# Recently Opened Object Monitor 1.1.2

*Administration Series*

**Installation and User's Guide**

# Table of Contents

# Overview

This document describes an administration utility used to display Recently Opened Objects (ROO). This administration utility is known as First Choice Software, Inc.'s *Recently Opened Objects Monitor* (ROOM).

It is often the case that Clarify users open up objects, close them, and then want to re-open them easily and quickly. Suppose, for example, that a case is found by executing a case query. After the case is closed, the user realizes that they need to look at the case again. If they don't remember the case number, they have to execute the query again to find the case.

Many programs provide a mechanism to retrieve recently opened objects. For example, in MS Word, you can find recently opened documents via the *File* menu. The ROOM utility provides similar functionality for Clarify.

ROOM allows users to view up to 100 recently opened objects. Objects that can be stored and reopened from the GUI are Case, Subcase, Quote, Contract, Opportunity, Site, Contact, Part, Site Part, Solution, Change Request, Part Request, Part Number, and Action Item. Users can choose which objects ROOM remembers. Re-opening any recently opened object is easy. Just double click on it in the ROOM form, and it will be opened.

ROOM also remembers objects between Clarify sessions. If you have a set of recently opened objects, and leave Clarify, they will be available for you when you restart Clarify.

ROOM provides one new form – the Recent Objects form. This form can be opened from a new menu item in the desktop menu.. Once the form is opened, just use Clarify normally. Each time an object is opened, ROOM adds it to the list.

**Note**: If you don't have the ROOM form opened, objects will not be added to the list.

If you open an object that is already on the ROOM list, the object is moved back to the top of the list.

# 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 1.1.2

This version of **ROOM** fixes a tiny bug in recent part numbers. There was a small bug on form 611 (part number form) that could cause a runtime error (but no further problems). The bug has been fixed in this release.

To apply the change, simply copy the new version of room611.cbs and recompile.

## What's New in Version 1.1.1

This version of **ROOM** fixes a very small bug that was discovered in baseline Clarify function. It was possible, in certain cases, that when a site was opened via **ROOM** that a runtime error would be generated when the site was saved. This was due to a documented (but not yet fixed) bug in Clarify in the App.ShowSite method.

To apply this change, just recompile the new version of room2250.cbs supplied with this patch.

## What's New in Version 1.1

This version of **ROOM** contains a small enhancement to the product. If you do not need this enhancement, you do not need to install the update.

It is an unfortunate reality that the Clarify client can crash. Sometimes this can happen more often than desired. **ROOM** has a feature where it stores the recent information for a user so that it is available the next time they use Clarify. The recently opened object information is stored in the database, and is stored whenever the Clarify client is closed, or the ROOM form is closed.

But if the Clarify client crashes unexpectedly, some of the recently opened object information could be lost, and not written to the database.

This version of **ROOM** introduces a new configuration item. This item controls how often **ROOM** will write your recently opened object information to the database. If you do not feel the need for frequent updates, you can leave the value set to 0 (the default). This will cause **ROOM** to only save information when the form is closed (or Clarify is stopped).

To apply the change, modify the configuration item in the room_config.dat file. Then, reimport the file with dataex. Finally, recompile the roomglob.cbs and room2250.cbs files. Stop and restart Clarify for the change to take effect.

## Using ROOM

### Adding Clarify Menu Items

In the file *roomglob.cbs* there is one menu item that can be added to the Clarify system. The code from that file is as follows:

```
cs_menu.MenuBarID = 1002
cs_menu.AddItem "Desktop", "Recent Objects", "recent_objects"
```

This section adds a new menu item to Clarify. It allows the ROO Monitor GUI to be called from ClearSupport. The administrator may wish to modify this entry. It can be removed entirely (if not desired). The MenuBarID property can be changed to make the menu item appear in a different Clarify application (Valid MenuBarID properties are described in the ClearBasic Object Reference Guide in the MenuBarID

section). In addition, the menu item can be moved to a different menu by changing the name of the menu in the first argument to the AddItem call. Finally, the name that appears in the menu bar can be modified by changing the second argument in the AddItem functional call.

It may be that you already have a global Clear Basic module that contains an *initialize_app* method. You may have only one of these per Clarify database. You must take the code from the method in the roomglob.cbs file, and incorporate it into your *initialize_app* method (including the *set_locale* call).
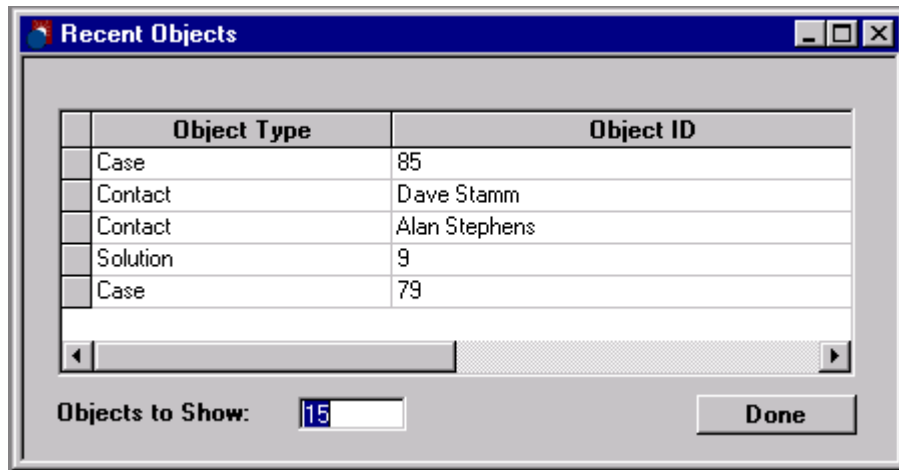
## Configuration Items

**ROOM** contains one configuration item. *ROOM Save Frequency* controls how often **ROOM** will save your recent object information to the database. If it is set to 0 (the default), **ROOM** will only save the data when you exit Clarify, or close the **ROOM** form. Setting it to an integer will cause ROOM to save the data to the database every *n* times data is changed in the **ROOM** form.

For example, suppose the configuration item was set to 10. Every $10^{th}$ opened item that changes the data in the form's grid would cause the data to be flushed to the database. This way, if a GPF does occur, not much data would be lost in the **ROOM** form. This configuration item is global for all users in a Clarify database.

## ROOM Form

This section describes the ROOM form provided with the utility. This form displays the recently opened objects for the user.



Double-clicking a row in the grid opens the selected record. The maximum number of objects to remember is shown in the text box at the bottom of the form. This may be any number from 1 to 100. Each user may remember a different number of objects.

The **Done** button is used to close the ROO Monitor form.

## Implementation

This section provides requirements, a list of the files that are included in this product, installation instructions, and other implementation considerations.

### Requirements

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

Clarify Version:     5.0 or later
Clarify Tools:       Data Dictionary Editor (ddeditor)

|  | User Interface Editor (uieditor) |
|  | Clear Basic Toolkit (cbex) |
|  | Data Exchange Utility (dataex) |
| Other: | An unzip utility |

## Packaging

**ROOM** is shipped as a zip file:

## Installation Tree

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

Switch to "c:\clarify" directory.
Create a "fchoice" directory, if necessary.
Switch to "fchoice" directory.
Create a "room" directory, if necessary.
Switch to "room" directory.
Unzip into "c:\clarify\fchoice\room" directory.

After uncompressing, the following installation tree would be established:
c:\clarify\fchoice\ROOM

The following file and directories are provided with this product:

| File Name | Purpose |
|---|---|
| room_user.pdf | ROOM user guide and reference document |
| cb | Directory containing the ClearBasic code |
| files | Directory containing data files to import |
| forms | Directory for the forms files |
| schema | Directory containing the schema modifications |

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

| File Name | Purpose |
|---|---|
| room.cbi | Clear Basic directives file for compilation |
| room2250.cbs | Clear Basic code for the ROO Monitor form |
| room321.cbs | Clear Basic code for the Solutions form |
| room328.cbs | Clear Basic code for the Select Solutions form |
| room334.cbs | Clear Basic code for the Edit Change Request form |
| room337.cbs | Clear Basic code for the New Change Request form |
| room411.cbs | Clear Basic code for the New Case form |
| room418.cbs | Clear Basic code for the Subcase form |
| room420.cbs | Clear Basic code for the Edit Case form |
| room502.cbs | Clear Basic code for the Part Request form |
| room611.cbs | Clear Basic code for the Part Number form |
| room672.cbs | Clear Basic code for the Site Part form |
| room712.cbs | Clear Basic code for the Contact form |
| room717.cbs | Clear Basic code for the Site form |
| room731.cbs | Clear Basic code for the Contract (entitlement) form |
| room775.cbs | Clear Basic code for the New Case form (Service Manager) |
| room776.cbs | Clear Basic code for the Edit Case form (Service Manager) |
| room9133.cbs | Clear Basic code for the Contract form (Clear Contracts) |
| room9600.cbs | Clear Basic code for the Select Opportunities form |
| room9601.cbs | Clear Basic code for the Opportunity form (Clear Sales) |

| | |
|---|---|
| room9620.cbs | Clear Basic code for the Action Item |
| room9630.cbs | Clear Basic code for the Opportunity form (Clear Telebusiness) |
| room9633.cbs | Clear Basic code for the Opportunity form (Clear Sales w/o TAS) |
| room9672.cbs | Clear Basic code for the Quote form |
| roomglob.cbs | Global Clear Basic routines for the ROO Monitor |
| STRING.CBS | Global Clear Basic routines for the ROO Monitor |

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

| File Name | Purpose |
|---|---|
| 2250_strings.dat | ROO Monitor form string definitions |
| room_config.dat | Configuration data for ROOM |

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

| File Name | Purpose |
|---|---|
| room2250.dat | ROO Monitor form |

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

| File Name | Purpose |
|---|---|
| recent.sch | Schema modifications required for the ROOM product |

## Installation

The **ROOM** 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 (*room.cbi*).

Now that the install package is unpacked, the following tasks need to be performed during the installation of **ROOM**:

Updating the Clarify Schema
Importing configuration files
Importing updated forms
Saving user versions of User Interface forms
Setting up resource configurations
Compiling ClearBasic code
Setting environment variables

---

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

---

### New Objects

First Choice Software has included the following object to implement the **ROOM**:

New Tables/Views

| Table Name | Table Number | Purpose |
|---|---|---|
| fc_string | 3500 | Internationalized Strings |
| fc_locale | 3505 | Internationalization |
| recent_obj | 3630 | Stores Recently Opened Objects |

If you are already using any of the above table numbers, you may modify the *recent.sch* file and change them to other, unused numbers in the user-defined range (1005-4999).

## New Form

The following new form will be added to the Clarify system for the ROO Monitor:

| Form Name | Form Number |
|-----------|-------------|
| ROO Monitor form | 2250 |

If you are already using any of the above form numbers, you may modify the appropriate .dat file and change them to other, unused numbers in the user-defined range (1005-4999).

## Updating the Schema

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

### Back up the Database

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

### Importing 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 recent.*sch* file provided in the ROO Monitor install tree.
5.  Choose *Apply Changes* from the *Actions* menu.
6.  Select the *my_db.sch* file.
7.  When asked, press *Continue*.
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

## Import Data Files

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

2250_strings.dat                    Internationalized strings for form 2250
room_con fig.dat                    Default locale settings

> **Note**: You should edit the room_config.dat file with your favorite editor, and change the value of the configuration item, as described above, if you do not wish to use the default value.

To import the data files, 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, 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.

## Importing the User Interface Forms

The ROO Monitor provides one new form that needs to be imported.

- room2250.dat

You can import the file through the UI Editor or by using DataEx.

To use dataex, use 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**: 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.

To import the form using the UI Editor:
1. Start the UI Editor.
2. Select Import from the File menu.
3. Select the file from your directory.

## Saving user versions of the User Interface Forms

Before compiling form code, you must first create a user version of the form and add that version to the resource configuration.  To create a user version of a form:

1.  Start the UI Editor
2.  Select ID from the dropdown list on the Form Select.
3.  Type in the form id number in the search text box.
4.  Press the *List* button.
5.  Double-click on the form.
6.  Go to File→ Save As
7.  Type **ro1.0** for the user version.
8.  Close the form.

> **Note**: You must decide which objects you want to have show up in the ROO Monitor. You only need to save new user versions for the forms for those objects.

## Set up Resource Configurations

After the form file has been imported successfully and you have saved user versions of the forms of the objects you wish to display as **ROOM**, the 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 (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.  Type **ro1.0** 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.  Press the *Replace* button.
9.  After the save is completed, press the *Done* button.
10. Repeat steps 4-9 for any other resource configurations you wish to modify.
11. Exit the User Interface Editor.

## Compile ClearBasic Code

You can now compile the supplied Clear Basic code against the forms you have imported or saved new versions of.

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.

Before you compile the code, you should edit the *room.cbi* file supplied. Verify that the Clarify version (the $1^{st}$ 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 ($2^{nd}$ string after the "F") should be the same. Also, make sure the lines are not commented out (line has no ' character before it).

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

```
<path>\cbex –dir room.cbi
          –batch
          –overwrite
```

Where:

        \<path\>  is the path to the cbex program.

You can also compile the code from a CBEX GUI.  To open the CBEX GUI, double-click the CBEX.exe icon in your directory.  Open room.cbi and click the import button.  You can turn off the files you don't want to compile  by selecting the row in the custom list, clicking the off radio button, and clicking the Replace button.

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.

### Setting Environment Variables

There is one environment variable that should be set for ROOM. Each user should set the environment variable LC_ALL to be equal to the locale they use. First Choice follows the XPG/4 specification for locale names. The environment variable controls the locale (language) of the labels displayed on the ROOM form.

An example of setting the environment variable (to US English, the default) is:

LC_ALL=EN_US

If you do not set this variable, ROOM assumes that you are operating in the EN_US locale.

## Internationalization

The ROO Monitor form is localized. That is, the labels and caption on the form will displayed for the locale set in the LC_ALL environment variable.

The base product only ships with strings for US English. To add other locales, copy the contents of the *2250_strings.dat* file to another file, change the locale strings from EN_US to your locale, and change the value of the string data. Then import the new file into the database with dataex.

## Known Limitations

There are a few limitations known to exist with **ROOM**:

1.  Internationalization issues. There is one that is currently known. It has been opened as an issue with Clarify. When it is corrected, a patch will be issued for this product.
    *   The column headers of the grids cannot be set at run time. To localize, you must have a new version of each form per locale.

For more information about internationalization and localization, please contact First Choice Software, Inc.

## Year 2000

ROOM has no date/time calls or functions. Thus, no year 2000 issues are expected.

## 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