

Modernizing Accept/Display Interfaces with Droplets

Independent Software Vendors (ISVs) are facing a daunting dilemma. Business applications that have stood the test of time, and have been refined and specialized for the tasks that they perform, are showing their age.

The business market is demanding modern, intuitive interfaces on a variety of non-Windows platforms, client workstations, mobile platforms, and thin-client terminals. And yet, when applications have been written in procedural code with display logic mixed in with business and data logic, the challenge to provide such an event-driven, graphical interface without interrupting the on-going development of business functionality is difficult to resolve.

A variety of choices are available to meet these challenges. Most involve considerable investment and huge opportunity costs. Rewriting the application as a client/server or Web services application will provide all the user-interface benefits required by current-day users but will typically require the ISV to halt enhancing business functionality for an extended period of time. Additionally, the available programming resources will likely require time and training to make the switch from a procedural programming language to an event-driven one. At the end of this exercise the ISV will emerge with an application with no new business features as part of the upgrade to the new platform support and user interface.

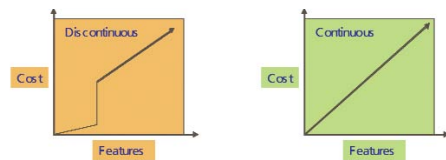


Figure 1: Overcoming the Discontinuity Dilemma in Modernization

This pause in feature development exposes the ISV to competitive risks, delays introduction of new services and entry into new opportunities. Ideally, the ISV would be able to continue its feature development and offer the modern aspects of business applications with minimal impact to the development schedule and resources.

A technology is available to ISVs today that removes the functional discontinuity normally associated with interface modernization initiatives. Droplets, a Micro Focus Alliance Partner, offers ISVs the ability to provide Rich Internet Applications (RIAs) user interfaces for their applications written in COBOL. RIAs run on standard network servers, can be accessed via the Internet/Intranet, and have user interfaces compatible with desktop applications and mobile platforms. They offer the end user ease-of-use, universal access, speed, power, and responsiveness. Application providers enjoy ease of deployment, maintenance upgrades, platform independence, small client-side footprints and low-bandwidth consumption. Developers benefit from working in a single development language and the "write once, run anywhere" technology.

How does Droplets work?

The Droplets RIA architecture provides API component level controls, server-side management of applications and sessions, and smart rendering of the client UI.

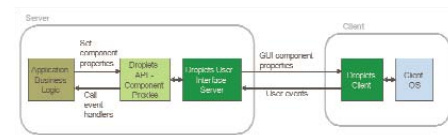


Figure 2: Overview of the Droplets RIA architecture

While Droplets customers have typically used a toolkit or SDK for reworking their applications, Droplets has gone a step further: Micro Focus applications can now directly benefit from a Droplets RIA solution through an adapter that does not require re-coding through a toolkit. This is very valuable, because the adapter addresses a large issue in discontinuity to new user interfaces: bridging between the "procedural" programming model of COBOL and the "event-driven" programming model of Windowed user-interfaces.

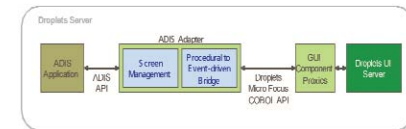


Figure 3. Accept/Display Interface System (ADIS) bridge

Once the COBOL application is connected through the adapter, Droplets also provides a COBOL SDK, if desired, to add other types of functionality common to graphical applications. Since the adapter has taken on the tough business of bridging procedural code to event-based, an application can now use the event-driven model of the Droplets SDK.

How easy is Droplets to start using?

The initial migration to Droplets requires no code changes. To get base-level functionality only a re-link of the application with the adapter may be required. After that incremental migration can be done to access numerous advanced features and benefits, putting the ISV squarely back on the proportionate effort and proportionate return line.

Each step in the process builds on the earlier ones or stands in isolation, and can itself represent an end-point depending on the needs of the project and whether full, native GUI deployment is a business requirement. These steps are outlined below:

1. Re-link to GUI: Immediate conversion and usability.

As an example, a simple 80x24 terminal displayed calculator program can be automatically converted to a RIA application by:

- Re-linking the application with the Droplets ADIS library
- Accessing the program as a Droplets application



Figure 4: The calculator example.

2. Add GUI Components: Add a component not normally found in COBOL but found in other languages today.

For instance, an image can be added to the application with a few lines of code:

```
local-storage section.
01 imagex      pic s(9) comp-5 value 250.
01 imagey      pic s(9) comp-5 value 50.
01 imagewidth  pic s(9) comp-5 value 50.
01 imageheight pic s(9) comp-5 value 100.
01 imagename   pic x(50) value z'mypicture'.
01 imagetoken  pointer value NULL.
...
PROCEDURE DIVISION.
...
call "DaDisplayImage"
using by reference imagetoken, imagename,
by value imagex, imagey,
imagewidth, imageheight
```

Figure 5: Code segment to add an image

3. Add Sub-windows: Create sub-windows to an application instead of pages. The main COBOL program is still the "master".

This is accomplished with a few more lines of code to add sub-windows:

```
local-storage section
01 mywindow   object reference mywindowtype.
...
PROCEDURE DIVISION
...
invoke mywindowtype "create" returning mywindow
call "DaShowWindow" using by value mywindow
```

Figure 6: Code segment to add sub-windows

4. Add ADIS Sub-windows: Now perhaps, we want a new application (written using Droplets COBOL SDK) to utilize an existing COBOL application within a window.

The main application window is now a Droplets application. The ADIS program is embedded in a Droplets window.

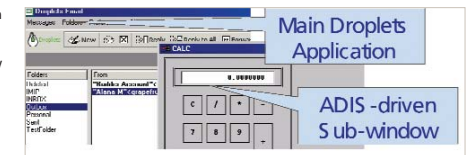


Figure 7: Embedding an ADIS window within a Droplet

5. Deploy Native GUI Application using the Droplets API: This step assumes that with time, the user interface functionality of the existing program has been subsumed into a full Droplets COBOL SDK approach. This is not necessary but is available to the programmer over time.

As you can see, by starting with an adapter approach and allowing it to tie with an SDK, many options and stages are opened up for the programmer. It is inviting because immediate success can be achieved and careful consideration based on user feedback can drive the next round of user interface enhancements.

Micro Focus and Droplets are working together to improve technology for the Micro Focus COBOL developer community. The Droplets method of addressing the needs of the marketplace for RIA will significantly benefit the ISV community, continuing to enhance their applications without the significant expense, risks, and opportunity costs of an application re-write.

To learn more about rapidly deploying your Micro Focus COBOL applications with Rich Internet Application technology please visit: www.droplets.com

Or e-mail Irving Abraham: irving.abraham@microfocus.com