software, Inc. (FCS) develops and supplies 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 Amdocs.

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

Overview	4
General Overview	4
Before You Upgrade to a New Version... ..	4
What's New in Version 1.0.1	5
What's New in Version 1.0	5
Mail Merge For Clarify Forms	6
Profile Administration Form	6
Path Editor Form	7
Mail Merge Documents	8
Standard Variables	8
Multiple Variables	9
Fomulas	10
Special Variables	10
Date Formatting	10
Attachments	11
Performing Mail Merge	11
A Sample Of How To Use Merge	12
Implementation	13
Strings	13
Forms	13
Menu Items	13
Configuration Items	13
Auto-numbering Schemes	14
Requirements	14
Packaging	14
Files Included in MM	14
Installation	15
Installation Tree	16
Updating the Schema	16
Backup the Database	16
Import the New Schema	16
Import the Data Files	16
Import the User Interface Forms	17
Set up Resource Configurations	18
Compile ClearBasic Code	18
Global Clear Basic Code	19
Environment Variables	19
Limitations	19
Year 2000	19
Internationalization	19
How to Contact Us	20
API Reference	21
mail_merge	22

Overview

General Overview

It is a fairly common requirement that Clarify users would like to perform “mail merge” operations. *Mail Merge* is the process of taking a generic template (in this case, in Microsoft Word), and replacing certain generic *variables* (or spots to be filled in) with data from a contact, case, quote, etc. In effect, this personalizes the document with the required data. This product, *Mail Merge for Clarify* (or **MM**), makes this process simple, painless, and quick.

For example, it might be useful to have a mail merge that when a button is pressed on the case form that a form letter is filled in with the customer’s name, address, case title, owner, etc. Or, on a quote form it might be helpful to have a mail merge automatically build a quote, including header information, and a table of line items, prices, and discounts, all read from the database.

Many Clarify customers have written customized code to fit this mail merge need. While this works, it is unsatisfying, because it is specific to the one mail merge operation being performed. When another mail merge is required, more custom coding must be performed. Any time an existing mail merge needs to be tailored, more customization is required.

What customers have requested is the ability to have a *generic* mail merge. In other words, an easy to use mechanism that would allow for administrators to quickly add new mail merge items, simply by creating a template in Microsoft Word, by manipulating a simple GUI, and by adding a button (or menu, or dropdown list) to start the mail merge process.

The first component of this is the ability, via a GUI, to specify mail merge *profiles*. These profiles include a root (base) object, and a set of variables that are data items from that object and/or from related objects. Any time a new variable is needed for a mail merge, the administrator only has to add it in the GUI.

For example, a profile might be driven off of the `case` table. There might be variables defined for the case ID number, the case title, the contact’s first name, last name, and phone number. The case type, priority, and severity may all be used as well. In addition, the case’s activity history might be a variable, so it can be displayed.

The second component of a generic mail merge is the creation of a Word template. This *Mail Merge For Clarify* defines a very simple syntax for **variables** that you insert into your Word document. When the mail merge process happens, each variable that is found is replaced with the proper data from the root object (or related object), as determined by the profile variables.

In the example above, a variable called `ID_NUMBER` might be defined in the profile. If the mail merge process found that variable in the Word document, it would know to look at the variable defined in the profile. In this case, the variable simply uses the `id_number` field of the case. The value for the `id_number` for the case is then placed in the Word document, replacing the variable name.

The final element of a generic mail merge is the ability to call on the mail merge process from any form in Clarify. This is accomplished by providing the administrator, in the product, a new API called *mail_merge*. This is a ClearBasic API that can be placed in the code for any form (Clarify-defined, or user-defined) in the thick client GUI. The API is simple. It requires a root object `objid` (which object is the mail merge based on), a profile name (to know the variables to substitute), a document template (the Word document to merge), and an output file name (the name when complete). It also has a variety of other flags to determine specific behavior. The API is documented in detail later in this manual.

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 (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 1.0.1

If you have a variable that finds a field (via the path), but has no data, it is now blank. Before it said "NOT FOUND". Also, the order of the variable resolution has changed. Now it does NON-table variables (regular ones) first. Then it resolves the tables. This way, if a table contains a regular variable (maybe a discount % for a site), it is filled in before the table is resolved, and recalculated. Finally, the tab order on form 4260 has been fixed.

To apply this patch, you must perform the following steps:

- 1) Import the new versions of the following form files into the database with dataex:
 - a. mm4260.dat
- 2) Compile the new versions of the following code files using the supplied directives file:
 - a. mail_merge.cbs

What's New in Version 1.0

This is the initial version of **MM**.

Mail Merge For Clarify Forms

This section describes the **MM** forms provided with the product. The first forms provide a method to set up and maintain the mail merge profiles.

Profile Administration Form

Mail Merge Profile Administration

Profiles:

Profile Name	Description	Root Object
Case Vars		case

Add
Replace
Delete

Clear

Profile Name/Object: Case Vars case

Description:

Variables

Variable Name	Variable Path
ID_Number	id_number
Site_Name	case_reporter2site:name
Address1	case_reporter2site:cust_primaddr2address:address
City	case_reporter2site:cust_primaddr2address:city
State	case_reporter2site:cust_primaddr2address:state
Zipcode	case_reporter2site:cust_primaddr2address:zipcode
First_Name	case_reporter2contact:first_name
Last_Name	case_reporter2contact:last_name
Phone	case_reporter2contact:phone

Add
Replace
Delete

Up
Down

Variable Name: Address1

Variable Path: case_reporter2site:cust_primaddr2address:address

Done

This form is used to create and modify the profiles and variables for mail merge. The top half of the form is used for maintaining profiles, while the lower half is used to maintain the variables for a specific profile.

The top grid shows all of the existing profiles. Each profile contains three pieces of information:

- 1) Profile name. This must be unique
- 2) Root object. This is a table name or view in Clarify (including user-defined ones) that is the **root** of the profile
- 3) An optional description of the profile.

To create a new profile, simply fill in the fields below the top grid and press the *Add* button. To modify a profile, single click on the desired profile, change the information below the grid, and press the *Replace* button near the top of the form. Similarly, to delete a profile, select it, and press the top *Delete* button. You will be asked to confirm the delete.

The *Clear* button clears out the profile data so you can enter a new profile easily and quickly.

Once you have created one or more profiles, you must populate them with variables. To do this, click on a profile in the top grid to select it. Then you can add variables to the profile. Each variable consists of two pieces of information:

- 1) A variable name. This is unique *for a given profile*
- 2) A **path** to the data. This is either A) a field name in the root object, or B) a path to a field name in a related object. A path is made up of a series of relations (starting with the root object), separated by colons, ended with a field name. Paths are created with the Path Editor (described below), and are very simple to build

To add a variable to a profile, enter a variable name, and press the *Variable Path* button to build the path. Then, press the lower *Add* button to add it to the profile. To modify a variable, select it in the lower grid, modify the fields below the grid, and press the *Replace* button. To delete a variable, select it and press the *Delete* button. The *Up* and *Down* buttons are provided to change the order of variables within a profile, although this is for display purposes only, and does not affect any functionality.

When you have finished modifying profiles, press the *Done* button to exit the form.

Path Editor Form

To create or modify a variable the administrator must select a *path* for a database column. Paths in **MM** are exactly like paths in base Clarify – the only difference is a new form, the Path Editor, which makes entering or modifying paths very simple and convenient.

A *path* is a description of how to arrive at a field, starting with a base table. For example, if one is setting up case variables, it might be useful to have a path set up to the *id_number* field of the case. Since that field is defined in the case table itself, the path is simply the name of the field “id_number”.

Suppose however that we want a path to the state for the primary address of the site of the case. That path would be: “case_reporter2site:cust_primaddr2address:address2state_prov:name”. Each relation traverses from one table to the next, and each item is separated from the previous one by a colon. The final entry in a path is **always** a field in the final destination table.

The path editor form makes entering these paths very simple. Previously, administrators had to type in paths (in either SQL or dataex). That process was very slow, and error-prone. Using the path editor, typing errors are not possible.

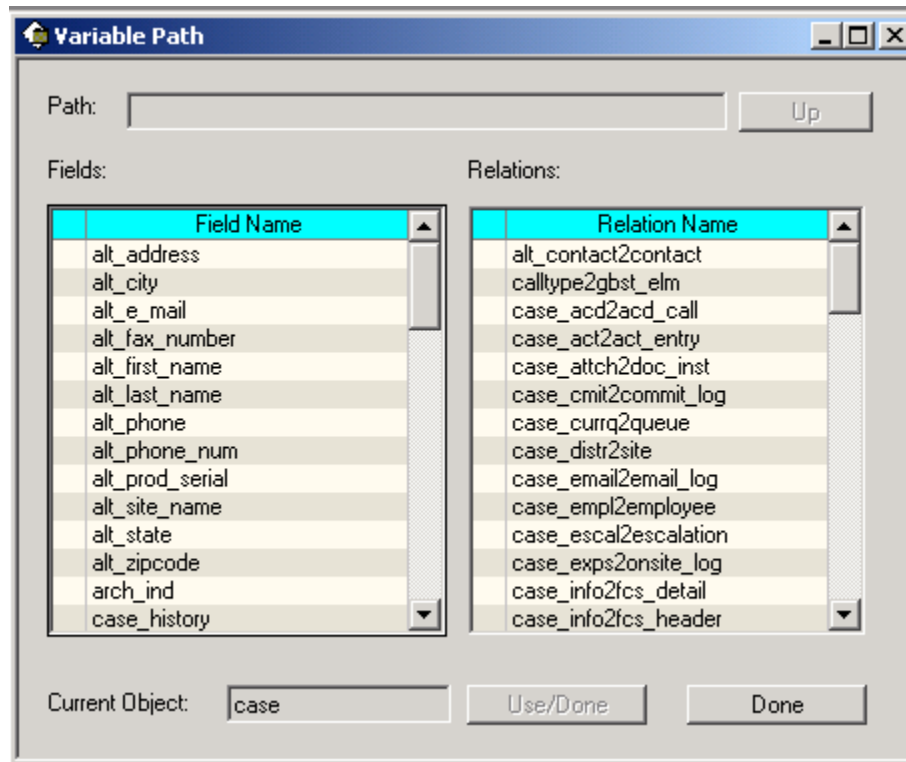
The path editor form has several parts. At the top of the form is the current path. As you build up the path, the current path is displayed in this read-only text box.

At the bottom of the form is another read-only text box that displays the current table. When you start (for example) from the case table, the word “case” would be displayed in that text box. If you traverse to another table (by selecting a relation), the new (current) table would be displayed in the box.

In the middle of the form are two grids. The left-hand grid displays all of the fields in the current table. The other grid displays relations from the current table to other tables.

To select an item from either grid, simply double click it. When the path is complete, press the *Use/Done* button (or double-click a field). If you want to leave the form without completing the

path, simply press the *Done* button. If you are entering a path, and make a mistake by traversing an incorrect relation, you may press the *Up* button to go back one level. Pressing that button will remove the last relation from the path you are building. It will change the current table (and the fields and relations displayed) back to the previous table.



So, to use the two examples above, to enter the first path, simply double click on the “title” field. That will enter a path of “title”. The other path is a little more complex.

To traverse from the case table to the site table, double click on the “case_reporter2site” relation. The current object is now “site”, and the fields and relations are those from the site table. In addition, the path is now “case_reporter2site”. Traverse to the address table by double clicking the “cust_primaddr2address” relation. The path is built up, and now you are presented with fields and relations from the address table.

The form shown above is one step further where the “address2state_prov” relation has been selected. To complete the path, double click on the “name” field.

Mail Merge Documents

The core of the **MM** product are the Microsoft word document “templates” that you use as a basis for merging. These template documents can be normal .doc files – they do not need to be Word template files (.dot files).

The basic idea is that in any location that you want to show data from the Clarify database you will place a *variable* in the document. The format of variables is described below. When **MM** performs the mail merge operation, each variable will be replaced with the appropriate data from the database.

Standard Variables

Most of the variables in your Word documents will be so-called standard variables. Standard variables represent a **SINGLE** piece of data taken from the database. For example, for a case, the case title, ID number, contact first name, site name, primary site address are all standard variables.

If you want to show data that has multiple values (for example, the activity log for a case, which might have 30 rows, or the line items for a quote), you need to use so-called "multiple variables". For more details on multiple variables, please see below.

To place a standard variable in your document, simply enclose the variable name in curly braces "{" and "}". For example, the following are legal variables:

{ID_NUMBER}

{CASE TITLE}

{FirstName}

You may place these variables anywhere you would like in your document, including in the header and footer sections. **MM** will replace all of them with the appropriate value.

From time to time you will make mistakes with your variable names. If **MM** cannot locate a variable name in the provided profile it will replace the variable with a pre-defined string. By default that string is set to "NOT FOUND!!!". However, you may edit the *mail_merge.cbs* file, and edit the constant near the top. If you do change the value in this string you must recompile *mail_merge.cbs* into your database with the provided directives file.

Note: Variable names may have spaces in them. You do not have to use the underscore character.

Note: Variable names are **NOT** case-sensitive. You may upper/lower case the variables as you see fit in both the GUI and in the Word document. If they match, case-insensitively, they will be used.

Multiple Variables

From time to time you will need to display/merge data in your document where the variable actually represents multiple pieces of data. As mentioned above, two common examples of that are:

- 1) Showing all of the line items for a given quote header
- 2) Showing the activity log history (each item) for a queueable object

You can easily show these multiple items using the concept of an *multiple* variable. Multiple variables are defined as are any others in the provided profile GUI. However, in defining them you will choose one (or more) relations that are actually OTM relations in Clarify. These relations will (possibly) traverse to multiple objects.

For example, in the 2nd example above, you might follow the case_act2act_entry relation. This relation will cause you, at merge time, to receive back potentially a great many items of data.

Multiple variables **MUST** be included in Word tables. If they are not located in tables, **MM** will not properly process them. This is not a very difficult restriction, however, because you typically want to show this type of data in a table, and you can always have a table not display the borders. In this latter case, the table simply looks like a list of items.

Multiple variables are surrounded by the pipe ("|") character. To show multiple rows of data for a variable you only need to include the variable in **ONE** row of the table. **MM** will automatically add as many additional rows as needed for the other pieces of data.

Often times you will want to have more than one piece of information that is displayed on a given row that is a multiple variable. For example, if you are displaying quote line items you might want

to display, on each row, the part number, revision, unit cost, number of units purchased, etc. To do this, simply place the multiple variable for each in the proper table cell.

A bit of care must be taken, however. **MM** will add as many rows to the table as needed for the *first* multiple variable on that row. Thus, you should make sure that all of the multiple variables on a given row return the same number of rows.

The following is an example of a table that includes several multiple variables to display data from the activity log for a case:

Activity	Additional Info	Created On	Created By
ACT_STRING	ADDNL_INFO	CREATED_ON	BY

Suppose that the supplied case had 50 activity logs. **MM** would then create a table with 51 rows. The first row would be the header, and the other rows would each represent one activity log.

Note: You may include data in your table that isn't multiple variables. While the example above was only multiple variables, you may include other data including labels, numbers, formatting, and formulas. Just create the table as you normally would.

Fomulas

It is important, for many mail merges, to have formulas be part of your tables. For example, you might wish to multiply the net price on each row by the discount for that line item. You might also wish to have other rows that include subtotals and totals.

MM will automatically recalculate any and all formulas that you place in your table. It will do this for each cell that **does not** include a multiple variable. It will process them from the top row down, as it adds data to the table.

Note: It is crucial to understand, however, that the additional rows for multiple variables are placed in the table via a copy/paste operation. Thus, if you use absolute formulas, you will see errors in your merged table. Instead, you should use relative cell locations in your formulas.

=(D2*D3) ' Bad. Each row will total D2*D3

=PRODUCT(LEFT) ' Good. Each row will calculate properly

Special Variables

There are special variables that you can include in your document. These variables are not read from the database, but are merged in at the same time. They may be included anywhere in your document, and are surrounded by curly braces, as are standard variables.

Special variables, however, do not need to be defined in your profile.

Currently the following special variables are defined:

- 1) {NOW} - This variable will be replaced by the current date and time. The format of the date time is defined by the **MM** Date Format configuration item (see below).

Date Formatting

You have the option, with **MM** to format dates into whatever format you desire. This is controlled by a configuration item (see below). When you include a variable in your document that is a date/time value, you have the option of either formatting it, or leaving it in the default database

format. If you want to format it (using the format specified in the configuration item) you must include a special character at the start of the variable name. By default this is the pound character (“#”). This can be modified in the *mail_merge.cbs* file. If you do not include this character, **MM** will simply display the date/time in the default format provided by Clarify. The following are some examples of date/time formatting:

{CREATE_DATE}	- Not formatted
{#CREATE_DATE}	- Same variable, but formatted
{#NOW}	- Current date/time, formatted
{NOW}	- Current date/time, not formatted

Note: You must decide to format the special current date variable, or not just as you would any other date/time variable.

Attachments

You have the ability with **MM** to automatically attach the merged document to a Clarify object. This is accomplished via arguments when you call the mail merge API. For more details on merging, see the API section below.

One very important feature has been added to the attachments feature of **MM** that is important to discuss here. It is often the case that the merged documents (or any documents in an attachments directory) can have the same name. This would be very bad because the new document would erase previously-written document data.

When you create a mail merge document (and choose to attach it to a Clarify object), you have the ability to turn on “auto-numbering”. If you do this, the output file name you provide will automatically have an auto-incremented number added to the end of the file name. This insures that all file names are unique, and no data is lost.

If a user asked for a merge file to be called “my_output.doc”, and turned on auto-numbering, the first document might be called something like “my_output_44.doc”. If the merge were immediately run again, the next document might well be “my_output_45.doc”.

Performing Mail Merge

Running the actual mail merge couldn’t be simpler. The **MM** product ships with an API named *mail_merge*. It is located in the *mail_merge.cbs* file. For the Clarify thick client, you simply have to declare the *mail_merge* API, and call on it wherever you wish to perform a mail merge.

Typically, mail merges are performed on a button click, but you may run them wherever and whenever you see fit. For example, you could associate a menu item with a mail merge operation, or a message such as Find Caller on the new case form. Where and when you run mail merge is up to you.

An example of this ships with the **MM** product. Please see the *mm420.cbs* file for the working example.

For details on the mail merge API, please see the reference manual at the end of this document.

Note: You can use the *mail_merge.cbs* file in a CBBatch script if you wish to perform mail merge operations from the command line. No other files are required.

A Sample Of How To Use Merge

This section includes a simple example of how to use **MM** in your Clarify thick client GUI. It is not intended as a serious example. But it does demonstrate most of the features of **MM**.

The scenario presented here is that a user might like to create documents from a case. We chose the case object as most Clarify users will have cases in their system with which to test. If we chose some other object (such as quote), many customers wouldn't be able to run this example.

Even though the example is on the case object we have called our two merge templates "sample_quote.doc" and "sample_license.doc". The reader should not infer anything from these names other than the fact that we wanted to demonstrate how to merge from a list.

In this example, a new dropdown list is shown on the case edit form (#420). The user selects the document they wish to create, and presses the new "Merge" button. The mail merge is then executed, and the resulting document is added automatically as an attachment to the case. As an interesting aside, you could add a checkbox to the case edit form so the user could decide to attach to the case, or not.

The following are the steps you must follow to install the sample customization:

1. In Policies & Customers, add a new user-defined list named "Sample Case Merge". It should contain the following values:
 - a. *sample_quote.doc*
 - b. *sample_license.doc*
2. Create two new directories, "c:\docs" and "c:\output". If you do not like these names you may choose others. If you do use other names, please edit the mm420.cbs file, and change the \docs and \output directories to your new names
3. Copy the *sample_quote.doc* and *sample_license.doc* files to the \docs directory (or the directory name you chose)
4. Import the *sample_profile.dat* file with data exchange. This will give you the "Case Vars" profile data you need for the merge
5. In UI Editor, save a new user version of form 420 (or form 776 if you use service manager). Add the following two controls to the form:
 - a. A new dropdown list
 - i. Name: DDL_SAMPLE_MERGE
 - ii. Type: Popup List
 - iii. List Type: Sample Case Merge
 - iv. Destination: CASE_OBJ.alt_e_mail (we chose this field so that you don't have to add a new contextual object to the form)
 - b. A new pushbutton
 - i. Name: BTN_MERGE
 - ii. Caption: Merge
6. Add the new 420 (or 776) form to the proper resource config(s)
7. Compile the supplied *mm420.cbs* against the new form 420 (or 776)

You are now ready to try out the sample.

Most of the interesting features are to be found in the *sample_quote.doc* template. It contains a sample of the **NOW** keyword, and date/time formatting. It also contains some "Not Found" variables. Most importantly, it contains a table with a set of multiple variables, and some formulas that will automatically recalculate.

To show how silly the example is, the table header shows that it is a quote. However you may notice that the data written to the table is a set of activity logs for the case. The additional

information field is shown, as well as the activity code. This is just an integer that you normally don't see in Clarify. But we used it to be a fake "price", so that you can see that formulas for discounting and totalling work just beautifully with **MM**.

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

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

Note: There are other files that contain strings that you will import during the installation of **MM**. They are for the path editor form and for the message box form. Typically you do not edit these values, but you may if you wish.

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

Three form updates are provided with **MM**. They are:

Form Name	Form Number	Description
Administration Form	4260	Set up profiles and variables
Message Box	1947	Internationalized message box
Path Editor Form	2200	GUI for setting column paths

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 *mm_global.cbs*. This module will be compiled against the database.

Menu	Menu Item	Description
Policies and Customers Setup	MM Admin	Allows an administrator to set up profiles and variables

Configuration Items

There is one configuration item that you will import into Clarify as part of the installation of **MM**. It is important to understand what this item is and what it does. In this way, you can set the value for the item to cause **MM** to operate in the manner that is best suited for your Clarify implementation.

The following table lists the configuration item (shipped in *mm_config.dat*).

Item Name	Default Value	Description
MM Date Format	dd-mmm-yyyy	In what format should date/time variables be displayed?

Auto-numbering Schemes

There is an auto-numbering scheme provided with **MM** in the *mm_config.dat* file. Normally you do not have to do anything with this object. It will be imported as part of installed **MM**, and will operate without any other actions on your part.

If, however, you already have the First Choice products **Attachments Anywhere** or **Super Emailclerk** installed, you will need to make sure that you do **NOT** import this auto-numbering sequence, as it will overwrite the object (of the same name) included with those other products.

Requirements

This version of **MM** 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

MM is shipped to you as a zip file .

Files Included in MM

The following files/directories are provided with this product:

File Name	Purpose
code	Directory containing the ClearBasic code
files	Directory for data files to be imported
forms	Directory containing the forms files
Sample	Directory with an example of how to use MM
schema	Directory containing the schema modifications
mm_user.doc	This manual

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

File Name	Purpose
mm.dir	Clear Basic directives file for compilation

mm4260.cbs	Source code for the MM administration form
mail_merge.cbs	Code module with mail merge API
1947.cbs	Internationalized message box source code
at2200.cbs at_2200_fd.cbs	Source code for the path editor
mm_global.cbs	Global source code for MM
string.cbs	Routines for I18N strings
lists.cbs	Routines for I18N lists

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

File Name	Purpose
msg_box_strings.dat	I18N strings for the I18N message box form
path_edit_strings.dat	I18N strings for the path editor
mm_strings.dat	I18N strings for the MM forms
mm_config.dat	Configuration items, auto-numbering scheme, and locales

The following file is provided in the *forms* directory:

File	Purpose
mm4260.dat	Forms for profile/variable administration
1947.dat	Internationalized message box form
2200.dat	Path editor form

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

File	Purpose
sample_profile.dat	File with a sample profile and variables
mm420.cbs	Sample code file for Case Edit form
sample_quote.doc	Sample quote template
sample_license.doc	Sample license template

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

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

Installation

MM 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 (*mm.dir*).

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

- Update the Clarify Schema
- Import configuration files
- Import updated forms

Set up resource configurations
Compile ClearBasic code
Setting Environment Variables

Note: It is highly recommended that the **MM** 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 **MM** 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 "**MM**" directory, if necessary.
Switch to "**MM**" directory.
Unzip into "c:\clarify\fchoice**MM**" directory.

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

Updating the Schema

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

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 *mm_schema.sch* file provided in the **MM** 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

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
path_edit_strings.dat	Internationalized strings for the path editor

mm_strings.dat	Internationalized strings
mm_config.dat	Overall configuration items

Note: Before you import these files, you **MUST** edit the *mm_config.dat* file with an editor. Change the values of the items to reflect your preferences. For information about the settings you can change, see the section above on configuration items.

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

MM is provided with a set of new forms.

The new forms to be imported:

- 1947.dat
- at2200.dat
- mm4260.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.

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 *mm* 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) and *at* (for form 2200) in the text box
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 *mm.dir* file supplied to point to the location of the **MM** 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 are two lines that must be commented, and two that must be uncommented in the directives file, depending on if you use Flexible Deployment (FD), or not. The default lines (for normal 2-tier Clarify) are uncommented. If you are using FD, you must comment these lines, and uncomment the two FD lines. 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 (*mm.dir*) are located in the same directory. Change directories to that location and enter the following command to compile the code:

```
<path>\cbex -dir mm.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 *mm_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 *mm_global.cbs* module.

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
```

Limitations

There are a few limitations with the **MM** product.

The first limitation is that multiple variables currently only work with tables. This should not be a major limitation, however.

The other limitation is that there currently is no way to automatically create a .pdf file as part of the mail merge. We had hoped to do this. However the OLE interface provided by Adobe is insufficient to do what we would like. Should they improve the interface, we will add this feature.

Year 2000

MM 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

MM 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.

MM 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. **MM** 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, **MM** 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 **MM** 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 **MM** 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

API Reference

The next section of this document lists the available APIs for **MM**. Each API is documented with sections for:

- File Included In
- Syntax
- Description
- Parameters
- Return Values
- Examples

mail_merge

Included in mail_merge.cbs

Standard API

```
Public Function mail_merge(doc_template As String, _  
    output_doc As String, _  
    profile_name As String, _  
    root_objid As Long, _  
    attach_doc As Boolean, _  
    attach_name As String, _  
    use_autonumber As Boolean, _  
    show_it As Boolean) As Integer
```

Description

This API causes a mail merge to take place. The Word document template must be specified, as must be the profile name. The output document name (including path) must be specified.

The mail merge will be performed for the specific Clarify object as determined by the root table type of the profile, and the objid as specified in the argument. Given these first four pieces of data, the mail merge is performed.

The other arguments control features of the merge. If you wish the generated document to be attached to the root object, you can specify it with the first boolean.

Note: Only certain base types (see below) can have attachments in Clarify. If you are merging off of a different type, you may not add the attachment (unless you own First Choice Software's Attachments Anywhere product).

If you attach the generated document you may specify a name for the attachment. If you do not, the file name of the attachment (without the path) is used as the name. You may specify if the output document name should have an auto-number added to it (which is recommended).

Finally, you can specify if the merged Word document should be displayed (and left open) after the merge. If not, Word is not shown to the user.

The following objects support attachments in Clarify, and can have **MM** attach documents to them:

- bug
- bus_org
- campaign
- case
- communication
- contact
- contract
- demand_hdr
- diag_hint
- eco_dtl
- eco_hdr

email_log
interact
lead
lit_req
lit_ship_req
mod_level
opportunity
site
site_part
subcase
template
workaround

Parameters

Parameter Name	Required?	Description
doc_template	Yes	Path and name of the Word document to merge
output_doc	Yes	Path and name of the output Word document to create
profile_name	Yes	Name of the mail merge profile to use to find variables
root_objid	Yes	Objid of the object to merge from
attach_doc	Yes	Should the document be attached to the root object?
attach_name	No	If attaching, what is the name of the attachment in Clarify? If blank, output document file name is used
use_autonumber	Yes	If attaching, should an auto-number be added to the end of the file name?
show_it	Yes	Should the Word session be shown as the merge takes place, and should it be left open after?

Returns

Value	Meaning
0	No errors
-1	The input document template cannot be found
-2	The specified profile cannot be found
-3	The root record cannot be found for the objid and profile object type
-4	Cannot create an OLE connection to Microsoft Word
-5	Cannot properly set up the OLE connection to Microsoft Word

Examples

Merge document “c:\docs\case_template1.doc” using profile “Case Vars” for a specific case objid. Attach it to the case with a name, and with auto-numbering. Show the resulting output, and put it in an attach directory:

```

Dim ret_int As Integer
Dim case_rec As Record

Set case_rec = Cobj_CASE_OBJ.Contents
ret_int = mail_merge("c:\docs\case_template1.doc", _
    "c:\attach\case_merged.doc", _
    "Case Vars", case_rec.GetField("objid"), _
    True, "My attachment!!", True, True)

If ret_int = 0 Then
    Debug.Print "Success"
Else
    Debug.Print "Failure. Return code = " & Trim$(Str$(ret_int))
End If

```

Merge document "c:\docs\case_template1.doc" using profile "Case Vars2" for a specific case objid. Do not attach it, nor show it.

```

Dim ret_int As Integer
Dim case_rec As Record

Set case_rec = Cobj_CASE_OBJ.Contents
ret_int = mail_merge("c:\docs\case_template1.doc", _
    "c:\attach\case_merged.doc", _
    "Case Vars2", case_rec.GetField("objid"), _
    False, "", False, False)

If ret_int = 0 Then
    Debug.Print "Success"
Else
    Debug.Print "Failure. Return code = " & Trim$(Str$(ret_int))
End If

```