FIRST CHOICE SOFTWARE

First View™

Volume 3, Number 2 August 2003

Welcome to the second issue of ***First View*** for the year. Like you, we've been extraordinarily busy since the last issue was published. We will explore some of the new people, new products, and new ideas that have enlivened the past three or four months at First Choice, and – as always – present a genuinely useful project in the Technical Corner.

This issue of ***First View*** discusses:

If you know of someone who might enjoy a copy of ***First View***, please share it with them. Also, have them send an email to add-firstview@fchoice.com. Make sure they include their name, phone number, and email address. We will send them future editions of the newsletter. If you do not wish to receive future copies of ***First View***, please send an email to remove-firstview@fchoice.com. Make sure your email address is either in the title or body of the email. We will remove you from our mailing list immediately.

Upcoming Training/Education

Regional User Group Meetings

October 8 – San Francisco
October 10 – Chicago
October 13 – Atlanta
October 15 – Boston

FCFL and fcClient Programming Class

October/November TBD – Boston
For more information about this class, visit

Employee Profile: Newt Hamlin

Newt Hamlin, a seasoned entrepreneur who started his first company in l97l, joined First Choice as CEO this spring. We thought you might like to know a little more about him.

In 1971, following graduate studies in communication and business at Northwestern University and the University of Texas, and stints with Alcoa, Whirlpool Corporation and NW Ayer Advertising, Newt co-founded a marketing services firm in New Orleans. He completed a merger with an existing company and grew the entity to one of the largest in the South. After selling to a group of local investors, Newt moved to Dallas where he founded a CRM software and training company, a financial services firm and another marketing services company. He sold the software and financial services companies to employees, and the marketing firm to J. Walter Thompson, and in 1991 retired to Austin. Failing miserably at retirement, Newt has since become partner in a fast growing third party logistics company which he sold to Wal-Mart; led the purchase of a great specialty foods company, Jardine Foods; and in 1999 co-founded VIEO, which is currently releasing into Beta the world's first appliance that combines application-aware networking hardware with intelligent management software to ensure data center application resources are available, on demand, to satisfy business needs. In the past two years, Newt has brought CEO level management to several other companies, and has led fundraising that produced in excess of $72M in venture funding for his companies in the past four years. In late 2002, he co-founded CN Group, which combines three entities that provide corporate management, investment banking and M&A services, and sponsorship of the leading investor conferences in the region.

Newt has been a long time supporter of community, civic and education initiatives. He serves on several boards and advisory boards and has been an active supporter of Miss. State University, the University of Texas, SMU, and other universities. He is an avid golfer, reader, and collector and enjoys travel. He is married to Dr. Judy Hamlin, a noted author and consultant. His only son Andrew, along with his wife and Newt's first grandson, also live in Austin.

New Products and Features

- **EEO for the fcClient**
  Enhanced Email Out features are now available for the thin clients fcSupport, fcQuality, fcCallCenter, and fcLogistics. We use this at First Choice to email unlimited literature or other attachments from within a case, send BCCs, use signature files, access Clarify and email address books, and enjoy the freedom of being able to enter whatever text we want in the subject line.
- **Connection pooling for the fcClient**
  Connection pooling allows you to support more clients than the number of DB connections available. It enables a virtually unlimited number of users to access the system.
- **Enhanced commitment calendar for the fcClient**
  View commitments by month, day, or to-do list. You can also open up and fulfill commitments from the calendar.
- **fcCodeConverter**
  This utility automatically converts CB programs into a variety of web-based languages, including JavaScript, VBScript and Visual Basic. Together with the fcFormsConverter, this tool makes it quick and easy to move to a thin client, web environment. Our developers came up with this idea for their own use, and we figured we shouldn't be selfish and keep it to ourselves. [In Quality Assurance, General Availability 4Q03.]

- **fcLogistics**
  This agent-facing thin client will allow your field and warehouse personnel to perform all logistics tasks in a simple web application that looks and feels like Clarify, but can be run from any location, including, for example, from wireless tablets. [In QA, GA 3Q03.]
- **Interested Parties**
  Think of this product as a kind of user-centric business rule manager: it allows every user to be notified (via email to a user-defined address) when selected events occur involving selected objects, for example, whenever anything is logged against a particular site of interest. Thank Rockwell for this one.
- **FCFL-based API Toolkits for ClearLogistics**
  We have moved our popular ClearBasic APIs for ClearLogistics to the FCFL/Web world.

Other near-term products and features will be demonstrated or discussed at the regional user group meetings (see below).

Connections and Regional User Group Meetings

We are working on a number of new and exciting projects that, unfortunately, will not be ready in time for a Connections Conference this fall, so we've decided to schedule the Conference early next year.

However, we do feel it's important to keep in touch with our customers, and we appreciate that travel budgets are likely to be constrained for the remainder of this year, so we will instead sponsor a series of one-day Regional User Group Meetings. The goal of these meetings will be to help you enhance the return on your existing Clarify investment, and to expose participants to useful production solutions for slaying what are generally perceived as the chief Clarify "dragons":

Dragon # 1	**Data cleansing/merge, purge/archive.**
Problem:	Over time, any CRM system will develop "too much data". Worse, the data will also tend to become "fuzzy" with, e.g., duplicate entries and other ambiguities. These reduce not only system performance but also data integrity.
Alternatives:	- Live with it. System performance and data integrity will continue to deteriorate; this will lead to call handling inefficiencies and probably to customer satisfaction issues.
	- On the purge/archive side of the problem, dataex may be used; this tool is, however, slow and – for high volume operations – unstable, leading to database corruption.
	- On the purge/archive side of the problem, a competitive product is available for use with the case object only.
	- First Choice Contact/Site Merge and PARE/DIET. Contact/Site Merge is the only easy way we know to eliminate data "fuzziness" by merging and cleansing duplicate entries. PARE/DIET is a bullet-proof, high-performance tool for both purging and archiving essentially any key object from a Clarify database. Such archived objects may be viewed and, if desired, instantly restored using the graphical DART product.

Dragon # 2	**Web integration.**
Problem:	How to provide flexible, web-based access to the Clarify database for both agents and customers?
Alternatives:	- For ClearSupport and ClearSales only, upgrade to release 12.x and purchase iClient, iSales and/or eSupport from Amdocs. This alternative provides no solution for ClearQuality, or ClearLogistics deployments; such deployments will require expensive, time-consuming custom work. Nor is this alternative available to customers unable to upgrade to 12.x, or to those who have multiple-version deployments in place. Finally, customers who require partitioned databases – i.e., those who are using data restriction – will find no solution here.
	- First Choice provides thin clients that will work with any Clarify release, support data restriction, and deliver the application functionality of ClearSupport, ClearSales, ClearQuality, ClearLogistics, and ClearCallCenter today. The fcSS self-service product also provides access to any Clarify database for your customers. In addition, fcSS is internationalized and localized. This means that your customers can view your self-help web system in their native language. Installation and deployment are quick and easy, and will not distract scarce people resources from other projects.

Dragon # 3	**Agent productivity.**
Problem:	Clarify users have identified many features that would significantly improve agent productivity. In large deployments, even a modest decrease in the time it takes to perform a particular call-handling task can yield tremendous cost savings and/or a measurable increase in customer satisfaction. If such features can be obtained at reasonable cost, the enterprise can easily improve ROI for the system.
Alternatives:	- Ask Amdocs to add a desired feature.
	- Undertake a custom project.
	- Obtain (where it already exists) or request the desired feature from FCS.

Dragon # 4	**System integration/extensibility.**
Problem:	Few CRM system deployments “stand alone” – they are typically tied into other systems with which they share information; and few systems of any kind “stand still” for very long. This means that new integration requirements are constantly added to the task of running a CRM system, and it also means that it’s probably a good idea to avoid “hard-wired” solutions. IT organizations need a robust, powerful, yet simple mechanism for effecting and maintaining the required changes and additions to the system.
Alternatives:	- ClearBasic: expensive to use because it’s a relatively low-level approach.
	- eBF: expensive to use, as above; a few high-level Business Objects are available and if they match enterprise needs, this may be a viable approach. There may be performance issues.
	- SQL: expensive to use, as above.
	- FCS API Toolkits and FCFL provide a high-level, bullet-proof, cost-effective solution.

Dragon # 5	**Upgrades/multiple instances at different versions.**
Problem:	Upgrading the Clarify DB is no easy matter, especially across multiple versions.

Alternatives:	- Clarify upgrade scripts may be used, typically for upgrades across one or at most two release levels per script. These do not support upgrading (or switching) the underlying RDBMS, e.g., from Oracle to SQLServer, for which little help of any kind – apart from consulting services - is available from Amdocs.
	- The FCS Multi-Version Upgrade Tool enables you to upgrade your database from any version of Clarify to any other version with a single automated process. It will handle both database and form customizations; it also supports moving from one RDBMS to another, as well as RDBMS upgrades (e.g., from SQLServer 7 to SQLServer 2000).

The Regional User Group meetings will also promote interaction with other regional customers, provide the opportunity to form local User Groups and Special Interest Groups, and encourage participants to help direct future FCS R&D.

Proposed Agenda:

0900	Breakfast
0930	Introduction: FCS status (what’s new, why; where we plan to go, why)
1000	Dragon #1
1500	User deployments: Case Study
1130	Dragon #2
• Lunch	
1300	Dragon #3
1330	General discussion of local user deployments (ad hoc)
1400	Dragon #4
• General discussion of other dragons	
• Dragon #5	
1530	Wrap-up
1600	Depart

Systems will be set up and available for individual use/exploration, with a focus on coalescing and upgrading Clarify DB instances. FCS experts will be on hand to assist and to answer questions.

Proposed Calendar:

10/08	San Francisco
10/10	Chicago
10/13	Atlanta
10/15	Boston

Specific locations will be announced within 30 days; but we wanted to give you the opportunity to mark your calendars with as much lead time as possible. Please visit our web site for the most current information about the regional user group meetings.

Technical Corner

In each issue of ***First View,*** the Technical Corner section contains a tip, example, or technique we have found useful. We include them, along with appropriate code examples, in the hope that you will find them useful as well.

In a previous edition of ***First View*** we described how to create "new" queueable objects in Clarify. For many years this has been a high priority for many customers. The solution we presented, while useful, is a bit of a "hack." The idea behind the customization is to use an existing queueable object that you are not using, and to "take it over" for your purposes. If you haven't read that article (in ***First View 2-1***), you can request a copy from First Choice Software.

There are some issues with this approach, however. The first is that there are a limited number (9 currently) of queueable objects in Clarify; you may not have many (if any) left to use. The approach also requires a lot of customization. Finally, users must be aware, when they upgrade to new releases of Clarify, to carefully check to make sure that the "new" objects are not broken.

What would be really wonderful is to have a true new queueable object. A new table in the database that is just natively queueable, as are objects such as case, part request, and quote. With the base Clarify architecture, however, that just isn't possible.

But now it is. This article describes how to create new queueable objects for your users with your existing Clarify system. It doesn't matter what version of Clarify you're using, or which Clarify modules you have implemented.

The very exciting thing about adding new queueable objects to the system (besides the fact that customers have been asking for this for years) is that customers can expand their systems in a truly *useful* way. They can have exactly the objects they need, and apply the proper workflow to those objects.

There is one catch – to get these new queueable objects you must use First Choice's new thin client environment, *fcClient*. *fcClient* has an open architecture which allows many new and exciting features, including custom queueable objects. Any objects that you create this way can be seen in the Clarify thick client, but cannot be viewed in WIPBINs and Queues in the Clarify thick client console.

This isn't a major limitation for the many customers that are now using *fcClient*, as it is 100% compatible with Clarify 4.5 -> 12.0. It can be installed in under an hour, and will work, out-of-the-box, with your existing system.

The rest of this article describes how to add a new queueable object to your system. The article will describe a new custom queueable object for the Quality module: The Build.

What is a build?

In a Defect/Enhancement system (such as *fcQuality*), customers have often requested the ability to have a workflow object of a *build*. A build represents the building of a software system. The object often will be created by a release manager. The release manager will determine which CRs (Change Requests) will be built in the specific build.

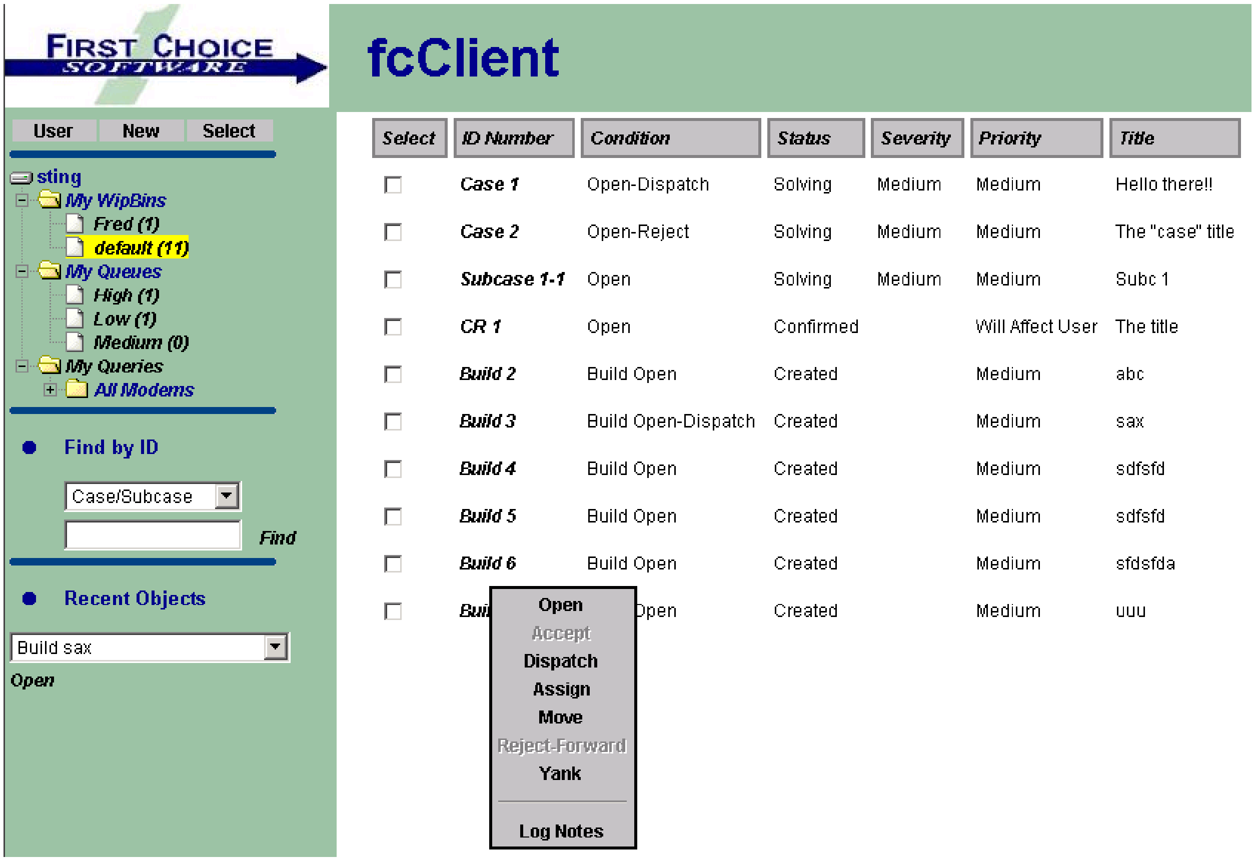
Typically, a build will be related to a part number (and revision). This is *what product* is being built. The build should also be related to some CRs. This tells the build master *what new files* should be included in the build.

Anatomy of a queueable object

To create a queueable object we must perform certain tasks:

- Create the proper data model
- Update the workflow operations (dispatch, accept, etc) to work with the new object
- Create the list and data items necessary for your object, including conditions, status codes, and auto-numbering scheme
- Write a form to create/update the object, a form to select the object
- Add menus (and web commands/privclasses) for the two new forms

- Modify the console
  - Modify the WIPBIN/Queue main display to show the object
  - Add a right click “context menu” for objects of the new type in a WIPBIN/queue
  - Enable drag/drop for workflow operations for the object
  - Add the object to the “Find By ID” section
  - Enable the Recently-Opened Object Monitor to work with the new object
  - Enhance the stored procedure that displays the totals of how many items are in each WIPBIN/queue to include the new object
- Modify the activity log form to display for the object
- (Optional) Customize business rules to fire on the new object



Select	ID Number	Condition	Status	Severity	Priority	Title
□	Case 1	Open-Dispatch	Solving	Medium	Medium	Hello there!!
□	Case 2	Open-Reject	Solving	Medium	Medium	The "case" title
□	Subcase 1-1	Open	Solving	Medium	Medium	Subc 1
□	CR 1	Open	Confirmed		Will Affect User	The title
□	Build 2	Build Open	Created		Medium	abc
□	Build 3	Build Open-Dispatch	Created		Medium	sax
□	Build 4	Build Open	Created		Medium	sdfsfd
□	Build 5	Build Open	Created		Medium	sdfsfd
□	Build 6	Build Open	Created		Medium	sfdsfda
□	Bui	Open	Created		Medium	uuu

Fig 1. The console enabled for the new build queueable object

Data modeling for the Build object

The first item that is required for data modeling is to add a very special new table called ***table_queueable_obj***. This object must be added only once to the schema. It contains one row for each new queueable object type. The columns of this table are the names of the key fields and relations in the new queueable object. For example, there is a column defined for the relation to the WIPBIN. Since the name of that relation (for the build object) is *build_wip2wipbin*, that is the value that is placed in the *wip_rel* column.

The next critical item is to perform the data modeling for the new object. Any new queueable object in Clarify **MUST** have certain fields and relations. They are:

- A field for the objid
- A field for the ID number
- A relation to the current WIPBIN (table_wipbin)

- A relation to the condition record (table_condition)
- A relation to the current owner (table_user)
- A relation to the current queue (table_queue)
- A relation to the previous queue (table_queue)
- A relation to the current status (table_gbst_elm)
- A relation to the status changes for the object (table_status_chg)

There are also certain fields/relations that are not required, but highly recommended. They are:

1. relation to creator (table_user)
2. field for date/time of creation
3. relation to activity logs (table_act_entry)
4. relation to notes for the object (table_notes_log)

There may be many others that are added to the new object, but they are completely optional.

For each new object that is added to the system you must also add three views. One is a view that joins the object with the WIPBIN table. The next joins the new object with the queue table. The final one joins the object with the activity log table. It is helpful (but not required) if you use the naming convention that Clarify has set up. For example, for the build object, the names of the views would be: *table_wipelm_build, table_queelm_build, table_build_alst*. These are the naming conventions used in our sample code, and simplify the cut/paste process for you.

Workflow Operations

It is important to be able to perform workflow operations on the new queueable object. The operations that must be allowed are: dispatch, accept, reject, forward, assign, move, yank, and change status. The good news is that version 3.4 (and later) of **FCFL** ships with workflow operation APIs available to you that already work with any custom queueable object you create. The only thing you have to do to enable these APIs is to add a row (with proper field/relation names) to the *table_queuable_obj* table. The APIs will read the names of the fields and relations in that table, and will perform the proper workflow operations, including activity logs and time bombs.

Define List Items

You must now determine certain key data for your object, and make a data import file for them. The most important items you must create include:

- Condition codes – You should have at least Open and Closed. You may decide on other conditions as appropriate,
- Status codes – Each condition has one or more status codes. You **must** create a *gbst_lst* list with the condition name as the title, and each status code as a *gbst_elm* item of the list,
- Autonumbering scheme – You must add a row to *table_num_scheme* for your new object,
- A row must be added to *table_queueable_obj* that describes your object for the workflow operations,
- Any other lists (such as user-defined lists) needed for your object.

Create New Forms

The next important step is for you to create two basic forms. The first of these will be used to create and modify your new queueable object (a build object in our scenario). The other form is a *Select* form that allows you to query the database to locate a specific build object.

There are many things that you can put on your queueable object form. We recommend that you look at some of the existing objects for ideas of what you should and should not include. You can use a lot of cut/paste from those objects to easily build your forms. Or you can contact First Choice Software for samples.

The following items are highly recommended to include on your form:

- Menu items (in the *Actions* and *Log* menus) for all of the workflow and logging operations, just like the *Case* form,
- Menu items to display the activity log, to print the object, to save the object, and to close the form,
- Menu items to close the item, reopen it, and to change the condition to other defined conditions,
- A tab with detail information for the object,
- A tab that displays other information such as the current owner, current queue, and current WIPBIN.

Menus And Privclasses

You should add (to the *\pages\console_left.asp* file) new menu items for the two new forms. You can add these to the menus (which are <div> controls named *newMenu* and *selMenu*. Each of these menu items should be controlled by a "web command." These commands allow you, for given privclasses, to turn on or off any function in the *fcClient*, including menu items.

You can create web commands either in the List section above, or by using the provided *fcAdmin* program. The name of the web command that you create in the GUI must match the name of the web command you create in the *\pages\include\inc_web_privclass.asp* file. You can then use the Privclass GUI provided in the *fcAdmin* program to assign the command to the proper privclass(es).

Note: The *fcClient* privclasses are exactly the same as the ones you use in the Clarify thick client.

Modify The Console

The "major" step in the process is the modification of the console to work with your new custom queueable object. The good news is that this is fairly easy, and consists mostly of a few cut/paste operations in a few .asp files. You can contact First Choice Software for a document that contains the exact changes you have to make. The following list details the items, and lists the files that you have to modify for each step:

- You must add code to the *\pages\console_main.asp* file so that the new queueable object is queried for the WIPBIN or queue, and is displayed in the console. This file also contains sorting code for the new object so you can sort the console items properly,
- You must add code to the *\pages\console_main.asp* file to tell it to display a context menu (by right-clicking on the queueable object). Then you must modify the *\pages\code\context_menus.js* file to add the context menu and functions that are called for each menu click. This file also contains code that enables/disables the menu items based on the condition/status of the queueable object,
- The queueable object items in the WIPBIN/queue display must also have drag/drop enabled for them. This is performed by making two simple changes to the *\pages\workflow\drop_action2.asp* file,
- The *Find By ID* section must be augmented to show (and process) find by ID requests for the new object. This is accomplished with modifications made to the *\pages\console_left.asp* and *\pages\query\find_by_id.asp* files,
- The *Recently-Opened Object Monitor* on the console is augmented to work with the new object with a small change made to the *\pages\console_left.asp* file,
- *fcClient* also augments base Clarify by displaying the number of items each WIPBIN or queue in the console view. For each new queueable object you should modify the system so that items of that type in the WIPBIN/queue add to the total. These totals are calculated in stored procedures. So, you must modify one of: *mssql_fc_wip_q_counts.sql*, *oracle_fc_wip_q_counts.sql*, or *sybase_fc_wip_q_counts.sql*, based on the database system you use. It should be mentioned that this feature is optional. If you don't use the feature then you do not need to modify the stored procedures.

Note: There are a few other files you must modify, but these are detailed in the document available from First Choice.

Activity Logs

It is almost a certainty that you will want to display the activity log for your new object. A very small change to the *\pages\workflow\grid_activity_log.asp* file will insure that your activity logs will display correctly.

Note: This is a general feature – you only need to make the change once for all new queueable objects.

Business Rules

All of the workflow operations automatically enter business rule events (in *table_time_bomb)* for your new queueable object. If you wish to, you can also add events for actions such as *Create* and *Close* when you write the code for your new form.

If you wish to, you can modify the business rule system to work with your new object. This is out-of-the-box functionality that is described in Clarify's Business Rule Guide.

Wrapping It Up

That's about it. If you follow the steps above, you can have any new queueable object available for your users in quick order. While this sounds pretty simple, there are a few cautions:

- There are a number of steps you must follow, and follow correctly. Please contact First Choice Software for more information before you ever attempt this operation,
- Any new queueable object you create will work properly within the confines of the *fcClient*. If you attempt to view them in the Clarify thick client, you will not see them in the WIPBINs and queues. This is because the Clarify thick client console cannot be customized. You can, if you wish, create thick client forms to display the objects, but you will not see them in the console.

But, if you can live with the limitation #2 listed above, this is an ***extremely*** powerful enhancement to your Clarify system that is guaranteed to delight your users.

Hopefully this information is useful to you. If you have questions about it, please feel free to post them in the *fcConnect* user forum located at: http://www.fchoice.com/fcconnect.

Clarify is a registered trademark of Amdocs Ltd.